IBM Operational Decision Manager
Version 8 Release 5

Configuring Operational Decision Manager on WebSphere Application Server

IBM
Chapter 3. Configuring Decision Server Events on WebSphere Application Server

Planning your Decision Server Events environment ........................................ 95
Creating a Decision Server Events environment ........................................ 95
Choosing a Decision Server Events topology ........................................ 96
Customizing the event runtime .............................................................. 96
Configuring a stand-alone Decision Server Events environment .......... 98
Creating a new Decision Server Events application server profile ........ 98
Creating a new profile by using the Profile Management Tool ............... 98
Creating a new profile by using the manageprofiles command .......... 100
Augmenting an existing WebSphere Application Server application server profile with Decision Server Events ........................................ 102
Augmenting an existing application server profile by using the Profile Management Tool ........................................ 103
Augmenting an existing application server profile by using the manageprofiles command ........................................ 105
Unaugmenting an existing application server profile by using the manageprofiles command ........................................ 107
Editing the setenv files ............................................................................. 108
Customizing the event runtime environment ........................................ 108
Configuring the event runtime database for Decision Server Events ...... 108
Creating the event runtime database ....................................................... 109
Manually creating event runtime tables .................................................... 110
Deploying tables in separate databases .................................................... 110
Deploying the event runtime in a case-sensitive database manager ...... 113
Changing the event runtime messaging provider ................................. 113
Configuring WebSphere Application Server default messaging to be the JMS provider ........................................ 113
Configuring WebSphere MQ to be the JMS provider ............................ 115
Configuring WebSphere MQ .................................................................. 115
Configuring Decision Server Events ....................................................... 115
Setting properties .................................................................................. 117
Configuring properties by using the WebSphere Application Server administrative console ........................................ 118
Configuring properties by using the wsadmin scripting tool ................ 118
Configuring properties by using the Properties tool ............................ 119
Configuring how step data is stored ....................................................... 121
Configuring the event runtime to store step data in memory ............... 121
Configuring the event runtime to store context data in ObjectGrid ..... 122
Configuring the event runtime to record history .................................... 122
Configuring the durable event queue ....................................................... 123
Configuring the log size of the durable event queue ............................. 124
Configuring how time-delayed event rules and actions are handled ...... 124
Configuring multiple Decision Server Events profiles to coexist ......... 125
Configuring technology connectors on a separate computer ............... 126
Configuring the User Console connector to use a WebSphere Application Server data source ........................................ 126
Tuning your event runtime configuration .............................................. 128
Tuning with the event runtime and WebSphere Application Server .... 129
Tuning with JMS messaging ................................................................. 130
Verifying your Decision Server Events configuration ........................ 131
Securing Decision Server Events ............................................................ 135
User authentication .................................................................................. 135
Configuring user authentication by using WebSphere Application Server ........................................ 136
Configuring user authentication by using LDAP ................................... 137
Configuring user authentication by using Microsoft Windows Active Directory ........................................ 139
Configuring user authentication by using User Console ................. 140
Securing access to charts ....................................................................... 141
Securing the service integration bus ....................................................... 142

Chapter 4. Configuring the Decision Center consoles on WebSphere Application Server

Configuring the Decision Center consoles on WebSphere Application Server for distributed platforms ........................................ 145
Step 4: Packaging Decision Validation
Services by using Ant 218
Step 5: Redeploying the Decision Validation Services archive 219

Chapter 5. Verifying your configuration of Decision Center 221
Publishing a project to Decision Center 221

Opening the Decision Center consoles 222

Notices 225
Trademarks 227

Index 229
Chapter 1. Clustering Operational Decision Manager servers for high availability and scalability

To achieve a highly available and scalable system, you can group a set of WebSphere® Application Server instances in a cluster. Clusters provide workload balancing and failover support for applications that run on WebSphere Application Server, enhancing throughput and availability.

Operational Decision Manager provides scripts that you can run to create and configure Decision Server clusters or Decision Center clusters.

WebSphere Application Server clustering and the gold topology

Planning a clustered topology with WebSphere Application Server Network Deployment is complicated because many combinations are possible but only a subset suits your needs. A gold topology helps you to make the right choices.

A gold topology is the one that is typically configured for a particular usage scenario. It aims to balance availability and scalability requirements, and other desirable system characteristics. It simplifies the planning task by providing a model for you to follow that is based on the experience of knowledgeable Operational Decision Manager users.

There are two gold topologies for WebSphere Application Server cells. The topology based on Operational Decision Manager cells facilitates rule authoring while the topology based on Decision Server cells is intended for the rules and event runtime environments.

Operational Decision Manager cell

Enables all of the applications that Operational Decision Manager provides. It includes a Decision Center cluster and a Decision Server cluster.
Decision Server cell
Contains the subset of Operational Decision Manager components that support the rules and event runtime execution environments and testing capabilities.

You can use these two types of cells to create a shared environment or a staged environment.

Shared environment
A single authoring server that feeds multiple runtime servers, which are specialized and isolated by lifecycle phases. You deploy the executable artifacts to run business rules and events over HTTP and HTTPS.

A shared environment has the following advantages:
- Provides functional isolation between business authoring and execution.
- Offers a single source for authoring and deployment.
- Supports branching and merging so that you can deploy executable rules on selected servers.
- Enables high availability for all Operational Decision Manager capabilities.
- Separates execution and simulation workloads.

A shared environment has the following disadvantages:
- Lacks the ability to customize Decision Center for a single phase (development, test, or production).
- Introduces a single point of failure for authoring because of the shared Decision Center repository. You can consider using a database replication to avoid this weakness.
- Requires management of access to actions and servers.

The following diagram shows how you can use a single Decision Center to propagate decision artifacts to multiple Decision Server instances.
Staged environment

Each cell can be managed independently and includes a Decision Center and a Decision Server cluster.

A staged environment has the following advantages:
- Provides full isolation between development lifecycle stages.
- Isolates authoring and execution by stage and cell.
- Offers the ability to customize Decision Center in each cell, including security.
- Supports high availability for clustered Decision Center and Decision Server instances.

A staged environment has the following disadvantages:
- Requires the provisioning and management of multiple JVMs and Decision Center databases.
- Requires the synchronization of Decision Center repository content across cells from the development cell to the production cell.

You must propagate rule and event projects from one stage to another by using Export and Import commands. You can automate the propagation process by using the Java APIs and Ant tasks provided.

The following diagram shows how the Operational Decision Manager cell is used in each stage in the development lifecycle.
You can configure the Operational Decision Manager cell and the Decision Server cell by using the provided profile template scripts, or manually by following the documented steps.

**Operational Decision Manager cell**

The Operational Decision Manager gold topology cell includes the full functionality of Decision Center and Decision Server.

The Operational Decision Manager cell can be used to stage your authoring, testing, and production environments. If some Decision Server components are not relevant to your particular requirements, you can remove them to reduce resource usage. Each node contains a Decision Server instance and a Decision Center instance.

As shown in the following diagram, the Decision Server instances in the cell form a cluster. The Decision Center instances in the cell form a separate cluster. Members of the same cluster are configured identically to enable workload balancing and failover support.

---

**Decision Center cluster**

This cluster contains all the server-side components of Decision Center for business authoring and simulation management. It also contains an EAR file for the web consoles and an EAR file for the event widgets.
**Decision Server cluster**

The Decision Server cluster includes all of the components for executing business rules and event logic. It also handles the execution of business rule test suites and simulations. This cluster includes the following Java EE artifacts:

- An EAR file for the Rule Execution Server console that is deployed at the node level.
- A RAR file for the rules execution unit that is deployed at the node level.
- An EAR file that contains the Scenario Service Provider, which executes business rule test suites and simulations.
- An EAR file that contains the hosted transparent decision service, which dynamically publishes and runs decision web services.
- An event runtime to execute and test event rules. The event runtime includes runtime and test EAR files. The test EAR file must be on the same cluster as the runtime EAR file.
- Event connectors.
- Message engines that are required for event processing and that are used by message-driven rule beans (MDBs).

Rule and event runtimes are collocated in the same JVMs. To enable a decision service that is triggered by an event detection, you must place the two engines in the same JVM for a local invocation.

**Decision Server gold topology cell**

The Decision Server gold topology cell includes the Decision Server Rules and Decision Server Events runtimes.

You can use the Decision Server cell to host your authoring, testing, and production environments. If some Decision Server components are not relevant to your particular requirements, you can remove them to minimize resource usage.

As shown in the following diagram, each node contains a Decision Server instance. The Decision Server instances in each cell form a cluster. The members of the cluster are configured identically to enable failover support and workload balancing.

**Note:**

- The scope of the Java Management Extensions (JMX) API is typically that of the WebSphere Application Server cell. Therefore, an error might be raised if the management stack finds no Rule Execution Server console or more than one. For more information, see technote Limitations.
- In the following diagram, the Rule Execution Server console is clustered. However, you can also deploy it to a single server in the cell. In this case, for the testing and simulation services to work, you must implement the IlrSSPResolver interface as explained in “Distributing Decision Validation Services tests to multiple servers” on page 183.
The Decision Server cluster includes all the components to execute business rules and event logic. It also covers the execution of business rule test suites and simulations. This cluster includes the following Java EE artifacts:

- An EAR file for the Rule Execution Server console that is deployed at the node level.
- A RAR file for the rules execution unit (XU) that is deployed at the node level.
- An EAR file for the Scenario Service Provider (SSP), which executes business rule test suites and simulations.
- An EAR file that contains the hosted transparent decision service, which dynamically publishes and runs decision web services.
- An event runtime to execute and test event rules. An event runtime includes runtime and test EAR files. The test EAR file must be on the same cluster as the runtime EAR file.
- Event connectors.
- Message engines that are required for event processing and that are used by message-driven rule beans (MDBs).

Rule and event runtimes are collocated in the same Java virtual machine (JVM). To enable a decision service that is triggered by an event detection, you must place the two engines in the same JVM for a local invocation.

**Operational Decision Manager profile templates**

When you install Operational Decision Manager, the installer copies profile templates to the WebSphere Application Server directory for profile templates.

The following table lists the files that are copied to each directory.

<code>&lt;ODM_InstallDir&gt;</code> is the installation directory of Operational Decision Manager
and `<WAS_InstallDir>` is the directory where you installed WebSphere Application Server. If you specify the `<WAS_InstallDir>` in Installation Manager, the profile templates are also copied to the `profileTemplates` directory.

**Important:** If you do not install the product Samples and Tutorials, you must copy the folders from `<ODM_InstallDir>` to `<WAS_InstallDir>`.

### Table 1. Installation of profile template files and directories for Decision Server and Decision Center

<table>
<thead>
<tr>
<th>Location</th>
<th>When</th>
<th>Files</th>
</tr>
</thead>
</table>
| `<ODM_InstallDir>/executionserver/` applicationservers/WebSphere8 | At product installation | • The EAR files for the Rule Execution Server console, Scenario Service Provider (SSP), and hosted transparent decision services  
• The RAR file for the execution unit (XU) |
| `<ODM_InstallDir>/teamserver/` applicationservers/WebSphere8 | At product installation | The Decision Center EAR file. |
| `<ODM_InstallDir>/shared/` profiles/profileTemplates/rules | At product installation | • The Decision Server profile template to augment a deployment manager profile. The `management/ds` subdirectory that contains the scripts to augment a deployment manager profile  
• The Decision Center profile template to augment a deployment manager profile. The `management/dc` subdirectory that contains the scripts to augment a deployment manager profile. |
| `<WAS_InstallDir>/profileTemplates/rules` | At product installation if `<WAS_InstallDir>` is selected in the installer | • The `management/ds` subdirectory that contains the scripts to augment a deployment manager profile.  
• The `management/dc` subdirectory that contains the scripts to augment a deployment manager profile. |
| `<WAS_InstallDir>/profileTemplates/rules/default/ds` | At product installation | The scripts to augment a node or a WebSphere Application Server profile with Decision Server. |
| `<WAS_InstallDir>/profileTemplates/rules/default/dc` | At product installation | The scripts to augment a node or a WebSphere Application Server profile with Decision Center. |

### Table 2. Installation of profile template files and directories for the event runtime

<table>
<thead>
<tr>
<th>Location</th>
<th>When</th>
<th>Files</th>
</tr>
</thead>
</table>
| `<ODM_InstallDir>/runtime` | At product installation | The event runtime EAR files:  
• `wberuntimeear.ear`  
• `WBETesterEar.ear` |
| `<ODM_InstallDir>/runtime` | At product installation | The event widgets EAR file: `EventWidgetsEar.ear` |
| `<WAS_InstallDir>/profileTemplates/wbe/management` | At product installation if `<WAS_InstallDir>` is selected in the installer | The scripts to augment a deployment manager profile with the event runtime. |
| `<WAS_InstallDir>/profileTemplates/wbe/default` | At product installation | The scripts to augment a node or a WebSphere Application Server profile with the event runtime. |
Configuring Rule Execution Server by using profile templates

You can create new profiles for Rule Execution Server and configure a WebSphere Application Server cluster for high availability and scalability.

To create a new stand-alone server to host Rule Execution Server, you must first create a new profile. You can create a profile by using the Profile Management Tool or by using the `manageprofiles` command.

Configuring Rule Execution Server on a new profile

To create a new stand-alone instance of Rule Execution Server environment, you must first create a Decision Server profile. You can create a profile by using the Profile Management Tool or by using the `manageprofiles` command.

**Before you begin**

During profile creation, you are asked to select the database manager that hosts the tables for the execution server run time. Unless you are using Apache Derby, create this database before you configure a new profile. See Creating a Rule Execution Server database.

**About this task**

Depending on your operating system, you can either use the Profile Management Tool or the `manageprofiles` command to create a new profile:

- On a 32-bit operating system, you can use the Profile Management Tool. The Profile Management Tool is not supported on an HP-UX operating system, nor is it on z/OS®.
- On a 32-bit operating system or any other operating system, you can use the `manageprofiles` command.

The procedure describes how to use the `manageprofiles` command to configure Rule Execution Server on a new profile.

**Procedure**

1. Open a command prompt (Windows) or a terminal emulator (Linux and UNIX environments) and navigate to the `<WAS_InstallDir>/bin` directory.
2. Run the `manageprofiles` command by using `manageprofiles.bat` for Windows or `manageprofiles.sh` for Linux and UNIX environments.
   a. Required: Provide the following parameters.
      - `create`  
        To create the new profile.
      - `templatePath <template_path>`  
        where `<template_path>` is the location of the profile template. Provide the path to `<WAS_InstallDir>/profileTemplates/rules/default/ds`.
   b. Optional: Specify the following optional parameters.
      - `dsHome installation location`  
        where `installation location` is the fully qualified path of the Decision Server installation directory `<InstallDir>`. You must specify this parameter if you have installed Decision Server in a non-default location.
- **profileName** profile_name
  where profile_name is the name of the profile.

- **cellName** cell_name
  where cell_name is the name of the cell.

- **nodeName** node_name
  where node_name is the name of the node.

- **hostName** host_name
  where host_name is the host name of the computer that hosts the profile.

- **serverName** server_name
  where server_name is the server name.

- **enableAdminSecurity** true|false
  where true turns on WebSphere Application Server administrative security. If you provide this parameter, you must also provide -adminUserName and -adminPassword. If you do not specify this parameter, the default is false.

- **adminUserName** username
  where username is the user ID to access WebSphere Application Server. Required only if enableAdminSecurity is enabled.

- **adminPassword** password
  where password is the password for the user ID to access WebSphere Application Server. Required only if enableAdminSecurity is enabled.

- **dbType** database_type
  where database_type specifies the database product. Valid values:
  - ```Derby_Embedded``` : the default value
  - ```DB2_Universal``` 
  - ```Derby_NetworkServer``` 
  - ```Oracle``` 
  - ```MS_SQL_Server``` 

- **dbName** database_name
  where database_name is the name of the database. Required unless -dbType is Derby_Embedded.

- **dbUserID** database_user_name
  where database_user_name is the user ID to access the database server. Required unless -dbType is Derby_Embedded.

- **dbPassword** password
  where password is the password for the user ID to access the database server. Required unless -dbType is Derby_Embedded.

- **dbJDBCClasspath** database_jdbc_classpath
  where database_jdbc_classpath specifies the path to the JDBC class path files. Required unless -dbType is Derby_Embedded.

- **dbJDBCLicenseClasspath** database_jdbc_license_classpath
  where database_jdbc_license_classpath specifies the path to the JDBC class path files. Required only if -dbType is DB2_Universal.

- **dbHostName** host_name
  where host_name is the host name for the database server. Required unless -dbType is Derby_Embedded.
-dbServerPort port_number

where port_number is the port of the TCP/IP service, or the port on which the database is listening. Required unless -dbType is Derby_Embedded.

Results

After you run the manageprofiles command, the profile is created.

Examples

The following examples show how to configure Rule Execution Server with the default messaging provider on Windows and Linux.

The first example creates a new application server profile on Windows by using embedded Apache Derby as the database provider and WebSphere Application Server default messaging as the messaging provider.

manageprofiles.bat -create -templatePath "C:\Program Files\IBM\ODM851\WAS\profileTemplates\rules\default\ds"
-dsHome "C:\Program Files\IBM\ODM851"

The second example creates a new application server profile on Linux and UNIX environments by using embedded Apache Derby as the database provider and WebSphere Application Server default messaging as the messaging provider.

manageprofiles.sh -create -templatePath "/opt/ibm/ODM851/ODM/WAS/profileTemplates/rules/default/ds"
-dsHome "/opt/ibm/ODM851/ODM"

Augmenting an existing profile with Rule Execution Server

To create an instance of Rule Execution Server on an existing WebSphere Application Server instance, you can augment the profile by using the manageprofiles command.

Before you begin

Before you augment an existing WebSphere Application Server profile, ensure that all application servers in the profile are stopped.

During profile creation, you are prompted to select the database manager, which hosts the tables for the event run time. If you are not using Apache Derby, create the database before you create a profile. See Creating a Rule Execution Server database.

About this task

You can use the manageprofiles command or the Profile Management Tool to augment an existing profile:

• The manageprofiles command is supported on all operating systems.
• The Profile Management Tool is supported only on 32-bit operating systems. The Profile Management Tool is not supported on HP-UX operating systems, nor is it on z/OS.

The procedure describes how to use the manageprofiles command to augment Rule Execution Server on an existing profile.
**Procedure**

1. Open a command prompt (Windows) or a terminal emulator (Linux and UNIX environments) and navigate to the `<WAS_InstallDir>/bin` directory.

2. Run the `manageprofiles` command. The files are `manageprofiles.bat` for Windows or `manageprofiles.sh` for Linux and UNIX environments.
   a. Required: Provide the following parameters.
      - `augment`
        To augment the existing profile.
      - `unaugment`
        To undo the augmentation of the profile.
      - `templatePath <template_path>`
        where `<template_path>` is the location of the profile template. Provide the following value for the application server profile: `<WAS_InstallDir>/profileTemplates/rules/default/ds`.
      - `profileName profile_name`
        where `profile_name` is the name of the existing application server profile that you are augmenting.
   b. Optional: Specify the following parameters.
      - `dsHome installation location`
        where `installation location` is the fully qualified path that is the installation location for Decision Server. You must specify this parameter if Decision Server is installed in a location other than the default.
      - `profileName profile_name`
        where `profile_name` is the name of the profile.
      - `cellName cell_name`
        where `cell_name` is the name of the cell.
      - `nodeName node_name`
        where `node_name` is the name of the node.
      - `hostName host_name`
        where `host_name` is the host name of the computer that hosts the profile.
      - `serverName server_name`
        where `server_name` is the server name.
      - `enableAdminSecurity true|false`
        where `true` turns on WebSphere Application Server administrative security. If you provide this parameter, you must also provide `-adminUserName` and `-adminPassword`. If you do not specify this parameter, the default is `false`.
      - `adminUserName username`
        where `username` is the user ID to access WebSphere Application Server. Required only if `enableAdminSecurity` is enabled.
      - `adminPassword password`
        where `password` is the password for the user ID to access WebSphere Application Server. Required only if `enableAdminSecurity` is enabled.
      - `dbType database_type`
        where `database_type` specifies the database product. Valid values:
        - `Derby_Embedded`: the default value
-dbName database name
  where database name is the name of the database. Required unless -dbType is Derby_Embedded.

-dbUserId database_user_name
  where database_user_name is the user ID to access the database server. Required unless -dbType is Derby_Embedded.

-dbPassword password
  where password is the password for the user ID to access the database server. Required unless -dbType is Derby_Embedded.

-dbJDBCClasspath database_jdbc_classpath
  where database_jdbc_classpath specifies the path to the JDBC class path files. Required unless -dbType is Derby_Embedded.

-dbJDBClicenseClasspath database_jdbc_license_classpath
  where database_jdbc_license_classpath specifies the path to the JDBC class path files. Required only if -dbType is DB2_Universal.

-dbHostName host_name
  where host_name is the host name for the database server. Required unless -dbType is Derby_Embedded.

-dbServerPort port_number
  where port_number is the port of the TCP/IP service, or the port on which the database is listening. Required unless -dbType is Derby_Embedded.

Results

After you run the manageprofiles command, the profile is augmented.

Examples

The following examples show how to configure Rule Execution Server with the default messaging provider.

The first example augments an existing application server profile, called AppSrv01, on Windows by using embedded Apache Derby as the database provider.

manageprofiles.bat -augment -templatePath "C:\Program Files\IBM\ODM851\WAS\profileTemplates\rules\default\ds" -dsHome "C:\Program Files\IBM\ODM851" -profileName AppSrv01

The second example augments an existing application server profile, called AppSrv01, on Linux and UNIX environments by using embedded Apache Derby as the database provider.

manageprofiles.sh -augment -templatePath "/opt/ibm/ODM851/ODM/WAS/profileTemplates/rules/default/ds" -dsHome /opt/ibm/ODM851/ODM -profileName AppSrv01
The following examples show how to unaugment Rule Execution Server from an existing profile.

The first example unaugments an existing Decision Server application server profile, called ODMSrv01, on Windows.

```
manageprofiles.bat -unaugment -templatePath
"C:\Program Files\IBM\ODM851\WAS\profileTemplates\rules\default\ds"
-profileName ODMSrv01
```

The second example unaugments an existing Decision Server application server profile, called ODMSrv01, on Linux and UNIX.

```
manageprofiles.sh -unaugment -templatePath
"/opt/ibm/ODM851/ODM/WAS/profileTemplates/rules/default/ds"
-profileName ODMSrv01
```

**Augmenting a deployment manager profile with Rule Execution Server**

For high availability and scalability of your applications, you can configure Rule Execution Server on a WebSphere Application Server cluster by running a configuration script. You must first augment a management profile.

**Before you begin**

Before you start the script to augment the profile, you must install the profile templates for WebSphere Application Server, create a deployment manager profile, and make sure that a node exists.

1. Install the profile templates for WebSphere Application Server.
2. Create a deployment manager profile by running the Profile Management Tool, unless one exists already. For more information, see the [WebSphere Application Server Information Center](#). This step creates a new directory, for example `profiles/Dmgr01/bin`.
3. Create a node. A target node is mandatory. The node receives the base configuration, that is, the first cluster member and a server for the Rule Execution Server console. The node must be started and visible by the deployment manager. You can create a default node on another computer, use an existing node, or use the `-createNode` option to create the target node on the computer where the configuration runs.

**About this task**

You augment the deployment manager profile so that you can create additional nodes using the provided template.

**Procedure**

Start the `manageprofiles` file with the `-augment` options.

- On Windows: `<WAS_InstallDir>\bin\manageprofiles.bat -augment
-profileName Dmgr01 -templatePath `<WAS_InstallDir>\profileTemplates\rules\management\ds`
- In UNIX environments: `<WAS_InstallDir>/bin/manageprofiles.sh -augment
-profileName Dmgr01 -templatePath `<WAS_InstallDir>/profileTemplates/rules/management/ds`

where
• `<WAS_InstallDir>` is the directory where you installed WebSphere Application Server.
• Dmgr01 is the name of the profile deployment manager that you created as a prerequisite.

**Results**

The `manageprofiles[.bat|.sh]` command has the following effects:
1. It creates the users `resAdmin`, `resMonitor`, and `resDeployers`.
2. It copies the `configureDSCluster.bat` script for Windows or the `configureDSCluster.sh` script for UNIX platforms to the deployment manager target directory, for example profiles/Dmgr01/bin.

**Running the configureDSCluster script to create a cluster**

After you have augmented the default management profile, you can run the script that configures Rule Execution Server as a WebSphere Application Server cluster.

**Before you begin**

Before you start the script to augment the profile, you must set the `ODM_HOME` environment variable. The configuration script that you run in step 2 on page 15 uses that variable.

**About this task**

After you have augmented the management profile, you set the cluster configuration properties and run the configuration script.

**Note:** The configuration script configures only the specified target node. You can run the script again for each additional node by changing the target node name. Alternatively, you can manually configure the execution unit (XU) and the database for the additional nodes. For help on adding a node to an existing cluster, see “Running the addNodeToDSCluster script” on page 16.

**Procedure**

1. Edit the `configureDSCluster.properties` file to set the cluster name and database values.
   
   You can find this file in the profiles/Dmgr01/bin/rules/ directory, where `Dmgr01` is the name of the deployment manager profile. The configuration properties must contain values for the following keys:

   - `wodm.dsrules.clusterName`
     
     The value can be the name of an existing cluster. If no cluster exists yet, a cluster is created. The default name is `DecisionServerCluster`.

   - `wodm.dsrules.rulesMgrServerName`
     
     The name of the server that hosts the Rule Execution Server console. For example: `RulesMgrSrv`.

   - `wodm.dsrules.db.type`
     
     The database type: DB2®, Oracle, or MSQL. The default is DB2.

   - `wodm.dsrules.db.jdbcDriverPath`
     
     The path to the JDBC drivers. Separate the driver names with a colon (:). For example: `C:/drivers/db2jcc.jar;C:/drivers/db2jcc_license_cu.jar;`
**wodm.dsrules.db.name**

The name of the database. For example: MyDB.

**wodm.dsrules.db.hostname**

The name of the host where the database is hosted: For example: MyDB_server.

**wodm.dsrules.db.port**

The port number to establish the connection to the database.

**wodm.dsrules.db.user**

The user name to connect to the database. For example: db_user1.

**wodm.dsrules.db.password**

The password for the user to connect to the database. For example: db_user1_pwd.

2. Navigate to the profiles/Dmgr01/bin directory and start the `configureDSCluster` script with the required parameters.

The script interpreter calls bash. If bash is not available on your system, an error is raised with the message **Bad interpreter**.

The command line syntax is: `configureDSCluster.[.sh|.bat] -ParameterName1 parameterValue1 -ParameterName2 parameterValue2 ...`. If you make a mistake, such as a missing parameter or a wrong node name, you can execute the script again.

- On Windows:
  
  ```bash
  <WAS_InstallDir>/profiles/Dmgr01/bin/configureDSCluster.bat -dmgrAdminUsername websphere -dmgrAdminPassword websphere -clusterPropertiesFile <WAS_InstallDir>/profiles/Dmgr01/bin/rules/configureDSCluster.properties -createNode -targetNodeName DecisionServerNode01 -dmgrHostName localhost -dmgrPort 8879
  ```

- On UNIX:
  
  ```bash
  <WAS_InstallDir>/profiles/Dmgr01/bin/configureDSCluster.sh -dmgrAdminUsername websphere -dmgrAdminPassword websphere -clusterPropertiesFile <WAS_InstallDir>/profiles/Dmgr01/bin/rules/configureDSCluster.properties -createNode -targetNodeName DecisionServerNode01 -dmgrHostName localhost -dmgrPort 8879
  ```

---

### Table 3. Cluster configuration parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-dmgrAdminUsername</td>
<td>Mandatory</td>
<td>The WebSphere Application Server administrator's user identifier.</td>
</tr>
<tr>
<td>-dmgrAdminPassword</td>
<td>Mandatory</td>
<td>The WebSphere Application Server administrator’s password.</td>
</tr>
<tr>
<td>-clusterPropertiesFile</td>
<td>Mandatory</td>
<td>The path to the properties file.</td>
</tr>
<tr>
<td>-uninstall</td>
<td>Optional</td>
<td>Removes all the applications and resources from the cluster. The cluster itself is not removed because it might contain other applications. See <a href="#">Uninstalling Rule Execution Server from a cluster</a> on page 19.</td>
</tr>
<tr>
<td>-createNode</td>
<td>Optional</td>
<td>If you specify this parameter, the script creates a default node profile with the name set in the <code>-targetNodeName</code> parameter.</td>
</tr>
<tr>
<td>-targetNodeName</td>
<td>Optional</td>
<td>By default, the name of the target installation node is set to RulesNode01.</td>
</tr>
</tbody>
</table>
Table 3. Cluster configuration parameters (continued)

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-dmgrHostName</td>
<td>Optional</td>
<td>Use this parameter to specify the name of the host on which the deployment manager is located. This parameter is mandatory if you create the target node by setting the <code>-targetNodeName</code> parameter.</td>
</tr>
<tr>
<td>-dmgrPort</td>
<td>Optional</td>
<td>The SOAP TCP port of the deployment manager. By default, 8879.</td>
</tr>
</tbody>
</table>

Results

The script performs the following actions:

- Installs the JDBC provider, JCA connector, and data source at node level.
- Installs the Rule Execution Server console to the cluster.
- Deploys the hosted transparent decision services and Scenario Service Provider (SSP) to the cluster member. Users are mapped to application groups when an application is deployed.
- Starts the deployment manager server if it is not already started.
- Configures security.
- Creates the resAdmin, resDeployer, resMonitor users. These users belong to the WebSphere Application Server monitor so that JMX notifications work on all cluster nodes.
- Configures users and groups.
- Installs the execution unit (XU).
- Maps users and groups to roles.
- Starts the cluster, servers, and applications.

When the script completes, the cluster is up and running. The user should then configure the load balancing system, such as IBM HTTP server.

Running the addNodeToDSCluster script

After you have configured Rule Execution Server as a WebSphere Application Server cluster, you can run the script that adds a node to the cluster.

Before you begin

Before you start the script to add a cluster node, you must set the ODM_HOME environment variable. The script that you run in step 2 on page 17 uses that variable.

About this task

After you have set up your cluster through the configuration script, you can configure a default node and add it to the cluster.

Procedure

1. Verify that the configureDSCluster.properties file contains the correct cluster name and database values.
   You can find this file in the profiles/Dmgr01/bin/rules/ directory, where Dmgr01 is the name of the deployment manager profile. The configuration properties must contain values for the following keys:
**wodm.dsrules.clusterName**
The value must be the name of an existing cluster. The default name is DecisionServerCluster.

**wodm.dsrules.rulesMgrServerName**
The name of the server that hosts the Rule Execution Server console. For example: RulesMgrSrv.

**wodm.dsrules.db.type**
The database type: DB2, Oracle, or MSQL. The default is DB2.

**wodm.dsrules.db.jdbcDriverPath**
The path to the JDBC drivers. Separate the driver names with a colon (;). For example: C:/drivers/db2jcc.jar;C:/drivers/db2jcc_license_cu.jar;

**wodm.dsrules.db.name**
The name of the database. For example: MyDB.

**wodm.dsrules.db.hostname**
The name of the host where the database is hosted. For example: MyDB_server.

**wodm.dsrules.db.port**
The port number to establish the connection to the database.

**wodm.dsrules.db.user**
The user name to connect to the database. For example: db_user1.

**wodm.dsrules.db.password**
The password for the user to connect to the database. For example: db_user1_pwd.

2. Navigate to the profiles/Dmgr01/bin directory and start the addNodeToDSCluster script with the required parameters.

   The script interpreter calls bash. If bash is not available on your system, an error is raised with the message Bad interpreter.

   The command line syntax is: addNodeToDSCluster[.sh|.bat] -ParameterName1 parameterValue1 -ParameterName2 parameterValue2 ... If you make a mistake, such as a missing parameter or a wrong target node name, you can execute the script again.

   - On Windows:
     <WAS_InstallDir>/profiles/Dmgr01/bin/addNodeToDSCluster.bat -dmgrAdminUsername websphere -dmgrAdminPassword websphere -clusterPropertiesFile <WAS_InstallDir>/profiles/Dmgr01/bin/rules/configureDSCluster.properties -targetNodeName DecisionServerNode01 -dmgrPort 8879
   - On UNIX:
     <WAS_InstallDir>/profiles/Dmgr01/bin/addNodeToDSCluster.sh -dmgrAdminUsername websphere -dmgrAdminPassword websphere -clusterPropertiesFile <WAS_InstallDir>/profiles/Dmgr01/bin/rules/configureDSCluster.properties -targetNodeName DecisionServerNode01 -dmgrPort 8879

**Table 4. Cluster configuration parameters**

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-dmgrAdminUsername</td>
<td>Mandatory</td>
<td>The WebSphere Application Server administrator’s user identifier</td>
</tr>
<tr>
<td>-dmgrAdminPassword</td>
<td>Mandatory</td>
<td>The WebSphere Application Server administrator’s password</td>
</tr>
<tr>
<td>-clusterPropertiesFile</td>
<td>Mandatory</td>
<td>The path to the properties file.</td>
</tr>
</tbody>
</table>
Table 4. Cluster configuration parameters (continued)

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-targetNodeName</td>
<td>Optional</td>
<td>By default, the name of the target installation node is set to RulesNode01.</td>
</tr>
<tr>
<td>-dmgrPort</td>
<td>Optional</td>
<td>The SOAP TCP port of the deployment manager. By default, 8879.</td>
</tr>
</tbody>
</table>

Results

The script configures a default WebSphere Application Server node, and adds it to the cluster that is specified in the configureDSCluster.properties file that you passed as a parameter to the script.

When the script completes, the cluster is up and running with the new node.

Unaugmenting Rule Execution Server from existing profiles

You can unaugment Decision Server from existing profiles without deleting the cluster.

About this task

Use the manageprofiles command to unaugment Decision Server from existing profiles in a cluster.

Procedure

Start the manageprofiles file with the following -unaugment options:

- On Windows: `<WAS_InstallDir>\bin\manageprofiles.bat -unaugment -profileName Dmgr01 -templatePath `<WAS_InstallDir>`\profileTemplates\rules\management\ds -nodeName DecisionNode -cellName DecisionCell -targetNodeName DecisionServerNode01
- In UNIX environments: `<WAS_InstallDir>/bin/manageprofiles.sh -unaugment -profileName Dmgr01 -templatePath `<WAS_InstallDir>`/profileTemplates/ rules\management\ds -nodeName DecisionNode -cellName DecisionCell -targetNodeName DecisionServerNode01

where

- `<WAS_InstallDir>` is the directory where you installed WebSphere Application Server.
- Dmgr01 is the name of the profile deployment manager that you created as a prerequisite to configuring the cluster.
- ds is the abbreviation of Decision Server.
- -nodeName and -cellName are the names of the node and cell that you defined when you created the deployment manager profile.
- -targetNodeName is the name of the node to which you installed Decision Server.

Results

When the process completes, a message similar to the following one is displayed:

INSTCONFSUCCESS: Profile unaugmentation succeeded
The command removes the resAdmin, resDeployer, and resMonitor users. The cluster is kept as is to prevent any risk of deleting previously existing configurations.

**Uninstalling Rule Execution Server from a cluster**

You can uninstall Rule Execution Server from an existing cluster.

**About this task**

To uninstall Rule Execution Server from a cluster, you use the same configuration script as for configuring the cluster.

**Procedure**

1. Navigate to the profiles/&lt;profile_name&gt;/bin directory.
2. Start the configureDSCluster[.sh|.bat] script with the -uninstall argument.
   
   The command line is:
   ```
   configureDSCluster[.sh|.bat] -dmgrAdminUsername username -dmgrAdminPassword password -clusterPropertiesFile filePath -uninstall
   ```
   - On Windows:
     ```
     &lt;WAS_InstallDir&gt;\profiles\Dmgr01\bin\configureDSCluster.bat -dmgrAdminUsername websphere -dmgrAdminPassword websphere -clusterPropertiesFile &lt;WAS_InstallDir&gt;\profiles\Dmgr01\bin\rules\configureDSCluster.properties -targetNodeName DecisionServerNode01 -dmgrHostName localhost -dmgrPort 8879 -uninstall
     ```
   - On UNIX:
     ```
     &lt;WAS_InstallDir&gt;/profiles/Dmgr01/bin/configureDSCluster.sh -dmgrAdminUsername websphere -dmgrAdminPassword websphere -clusterPropertiesFile &lt;WAS_InstallDir&gt;/profiles/Dmgr01/bin/rules/configureDSCluster.properties -targetNodeName DecisionServerNode01 -dmgrHostName localhost -dmgrPort 8879 -uninstall
     ```

**Table 5. Cluster configuration parameters**

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-dmgrAdminUsername</td>
<td>Mandatory</td>
<td>The WebSphere Application Server administrator’s user identifier that you defined when you installed Decision Server.</td>
</tr>
<tr>
<td>-dmgrAdminPassword</td>
<td>Mandatory</td>
<td>The WebSphere Application Server administrator’s password that you defined when you installed Decision Server.</td>
</tr>
<tr>
<td>-clusterPropertiesFile</td>
<td>Mandatory</td>
<td>The full path to the properties file.</td>
</tr>
<tr>
<td>-uninstall</td>
<td>Mandatory</td>
<td>Removes all the applications and resources from the cluster. The cluster itself is not removed because it might contain other applications.</td>
</tr>
<tr>
<td>-createNode</td>
<td>Optional</td>
<td>If you specify this parameter, the script creates a default node profile with the name set in the -targetNodeName parameter.</td>
</tr>
<tr>
<td>-targetNodeName</td>
<td>Optional</td>
<td>By default, the name of the target installation node is set to RulesNode01.</td>
</tr>
<tr>
<td>-dmgrHostName</td>
<td>Optional</td>
<td>Use this parameter to specify the name of the host on which the deployment manager is located. This parameter is mandatory if you create the target node by setting the -targetNodeName parameter.</td>
</tr>
<tr>
<td>Parameter name</td>
<td>Mandatory/Optional</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>-dmgrPort</td>
<td>Optional</td>
<td>The SOAP TCP port of the deployment manager. By default, 8879.</td>
</tr>
</tbody>
</table>

**Results**

This command has the following results:

- Uninstalls hosted transparent decision services and Scenario Service Provider (SSP).
- Uninstalls the Rule Execution Server console.
- Removes the Rule Execution Server console server.
- Uninstalls the execution unit (XU).
- Uninstalls the data source.
- Removes the administrative and applicative groups.

**Configuring the Decision Center consoles by using profile templates**

You can create new profiles for the Decision Center consoles and configure a WebSphere Application Server cluster for high availability and scalability.

To create a new stand-alone server to host Decision Center, you must first create a new profile. You can create a profile by using the Profile Management Tool or the `manageprofiles` command.

**Configuring Decision Center on a new profile**

To create a new stand-alone instance of Decision Center, you must first create a Decision Center profile. You can create a profile by using the Profile Management Tool, or by using the `manageprofiles` command.

**Before you begin**

During profile creation, you are asked to select the database manager that hosts the tables for the execution server run time. Unless you are using Apache Derby, create this database before you configure a new profile. See [Creating a data source and connection pool](#).

**About this task**

Depending on your operating system, you can either use the Profile Management Tool or the `manageprofiles` command to create a new profile:

- On a 32-bit operating system, you can use the Profile Management Tool. The Profile Management Tool is not supported on an HP-UX operating system, nor is it on z/OS.
- On a 32-bit operating system or any other operating system, you can use the `manageprofiles` command.

The following procedure describes how to use the `manageprofiles` command to configure Decision Center on a new profile.
Procedure

1. Open a command prompt (Windows) or a terminal emulator (Linux and UNIX environments) and navigate to the `<WAS_InstallDir>/bin` directory.

2. Run the `manageprofiles` command by using `manageprofiles.bat` for Windows or `manageprofiles.sh` for Linux and UNIX environments. You must provide the following parameters:

   - **-create**
     To create the new profile.

   - **-templatePath <template_path>**
     where `<template_path>` is the location of the profile template. Provide the path to `<WAS_InstallDir>/profileTemplates/rules/default/dc`.

   Use the following optional parameters:

   - **-dsHome installation location**
     where `installation location` is the fully-qualified path of the Decision Center installation directory `<InstallDir>`. You must specify this parameter if you have installed Decision Center in a non-default location.

   - **-profileName profile_name**
     where `profile_name` is the name of the profile.

   - **-cellName cell_name**
     where `cell_name` is the name of the cell.

   - **-nodeName node_name**
     where `node_name` is the name of the node.

   - **-nodeName host_name**
     where `host_name` is the host name of the computer hosting the profile.

   - **-serverName server_name**
     where `server_name` is the server name.

   - **-enableAdminSecurity true|false**
     where `true` turns on WebSphere Application Server administrative security. If you provide this parameter, you must also provide `-adminUserName` and `-adminPassword`. The default is false if you do not specify this parameter, and security will not be active. You must also provide `-adminUserName` and `-adminPassword`.

   - **-adminUserName username**
     where `username` is the user ID to access WebSphere Application Server. Only required if `enableAdminSecurity` is enabled.

   - **-adminPassword password**
     where `password` is the password for the user ID to access WebSphere Application Server. Only required if `enableAdminSecurity` is enabled.

   - **-dbType database_type**
     where `database_type` specifies the database product. Valid values are:
     
     - ` Derby_Embedded`: the default value
     - ` DB2_Universal`
     - ` Derby_NetworkServer`
     - ` Oracle`
     - ` MS_SQL_Server`
-dbName  database_name
  where database_name is the name of the database. Required unless -dbType is Derby_Embedded.

-dbUserId  database_user_name
  where database_user_name is the user ID to access the database server. Required unless -dbType is Derby_Embedded.

-dbPassword  password
  where password is the password for the user ID to access the database server. Required unless -dbType is Derby_Embedded.

-dbJDBCClasspath  database_jdbc_classpath
  where database_jdbc_classpath specifies the path to the JDBC class path files. Required unless -dbType is Derby_Embedded.

-dbJDBCLicenseClasspath  database_jdbc_license_classpath
  where database_jdbc_license_classpath specifies the path to the JDBC class path files. Only required if -dbType is DB2_Universal.

-dbHostName  host_name
  where host_name is the host name for the database server. Required unless -dbType is Derby_Embedded.

-dbServerPort  port_number
  where port_number is the port of the TCP/IP service, or the port on which the database is listening. Required unless -dbType is Derby_Embedded.

Results

After running the manageprofiles command, the profile is created.

Examples

The following examples show how to configure Decision Center with the default messaging provider on Windows and Linux.

The first example creates a new application server profile on Windows by using embedded Apache Derby as the database provider and WebSphere Application Server default messaging as the messaging provider.

manageprofiles.bat -create
  -templatePath "C:\Program Files\IBM\ODM851\WAS\profileTemplates\rules\default\dc"
  -dcHome "C:\Program Files\IBM\ODM851"

The second example creates a new application server profile on Linux and UNIX environments by using embedded Apache Derby as the database provider and WebSphere Application Server default messaging as the messaging provider.

manageprofiles.sh -create
  -templatePath "/opt/ibm/ODM851/ODM/WAS/profileTemplates/rules/default/dc"
  -dcHome "/opt/ibm/ODM851/ODM"

Augmenting an existing profile with Decision Center

To create an instance of Decision Center on an existing WebSphere Application Server, you can augment the profile by using the manageprofiles command.
Before you begin

Before you augment an existing WebSphere Application Server application server profile, ensure that all application servers in the profile are stopped.

During profile creation, you are asked to select the database manager, which hosts the tables for the event runtime. If you are not using Apache Derby, create the database before you create a profile. See [Creating a data source and connection pool](#).

About this task

You can use the `manageprofiles` command or the Profile Management Tool to augment an existing profile:

- The `manageprofiles` command is supported on all operating systems.
- The Profile Management Tool is only supported on 32-bit operating systems. The Profile Management Tool is not supported on an HP-UX operating system, nor is it on z/OS.

The procedure describes how to use the `manageprofiles` command to augment Decision Center on an existing profile.

Procedure

1. Open a command prompt (Windows) or a terminal emulator (Linux and UNIX environments) and navigate to the `<WAS_InstallDir>/bin` directory.
2. Run the `manageprofiles` command. The file is `manageprofiles.bat` for Windows or `manageprofiles.sh` for Linux and UNIX environments. Provide the following parameters:
   - `-augment` To augment the existing profile.
   - `-unaugment` To undo the augmentation of the profile.
   - `-templatePath <template_path>` where `<template_path>` is the location of the profile template. Provide the following value for the application server profile: `<WAS_InstallDir>/profileTemplates/rules/default/dc`.
   - `-profileName profile_name` where `profile_name` is the name of the existing application server profile that you are augmenting.

   Optionally, you can specify the following parameters:

   - `-dsHome installation location` where `installation location` is the fully qualified path that is the installation location for Decision Center. You must specify this parameter if you have installed Decision Center in a location different to the default.
   - `-profileName profile_name` where `profile_name` is the name of the profile.
   - `-nodeName node_name` where `node_name` is the name of the node.
-**hostName** *host_name*
  where *host_name* is the host name of the computer that hosts the profile.

-**serverName** *server_name*
  where *server_name* is the server name.

-**enableAdminSecurity** *true|false*
  where *true* turns on WebSphere Application Server administrative security. If you provide this parameter, you must also provide -**adminUserName** and -**adminPassword**. The default is *false* if you do not specify this parameter.

-**adminUserName** *username*
  where *username* is the user ID to access WebSphere Application Server. Only required if **enableAdminSecurity** is enabled.

-**adminPassword** *password*
  where *password* is the password for the user ID to access WebSphere Application Server. Only required if **enableAdminSecurity** is enabled.

-**dbType** *database_type*
  where *database_type* specifies the database product. Valid values are:
  - *Derby_Embedded*: the default value
  - *DB2_Universal*
  - *Derby_NetworkServer*
  - *Oracle*
  - *MS_SQL_Server*

-**dbName** *database_name*
  where *database_name* is the name of the database. Required unless **dbType** is *Derby_Embedded*.

-**dbUserId** *database_user_name*
  where *database_user_name* is the user ID to access the database server. Required unless **dbType** is *Derby_Embedded*.

-**dbPassword** *password*
  where *password* is the password for the user ID to access the database server. Required unless **dbType** is *Derby_Embedded*.

-**dbJDBCClasspath** *database_jdbc_classpath*
  where *database_jdbc_classpath* specifies the path to the JDBC class path files. Required unless **dbType** is *Derby_Embedded*.

-**dbJDBCLicenseClasspath** *database_jdbc_license_classpath*
  where *database_jdbc_license_classpath* specifies the path to the JDBC class path files. Required only if **dbType** is *DB2_Universal*.

-**dbHostName** *host_name*
  where *host_name* is the host name for the database server. Required unless **dbType** is *Derby_Embedded*.

-**dbServerPort** *port_number*
  where *port_number* is the port of the TCP/IP service, or the port on which the database is listening. Required unless **dbType** is *Derby_Embedded*.

**Results**

After the **manageprofiles** command has completed, the profile is augmented.
Examples

The following examples show how to configure Rule Execution Server with the default messaging provider on Windows and Linux.

The first example augments an existing application server profile called AppSrv01 on Windows by using embedded Apache Derby as the database provider.

```bash
manageprofiles.bat -augment
-templatePath "C:\Program Files\IBM\OMNI851\WAS\profileTemplates\rules\default\dc"
-dcHome "C:\Program Files\IBM\OMNI851"
-profileName AppSrv01
```

The second example augments an existing application server profile called AppSrv01 on Linux and UNIX environments by using embedded Apache Derby as the database provider.

```bash
manageprofiles.sh -augment
-templatePath "/opt/ibm/OMNI851/OMNI/WAS/profileTemplates/rules/default/dc" -dcHome /opt/ibm/OMNI851/OMNI -profileName AppSrv01
```

The following examples show how to unaugment Decision Center from an existing profile.

The first example unaugments an existing Decision Center application server profile called WODM5rv01 on Windows.

```bash
manageprofiles.bat -unaugment -templatePath "C:\Program Files\IBM\OMNI851\WAS\profileTemplates\rules\default\dc"
-profileName WODM5rv01
```

The second example unaugments an existing Decision Center application server profile called WODM5rv01 on Linux and UNIX.

```bash
manageprofiles.sh -unaugment -templatePath "/opt/ibm/OMNI851/OMNI/WAS/profileTemplates/rules/default/dc"
-profileName WODM5rv01
```

Augmenting a deployment manager profile with Decision Center

For high availability and scalability of your applications, you can configure Decision Center on a WebSphere Application Server cluster by running a configuration script. You must first augment a management profile.

Before you begin

Before you start the script to augment the profile, you must install Profile templates for WebSphere Application Server, create a default deployment manager profile, and make sure that a node exists:

1. Install Decision Center, which includes WebSphere Application Server and its profile templates.
2. Create a default deployment manager profile by running the Profile Management Tool, unless one exists already. For more information, see the WebSphere Application Server Information Center. This step creates a new directory, for example profiles/0mgr01/bin.
3. Create a node. A target node is mandatory. The node receives the base configuration. The node must be started and visible by the deployment
manager. You can create a default node on another computer, use an existing node, or use the -createNode option to create the target node on the computer where the configuration is run.

**About this task**

You augment the deployment manager profile so that you can create additional nodes using the provided template.

**Procedure**

Launch the manageprofiles file with the -augment options.

- On Windows: `<WAS_InstallDir>\bin\manageprofiles.bat -augment
  -profileName Dmgr01 -templatePath <WAS_InstallDir>\profileTemplates\rules\management\dc`
- In UNIX environments: `<WAS_InstallDir>/bin/manageprofiles.sh -augment
  -profileName Dmgr01 -templatePath <WAS_InstallDir>/profileTemplates/rules/management/dc`

where
- `<WAS_InstallDir>` is the directory where you installed WebSphere Application Server.
- Dmgr01 is the name of the profile deployment manager that you created as a prerequisite.

**Results**

The manageprofiles[.bat|.sh] command has the following effects:
1. It creates the users rtsAdmin, rtsUser1, and rtsConfig.
2. It copies the configureDCCluster.bat script for Windows or configureDCCluster.sh script for UNIX platforms to the deployment manager target directory, for example profiles/Dmgr01/bin.

**Running the configureDCCluster script to create a cluster**

After you have augmented the default management profile, you can run the script that configures Decision Center on a WebSphere Application Server cluster.

**Before you begin**

Before you start the script to augment the profile, you must set the **ODM_HOME** environment variable. The configuration script that you run in step 2 on page 27 uses that variable.

**About this task**

After you have augmented the management profile, you set the cluster configuration properties, and run the configuration script.

**Note:** The configuration script only configures the specified target node. You can run the script again for each additional node by changing the target node name. Alternatively, you can manually configure the execution unit (XU) and the datasource for the additional nodes. For help on adding a node to an existing cluster, see “Running the addNodeToDCCluster script” on page 28.
Procedure

1. Edit the `configureDCCluster.properties` file to set the cluster name and database values.

   You can find this file in the `profiles/Dmgr01/bin/rules/` directory, where `Dmgr01` is the name of the deployment manager profile. The configuration properties must contain values for the following keys:

   - `wodm.dcrules.clusterName`
     The value can be the name of an existing cluster. If no cluster exists yet, a cluster is created. The default name is DecisionCenterCluster.

   - `wodm.dcrules.db.type`
     The database type: DB2, Oracle, or MSSQL. The default is DB2.

   - `wodm.dcrules.db.jdbcDriverPath`
     The path to the JDBC drivers. Separate the driver names with a colon (;). For example: `C:/drivers/db2jcc.jar;C:/drivers/db2jcc_license_cu.jar;`

   - `wodm.dcrules.db.name`
     The name of the database. For example: `MyDB`.

   - `wodm.dcrules.db.hostname`
     The name of the host where the database is hosted: For example: `MyDB_server`.

   - `wodm.dcrules.db.port`
     The port number to establish the connection to the database.

   - `wodm.dcrules.db.user`
     The user name to connect to the database. For example: `db_user1`.

   - `wodm.dcrules.db.password`
     The password for the user to connect to the database. For example: `db_user1_pwd`.

2. Navigate to the `profiles/Dmgr01/bin` directory and start the `configureDCCluster` script with the required parameters.

   The script interpreter calls bash. If bash is not available on your system, an error is thrown with the message `Bad interpreter`.

   The command line syntax is: `configureDCCluster[.sh|.bat] -ParameterName1 parameterValue1 -ParameterName2 parameterValue2 ...`. If you make a mistake, such as a missing parameter or a wrong node name, you can execute the script again.

   - On Windows:
     ```bash
     <WAS_InstallDir>/profiles/Dmgr01/bin/configureDCCluster.bat
     -dmgrAdminUsername websphere -dmgrAdminPassword websphere
     -clusterPropertiesFile
     -<WAS_InstallDir>/profiles/Dmgr01/bin/rules/configureDCCluster.properties
     -createNode -targetNodeName DecisionCenterNode01
     -dmgrHostName localhost -dmgrPort 8879
     ```

   - On UNIX:
     ```bash
     <WAS_InstallDir>/profiles/Dmgr01/bin/configureDCCluster.sh
     -dmgrAdminUsername websphere -dmgrAdminPassword websphere
     -clusterPropertiesFile
     -<WAS_InstallDir>/profiles/Dmgr01/bin/rules/configureDCCluster.properties
     -createNode -targetNodeName DecisionCenterNode01
     -dmgrHostName localhost -dmgrPort 8879
     ```
Table 6. Cluster configuration parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-dmgrAdminUsername</td>
<td>Mandatory</td>
<td>The WebSphere Application Server administrator’s user identifier that you defined when you installed Decision Center.</td>
</tr>
<tr>
<td>-dmgrAdminPassword</td>
<td>Mandatory</td>
<td>The WebSphere Application Server administrator’s password that you defined when you installed Decision Center.</td>
</tr>
<tr>
<td>-clusterPropertiesFile</td>
<td>Mandatory</td>
<td>The full path to the properties file.</td>
</tr>
<tr>
<td>-uninstall</td>
<td>Optional</td>
<td>Removes all the applications and resources from the cluster. The cluster itself is not removed because it might contain other applications. See &quot;Uninstalling Decision Center from a cluster&quot; on page 31.</td>
</tr>
<tr>
<td>-createNode</td>
<td>Optional</td>
<td>If you specify this parameter, the script creates a default node profile with the name set in the -targetNodeName parameter.</td>
</tr>
<tr>
<td>-targetNodeName</td>
<td>Optional</td>
<td>By default, the name of the target installation node is set to RulesNode01.</td>
</tr>
<tr>
<td>-dmgrHostName</td>
<td>Optional</td>
<td>Use this parameter to specify the name of the host on which the deployment manager is located. This parameter is mandatory if you create the target node by setting the -targetNodeName parameter.</td>
</tr>
<tr>
<td>-dmgrPort</td>
<td>Optional</td>
<td>The SOAP TCP port of the deployment manager. By default, 8879.</td>
</tr>
</tbody>
</table>

Results

The script performs the following actions:
- Installs the JDBC provider and the data source at node level.
- Installs the Decision Center application at the cluster level. Users are mapped to application groups when an application is deployed.
- Starts the deployment manager server if it is not already started.
- Configures security.
- Creates the rtsAdmin, rtsInstaller, rtsUser1, and rtsConfig users.
- Configures users and groups.
- Maps users and groups to roles.
- Starts the cluster, servers, and applications.

When the script completes, the cluster is up and running. The user should then configure the load balancing system, such as IBM® HTTP Server (IHS).

Note: Depending on your hardware configuration, and because the Decision Center EAR file is very large, you might need to connect to the Decision Center administration console to verify whether the cluster is correctly started and, if it is not, start it manually.

Running the addNodeToDCCluster script

After you have configured Decision Center on a WebSphere Application Server cluster, you can run the script that adds a node to the cluster.
Before you begin

Before you start the script to add a cluster node, you must set the `ODM_HOME` environment variable. The script that you run in step 2 uses that variable.

About this task

After you have set up your cluster through the configuration script, you can configure a default node and add it to the cluster.

Procedure

1. Verify that the `configureDCCluster.properties` file contains the correct cluster name and database values.
   
   You can find this file in the `profiles/Dmgr01/bin/rules/` directory, where `Dmgr01` is the name of the deployment manager profile. The configuration properties must contain values for the following keys:
   
   - `wodm.dcrules.clusterName`
     - The value must be the name of an existing cluster. The default name is `DecisionCenterCluster`
   
   - `wodm.dcrules.db.type`
     - The database type: DB2, Oracle, or MSSQL. The default is DB2.
   
   - `wodm.dcrules.db.jdbcDriverPath`
     - The path to the JDBC drivers. Separate the driver names with a colon (;).
       
       For example: `C:/drivers/db2jcc.jar;C:/drivers/db2jcc_license_cu.jar;
   
   - `wodm.dcrules.db.name`
     - The name of the database. For example: `MyDB`
   
   - `wodm.dcrules.db.hostname`
     - The name of the host where the database is hosted: For example: `MyDB_server`
   
   - `wodm.dcrules.db.port`
     - The port number to establish the connection to the database
   
   - `wodm.dcrules.db.user`
     - The user name to connect to the database. For example: `db_user1`
   
   - `wodm.dcrules.db.password`
     - The password for the user to connect to the database. For example: `db_user1_pwd`

2. Navigate to the `profiles/Dmgr01/bin` directory and start the `addNodeToDCCluster` script with the required parameters.
   
   The script interpreter calls bash. If bash is not available on your system, an error is thrown with the message `Bad interpreter`.

   The command line syntax is: `addNodeToDCCluster[.sh|.bat] -ParameterName1 parameterValue1 -ParameterName2 parameterValue2 ...`. If you make a mistake, such as a missing parameter or a wrong target node name, you can execute the script again.

   - On Windows:
     
     `<WAS_InstallDir>\profiles\Dmgr01\bin\addNodeToDCCluster.bat -dmgrAdminUsername websphere -dmgrAdminPassword websphere -clusterPropertiesFile <WAS_InstallDir>\profiles\Dmgr01\bin\rules\configureDCCluster.properties -targetNodeName DecisionCenterNode01 -dmgrPort 8879`
On UNIX:

```bash
<WAS_InstallDir>/profiles/Dmgr01/bin/addNodeToDCCluster.sh
-dmgrAdminUsername websphere -dmgrAdminPassword websphere
-clusterPropertiesFile
<WAS_InstallDir>/profiles/Dmgr01/bin/rules/configureDCCluster.properties
-targetNodeName DecisionCenterNode01
-dmgrPort 8879
```

![Table 7. Cluster configuration parameters](image)

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-dmgrAdminUsername</td>
<td>Mandatory</td>
<td>The WebSphere Application Server administrator’s user identifier that you defined when you installed Decision Center</td>
</tr>
<tr>
<td>-dmgrAdminPassword</td>
<td>Mandatory</td>
<td>The WebSphere Application Server administrator’s password that you defined when you installed Decision Center.</td>
</tr>
<tr>
<td>-clusterPropertiesFile</td>
<td>Mandatory</td>
<td>The full path to the properties file.</td>
</tr>
<tr>
<td>-targetNodeName</td>
<td>Optional</td>
<td>By default, the name of the target installation node is set to RulesNode01.</td>
</tr>
<tr>
<td>-dmgrPort</td>
<td>Optional</td>
<td>The SOAP TCP port of the deployment manager. By default, 8879.</td>
</tr>
</tbody>
</table>

**Results**

The script configures a default WebSphere Application Server node, and adds it to the cluster that is specified in the `configureDCCluster.properties` file that you passed as a parameter to the script.

When the script completes, the cluster is up and running with the new node added.

**Note:** Depending on your hardware configuration, and because the Decision Center EAR file is very large, you might need to connect to the Decision Center administration console to verify whether the node is correctly added to the cluster, and if the cluster is correctly started and, if it is not, start it manually.

**Unaugmenting Decision Center from an existing cluster**

You can unaugment Decision Center from existing profiles without deleting the cluster.

**About this task**

Use the `manageprofiles` command to unaugment Decision Center from existing profiles in a cluster.

**Procedure**

Launch the `manageprofiles` file with the following `-unaugment` options:

- On Windows: `<WAS_InstallDir>/bin\manageprofiles.bat -unaugment -profileName Dmgr01 -templatePath <WAS_InstallDir>/profileTemplates\rules\management\dc -nodeName DecisionNode -cellName DecisionCell -targetNodeName DecisionCenterNode01`
In UNIX environments: `<WAS_InstallDir>/bin/manageprofiles.sh -unaugment -profileName Dmgr01 -templatePath `<WAS_InstallDir>/profileTemplates/rules/management/dc -nodeName DecisionNode -cellName DecisionCell -targetNodeName DecisionCenterNode01

where

- `<WAS_InstallDir>` is the directory where you installed WebSphere Application Server.
- Dmgr01 is the name of the profile deployment manager that you created as a prerequisite to configuring the cluster.
- dc is the abbreviation of Decision Center.
- -nodeName and -cellName are the names of the node and cell that you defined when you created the deployment manager profile.
- -targetNodeName is the name of the node to which you installed Decision Center.

Results

The command removes the rtsAdmin, rtsInstaller, rtsUser1, and rtsConfig users. The cluster is kept as is to prevent any risk of deleting previously existing configurations.

Uninstalling Decision Center from a cluster

You can uninstall Decision Center from an existing cluster.

About this task

To uninstall Decision Center from a cluster, you use the same configuration script that you use to configure the cluster.

Procedure

1. Navigate to the profiles/`<profile_name>/bin` directory.
2. Launch the configureDCCluster[.sh|.bat] script with the -uninstall argument.

   The command line is:
   
   ```
   configureDCCluster[.sh|.bat] -dmgrAdminUsername username -dmgrAdminPassword password -clusterPropertiesFile filePath -uninstall
   ```

   - On Windows:
     ```
     <WAS_InstallDir>/profiles/Dmgr01/bin/configureDCCluster.bat -dmgrAdminUsername websphere -dmgrAdminPassword websphere -clusterPropertiesFile <WAS_InstallDir>/profiles/Dmgr01/bin/rules/configureDCCluster.properties -targetNodeName DecisionCenterNode01 -dmgrHostName localhost -dmgrPort 8879 -uninstall
     ```
   - On UNIX:
     ```
     <WAS_InstallDir>/profiles/Dmgr01/bin/configureDCCluster.sh -dmgrAdminUsername websphere -dmgrAdminPassword websphere -clusterPropertiesFile <WAS_InstallDir>/profiles/Dmgr01/bin/rules/configureDCCluster.properties -targetNodeName DecisionCenterNode01 -dmgrHostName localhost -dmgrPort 8879 -uninstall
     ```
Table 8. Cluster configuration parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-dmgrAdminUsername</td>
<td>Mandatory</td>
<td>The WebSphere Application Server administrator’s user identifier that you defined when you installed Decision Center.</td>
</tr>
<tr>
<td>-dmgrAdminPassword</td>
<td>Mandatory</td>
<td>The WebSphere Application Server administrator’s password that you defined when you installed Decision Center.</td>
</tr>
<tr>
<td>-clusterPropertiesFile</td>
<td>Mandatory</td>
<td>The full path to the properties file.</td>
</tr>
<tr>
<td>-uninstall</td>
<td>Mandatory</td>
<td>Removes all the applications and resources from the cluster. The cluster itself is not removed because it might contain other applications.</td>
</tr>
<tr>
<td>-createNode</td>
<td>Optional</td>
<td>If you specify this parameter, the script creates a default node profile with the name set in the <code>-targetNodeName</code> parameter.</td>
</tr>
<tr>
<td>-targetNodeName</td>
<td>Optional</td>
<td>By default, the name of the target installation node is set to <code>RulesNode01</code>.</td>
</tr>
<tr>
<td>-dmgrHostName</td>
<td>Optional</td>
<td>Use this parameter to specify the name of the host on which the deployment manager is located. This parameter is mandatory if you create the target node by setting the <code>-targetNodeName</code> parameter.</td>
</tr>
<tr>
<td>-dmgrPort</td>
<td>Optional</td>
<td>The SOAP TCP port of the deployment manager. By default, 8879.</td>
</tr>
</tbody>
</table>

Results

This command has the following results:
- Uninstalls the clustered application (`teamserver`).
- Uninstalls the data source.
- Removes application users and groups.

Configuring the event runtime by using profile templates

You can create new profiles for the event runtime and configure a WebSphere Application Server cluster for high availability and scalability.

To create a new stand-alone server to host the event runtime, you must first create a new profile. You can create a profile by using the Profile Management Tool or by using the `manageprofiles` command.

Augmenting an existing WebSphere Application Server management profile

To create a managed Decision Server Events environment, for example, a cluster, you must have a WebSphere Application Server management profile that you then augment. You can augment a profile by using the Profile Management Tool or by using the `manageprofiles` command.

Before you begin

Before you augment a WebSphere Application Server management profile, ensure that the deployment manager in the profile is stopped.
During the profile creation task, you are prompted to select the database manager which hosts the tables for the event runtime. Unless you are using Apache Derby, you must create this database before starting to create the profile. For more information, see “Creating the event runtime database” on page 109.

After you have augmented the profile, review the profile security configuration to check the administrative security and application security settings. If administrative security is enabled, application security must also be enabled.

About this task

You can augment an existing profile as follows:

- On a 32-bit operating system, you can use the Profile Management Tool graphical user interface. The Profile Management Tool is not supported on an HP-UX operating system, nor is it on z/OS.
- On a 32-bit operating system or any other operating system, you can use the manageprofiles command.

Augmenting an existing management profile by using the Profile Management Tool

About this task

To invoke the Profile Management Tool directly:

- On Windows only, click Start > All Programs > IBM WebSphere > Application Server Network Deployment V8.0 > Profile Management Tool.
- On multiplatforms, switch to the <WAS_HOME>/bin/ProfileManagement directory and run the pmt script, where <WAS_HOME> is the installation location of WebSphere Application Server. On Windows run pmt.bat, and on Linux and UNIX run pmt.sh.

To augment an existing management profile with Operational Decision Manager Decision Server Events:

Procedure

1. On the Welcome page, click Launch Profile Management Tool.
2. On the Profiles page, highlight an existing management profile and click Augment.
3. On the Augment Selection page, select Application server profile augmented with Operational Decision Manager Decision Server Events.
4. On the Installation Location page, enter or browse for the fully-qualified location where Decision Server Events is installed or accept the default location if correct.
5. If WebSphere Application Server administrative security is turned on in the profile, the Administrative Security page is displayed. Enter the user ID and password.
6. On the Database Configuration page,
   a. Select a database manager from the list and specify the appropriate connection details. This database manager hosts the tables for the event runtime.
      If you select embedded Apache Derby, the repository database is created for you. In all other cases, you must have already created the database for the repository.
b. If you did not select embedded Apache Derby, you must provide connection details including the fully-qualified location and name of the JDBC driver JAR file for your chosen database manager. Enter the following details for the database server:

- **Database name**: The name of the database that hosts the Decision Server Events event runtime tables.
- **Database server host name or IP address**: The name of the computer where the database server is located. The default is localhost.
- **Database TCP/IP service port or listener port**: The connection port number for the database manager.
- **Fully-qualified location and name of the JDBC driver file**: Enter the file name and location.
- **User name**: The user name that is used by Decision Server Events to connect to the database server. This user name must have administrative privileges.
- **Password**: The password associated with the user name. Confirm the password.

c. Click **Test Connection** to validate the connection to the database. Ensure that the connection is successful before proceeding with profile creation.

7. On the Messaging Provider Configuration page, select one of:

- **WebSphere Application Server default messaging**. Decision Server Events is configured to use the default messaging provider that is embedded in WebSphere Application Server as the JMS provider.
- **WebSphere MQ JMS messaging**. Supply the WebSphere MQ related information and Decision Server Events is configured to use WebSphere MQ as the JMS provider.
- **Do not configure a messaging provider**. No messaging provider is configured during profile creation. You must configure a single JMS provider before starting Decision Server Events.

After profile augmentation has completed, you can change JMS providers, or configure the JMS provider of your choice, but you must configure a single JMS provider before starting Decision Server Events.

To use, or to switch to, WebSphere Application Server default messaging as the JMS provider, see "Configuring WebSphere Application Server default messaging to be the JMS provider" on page 113.

To use, or to switch to, WebSphere MQ as the JMS provider, see "Configuring WebSphere MQ to be the JMS provider" on page 115.

8. On the Profile Augmentation Summary page, review the information and click **Augment** to augment the profile.

**Results**

The existing management profile is augmented with Operational Decision Manager Decision Server Events.

**What to do next**

If administrative security is enabled for the profile, you must also enable application security for the profile.
Augmenting an existing management profile by using the manageprofiles command

Procedure

1. Open a command prompt (Windows) or a terminal emulator (Linux and UNIX) and navigate to the was_install_dir/bin directory.
2. Run the manageprofiles command by using manageprofiles.bat for Windows or manageprofiles.sh for Linux and UNIX. Provide the following parameters:

   - **-augment**
     
     To augment the existing profile.

   - **-templatePath <template_path>**
     
     where <template_path> is the location of the profile template. Provide the following value for the management profile: was_install_dir/profileTemplates/wbe/management.

   - **-profileName profile_name**
     
     where profile_name is the name of the existing management profile that you are augmenting.

   Optionally, you can specify the following parameters:

   - **-wbeHome installation location**
     
     where installation location is the fully-qualified path that is the installation location for Decision Server Events. You must specify this parameter if you have installed Decision Server Events in a nondefault location.

   - **-HostName host_name**
     
     where host_name is the host name of the computer hosting the profile.

   - **-adminUserName username**
     
     where username is the user ID that is used to access WebSphere Application Server. Only required if WebSphere Application Server administrative security is turned on.

   - **-adminPassword password**
     
     where password is the password for the user ID that is used to access WebSphere Application Server. Only required if WebSphere Application Server administrative security is turned on.

   - **-wbeDbType database_type**
     
     where database_type specifies the database product. Valid values are:
     - Derby_Embedded: not supported for use in a clustered environment
     - DB2_Universal: the default for the management profile
     - Derby_NetworkServer
     - Oracle
     - MS_SQL_Server

   - **-wbeDbName database_name**
     
     where database_name is the name of the database. Required unless -wbeDbType is Derby_Embedded.

   - **-wbeUserId database_user_name**
     
     where database_user_name is the user ID that is used to access the database server. Required unless -wbeDbType is Derby_Embedded.

   - **-wbeDbPassword password**
     
     where password is the password for the user ID that is used to access the database server. Required unless -wbeDbType is Derby_Embedded.
-\texttt{wbeDbJDBCClasspath \textit{database\_jdbc\_classpath}}

where \textit{database\_jdbc\_classpath} specifies the path to the JDBC class path files. Required unless \texttt{-wbeDbType} is \textit{Derby\_Embedded}.

-\texttt{wbeDbHostName \textit{host\_name}}

where \textit{host\_name} is the host name for the database server. Required unless \texttt{-wbeDbType} is \textit{Derby\_Embedded}.

-\texttt{wbeDbServerPort \textit{port\_number}}

where \textit{port\_number} is the port where the TCP/IP service is assigned or the port on which the database is listening. Required unless \texttt{-wbeDbType} is \textit{Derby\_Embedded}.

-\texttt{wbeMsgingType \textit{messaging\_type}}

where \textit{messaging\_type} specifies which JMS provider is to be configured. Valid values are:

- \textit{Default\_Messaging}: Decision Server Events is configured to use
  WebSphere Application Server default messaging. This value is the default value.
- \textit{MQ\_JMS\_Messaging}: Decision Server Events is configured to use WebSphere MQ.
- \textit{No\_Messaging}: No JMS provider is configured. Before executing Decision Server Events, you must configure a JMS provider manually. See "Configuring WebSphere Application Server default messaging to be the JMS provider" on page 113 or "Configuring WebSphere MQ to be the JMS provider" on page 115.

-\texttt{wbeMqMsgingQmgrName \textit{queue\_manager\_name}}

where \textit{queue\_manager\_name} is the WebSphere MQ queue manager name. Required only if \texttt{-wbeMsgingType} is \textit{MQ\_JMS\_Messaging}.

-\texttt{wbeMqMsgingQmgrHostName \textit{host\_name}}

where \textit{host\_name} is the host name of the WebSphere MQ queue manager. Required only if \texttt{-wbeMsgingType} is \textit{MQ\_JMS\_Messaging}.

-\texttt{wbeMqMsgingQmgrPort \textit{port\_number}}

where \textit{port\_number} is the port on which the WebSphere MQ queue manager is listening. Required only if \texttt{-wbeMsgingType} is \textit{MQ\_JMS\_Messaging}.

-\texttt{wbeMqMsgingTransType \textit{transport\_type}}

where \textit{transport\_type} is the WebSphere MQ client transport type, either \texttt{BINDINGS} or \texttt{CLIENT}. Required only if \texttt{-wbeMsgingType} is \textit{MQ\_JMS\_Messaging}.

-\texttt{disableBusSecurity}

This parameter disables bus security for the service integration bus, WbeBus, when the bus is created.

**Results**

The profile is now augmented.

**Examples**

The first example augments an existing management profile called Dmgr01 on Windows using embedded Apache Derby as the database provider and WebSphere Application Server default messaging as the messaging provider.
manageprofiles.bat -augment -templatePath "C:\Program Files\IBM\ODM851\WAS\profileTemplates\wbe\management" -profileName Dmgr01

The second example augments an existing management profile called Dmgr01 on Linux and UNIX using embedded Apache Derby as the database provider and WebSphere Application Server default messaging as the messaging provider.

manageprofiles.sh -augment -templatePath "/opt/ibm/ODM851/ODM/WAS/profileTemplates/wbe/management" -profileName Dmgr01

**What to do next**

If administrative security is enabled for the profile, you must also enable application security for the profile.

**Unaugmenting an existing profile by using the manageprofiles command**

**About this task**

To undo the augmentation of an existing profile by using the manageprofiles command:

**Procedure**

1. Open a command prompt (Windows) or a terminal emulator (Linux and UNIX) and navigate to the was_install_dir/bin directory.
2. Run the manageprofiles command by using manageprofiles.bat for Windows or manageprofiles.sh for Linux and UNIX. Provide the following parameters:

   -unaugment
     To unaugment the existing profile.
   -templatePath <template_path>
     where <template_path> is the location of the profile template. Provide the following value for the management profile: was_install_dir/profileTemplates/wbe/management.
   -profileName profile_name
     where profile_name is the name of the existing management profile that you are unaugmenting.

**Results**

The profile is now unaugmented. All Decision Server Events features are removed from the profile.

**Examples**

The first example unaugments an existing Decision Server Events management profile called WODMDmgr01 on Windows.

manageprofiles.bat -unaugment -templatePath "C:\Program Files\IBM\ODM851\WAS\profileTemplates\wbe\management" -profileName WODMDmgr01

The second example unaugments an existing Decision Server Events management profile called WODMDmgr01 on Linux and UNIX.
Augmenting an existing WebSphere Application Server custom profile

To create a managed Decision Server Events environment, for example, a cluster, you must have a WebSphere Application Server custom profile that you then augment. You can augment a profile by using the Profile Management Tool or by using the `manageprofiles` command.

**Before you begin**

Before you augment a WebSphere Application Server custom profile, you must ensure that any node agents or servers in the profile are stopped.

**About this task**

You can augment an existing profile as follows:

- On a 32-bit operating system, you can use the Profile Management Tool graphical user interface. The Profile Management Tool is not supported on an HP-UX operating system, nor is it on z/OS.
- On a 32-bit operating system or any other operating system, you can use the `manageprofiles` command.

**Augmenting an existing custom profile by using the Profile Management Tool**

**About this task**

To invoke the Profile Management Tool directly:

- On Windows only, click `Start > All Programs > IBM WebSphere > Application Server Network Deployment V8.0 > Profile Management Tool`.
- On multiplatforms, switch to the `<WAS_HOME>/bin/ProfileManagement` directory and run the `pmt` script, where `<WAS_HOME>` is the installation location of WebSphere Application Server. On Windows run `pmt.bat`, and on Linux and UNIX run `pmt.sh`.

To augment an existing custom profile with Operational Decision Manager Decision Server Events:

**Procedure**

1. On the Welcome page, click **Launch Profile Management Tool**.
2. On the Profiles page, highlight an existing custom profile and click **Augment**.
3. On the Augment Selection page, select **Application server profile augmented with Operational Decision Manager Decision Server Events**.
4. On the Profile Augmentation Summary page, review the information and click **Augment** to augment the profile.

**Results**

The existing custom profile is augmented with Operational Decision Manager Decision Server Events.
Augmenting an existing custom profile by using the manageprofiles command

Procedure
1. Open a command prompt (Windows) or a terminal emulator (Linux and UNIX) and navigate to the was_install_dir/bin directory.
2. Run the manageprofiles command by using manageprofiles.bat for Windows or manageprofiles.sh for Linux and UNIX. Provide the following parameters:

   - **-augment**
     To augment the existing profile.

   - **-templatePath <template_path>**
     Where <template_path> is the location of the profile template. Provide the following value for the custom profile: was_install_dir/profileTemplates/wbe/managed.

   - **-profileName profile_name**
     Where profile_name is the name of the existing custom profile that you are augmenting.

Results
The profile is now augmented.

Examples
The first example augments an existing custom profile called Custom01 on Windows.
manageprofiles.bat -augment -templatePath "C:\Program Files\IBM\ODM851\WAS\profileTemplates\wbe\managed"
   -profileName Custom01

The second example augments an existing custom profile called Custom01 on Linux and UNIX.
manageprofiles.sh -augment -templatePath "/opt/ibm/ODM851/ODM/WAS/profileTemplates/wbe/managed"
   -profileName Custom01

Unaugmenting an existing profile by using the manageprofiles command

About this task
To undo the augmentation of an existing profile by using the manageprofiles command:

Procedure
1. Open a command prompt (Windows) or a terminal emulator (Linux and UNIX) and navigate to the was_install_dir/bin directory.
2. Run the manageprofiles command by using manageprofiles.bat for Windows or manageprofiles.sh for Linux and UNIX. Provide the following parameters:

   - **-unaugment**
     To unaugment the existing profile.
-templatePath <template_path>
   where <template_path> is the location of the profile template. Provide the
   following value for the custom profile: was_install_dir/profileTemplates/
   wbe/managed.

-profileName profile_name
   where profile_name is the name of the existing custom profile that you are
   unaugmenting.

Results

The profile is now unaugmented. All Decision Server Events features are removed
from the profile.

Examples

The first example unaugments an existing Decision Server Events custom profile
called WODMCustom01 on Windows.
manageprofiles.bat -unaugment -templatePath
   "C:\Program
   Files\IBM\ODM851\WAS\profileTemplates\wbe\managed"
   -profileName WODMCustom01

The second example unaugments an existing Decision Server Events custom profile
called WODMCustom01 on Linux and UNIX.
manageprofiles.sh -unaugment -templatePath
   "/opt/ibm/ODM851/ODM/WAS/profileTemplates/wbe/managed"
   -profileName WODMCustom01

Creating profiles for a gold topology cluster

In this task, you create the WebSphere Application Server profiles augmented with
Decision Server Events that are required for a gold topology cluster.

Procedure

1. Install Decision Server Events on both Computer1 and Computer2.
2. On Computer1, create a WebSphere Application Server management profile.
   a. Navigate to the was_install_dir/bin/ProfileManagement directory and start
      the Profile Management Tool by running the pmt.bat or pmt.sh command.
   b. On the Welcome page, select Launch Profile Management Tool then Create.
   c. On the Environment Selection page, under WebSphere Application Server,
      click Management then Next then Deployment manager.
   d. Click Typical profile creation or Advanced profile creation. For more
      information about typical and advanced profile creation, see Creating
      management profiles with deployment managers.
   e. Clear Launch the First steps console and click Finish.

Note: As an alternative to using the Profile Management Tool, for example
where a graphical user interface is not available, navigate to the
was_install_dir/bin directory and run the manageprofiles command. To create
the deployment manager, use the was_install_dir/profileTemplates/
management/ profile template. For more details, see manageprofiles command.
3. Ensure that the deployment manager is stopped, then augment the WebSphere
   Application Server management profile with Decision Server Events. See

IBM Operational Decision Manager: Configuring Operational Decision Manager on WebSphere Application Server
4. Start the deployment manager on Computer1:
   a. Navigate to the `was_install_dir/profiles/dmgr_profile_name/bin` directory.
   b. Run the `startManager.bat` for Windows or the `startManager.sh` command for Linux and UNIX.

5. On Computer1, create a WebSphere Application Server custom profile.
   a. Navigate to the `was_install_dir/bin/ProfileManagement` directory and start the Profile Management Tool by running the `pmt.bat` or `pmt.sh` command.
   b. On the Welcome page, select Launch Profile Management Tool then Create.
   c. On the Environment Selection page, under WebSphere Application Server, click Custom profile then Next.
   d. Click Typical profile creation or Advanced profile creation. For more information about typical and advanced profile creation, see Creating custom profiles.
   e. On the Federation page, federate the custom profile into the cell of the deployment manager.
      • Enter the connection details for the deployment manager that you created in step 2 on page 40.
      • Clear Federate this node later.
      • Click Next then click Create.
      If federation is successful, you can skip step 7.
   f. Clear Launch the First steps console and click Finish.

**Note:** As an alternative to using the Profile Management Tool, for example where a graphical user interface is not available, navigate to the `was_install_dir/bin` directory and run the `manageprofiles` command. To create the custom profile, use the `was_install_dir/profileTemplates/managed/profile` template. For more details, see manageprofiles command.

6. Check that the system clocks on all of the computers that you are using to create the cluster are synchronized and, to ensure that any time-based logic evaluates correctly, that the clocks are set to the same time zone.

7. If you successfully federated in step 5 you can skip this step. Otherwise, on Computer1, federate the custom profile into the cell of the deployment manager:
   a. Navigate to the `was_install_dir/bin` directory.
   b. Run the following command:
      ```
      addnode -profileName profile_name
      <deployment manager host> <deployment manager port>
      ```
      where `profile_name` is the name of the profile that you created in step 5, where `<deployment manager host>` (required) is the host name of the computer where the deployment manager is running, for example localhost, and where `<deployment manager port>` (optional) is the SOAP port of the deployment manager, with a default of 8879.
   c. Start the WebSphere Application Server administrative console for the deployment manager.
   d. Log in to the WebSphere Application Server administrative console and validate that the application server has been successfully federated to the deployment manager. Click System administration > Nodes.
On *Computer1*, the custom profile is now federated into the cell of the deployment manager.

8. Augment the WebSphere Application Server custom profile with Decision Server Events.
   a. Ensure that the node agent is stopped. To stop the node agent in the WebSphere Application Server administrative console, click **System administration > Node agents**. Select the node agent and click **Stop**.
   b. Augment the custom profile with Decision Server Events. See "Augmenting an existing WebSphere Application Server custom profile" on page 38.

9. On *Computer2*, repeat steps 5 on page 41 to 8 to create, federate, and augment a second WebSphere Application Server custom profile.

**What to do next**

Now that the profiles have been created, you can configure your gold topology cluster. See "Configuring a gold topology cluster."

**Configuring a gold topology cluster**

In this task, you configure the gold topology cluster for which you created profiles, to complete creating the cluster environment.

**Before you begin**

Before configuring the cluster, you must first create the profiles. See "Creating profiles for a gold topology cluster" on page 40.

The following steps assume that security is enabled for the service integration bus, as described in "Securing the service integration bus" on page 142. If security is not enabled, replace `SIB_ENDPOINT_SECURE_ADDRESS` with `SIB_ENDPOINT_ADDRESS` and omit the `BootstrapSecureMessaging` chain from each `hostname:port` pair in steps 8 on page 44, 9 on page 44, 10 on page 45.

**Procedure**

1. Create a WebSphere Application Server cluster and add two application servers as members of the cluster:
   a. Click **Servers > Clusters > WebSphere application server clusters**.
   b. Click **New**.
   c. In the **Cluster name** field, enter a name for the cluster, for example `EventRuntimeCluster`, then click **Next**.
   d. In the **Member name** field, enter a name for the first application server that you are adding to the cluster, for example `eventruntime01`.
   e. Select the name of the node on which the first application server runs, then click **Next**.
   f. In the **Member name** field, enter a name for the next application server that you are adding to the cluster, for example `eventruntime02`.
   g. Select the name of the node on which this application server runs. Click **Add member**, then click **Next**.
   h. If you want to add more application servers to the cluster, repeat steps f. and g.
   i. Click **Finish** and **Save**.

2. Identify and record the `hostname:port`:
   `BootstrapSecureMessaging,hostname:port:BootstrapSecureMessaging`, where
each hostname:port pair corresponds to the host name and SIB_ENDPOINT_SECURE_ADDRESS of the application servers in your event runtime cluster. To identify this information from the administrative console:

a. Click Servers > WebSphere application server clusters > Event Runtime Cluster > Cluster members and note the hostname.

b. For each cluster member, click Ports and record the port number associated with the SIB_ENDPOINT_SECURE_ADDRESS.

3. Create a JVM custom property that points to the Operational Decision Manager installation directory.

Note: You must do this step for every application server in the cluster.

a. Click Servers > Server Types > WebSphere application servers > server-name > Java and Process Management > Process Definition > Java Virtual Machine > Custom properties, where server-name is an application server in the cluster.

b. Click New.

c. In the Name field, type wbe.home

d. As the value, enter the location of the home directory of your Operational Decision Manager installation.
   • For example, if you installed using the launchpad and accepted the default install location:
     On Windows:
     C:\Program Files\IBM\ODM851
     On Linux and UNIX:
     /opt/ibm/ODM851/ODM
   • For example, if you installed using Installation Manager and accepted the default install location:
     On Windows:
     C:\Program Files\IBM\ODM851
     On Linux and UNIX:
     /opt/ibm/ODM851

e. Click Apply and Save.

4. Modify the server heapsize parameters.

Note: You must do this step for every application server in the cluster.

a. Click Servers > Server Types > WebSphere application servers > server-name > Java and Process Management > Process Definition > Java Virtual Machine, where server-name is an application server in the cluster.

b. In the Initial heap size field, enter 768.

c. In the Maximum heap size field, enter 1024.

d. Click Apply and Save.

For more information about setting optimal heap settings, see “Tuning your event runtime configuration” on page 128.

5. Enable the Startup Bean service for each application server in the cluster.

Note: You must do this step for every application server in the cluster.

a. Click Servers > Server Types > WebSphere application servers > server-name > Container Services > Startup beans service, where server-name is an application server in the cluster.
b. Select **Enable service at server startup**.
c. Click **Apply** and **Save**.

6. Add the event runtime cluster as a member to the service integration bus:
   a. Click **Service integration > Buses > WbeBus > Bus members**.
   b. Click **Add**. Click **Cluster > DecisionServerCluster**, then click **Next**.
   c. Select a messaging engine policy setting from the options **High availability**, **Scalability**, or **Scalability with high availability**. For more details about your selection, see [Messaging engine policy assistance](#).
   d. To ensure the recovery of JMS messages after a failover, you must configure a message store. Set the type of message store by selecting either **File store** or **Data store**. For more information about this concept and other high availability concepts, see [Establishing high availability](#) Click **Next**
   e. On the **Configure messaging engines** panel, click the messaging engine name.
   f. Set the data store values:
      1) Either, if you are using a file store, set the value of **Log directory path** to point to a data store on a shared file system, for example, c:\filestoredirectory. Set the **Permanent store directory path** to point to a data store on a shared file system.
      2) Or, if you are using a database, set the **Data connection JNDI name**, the schema name and the authentication alias, to point to a data store on a database that you have already created.
   g. Click **Next**.
   h. Click **Change heap sizes**. Accept the default values.
   i. Review the Summary information. Click **Finish** and **Save**.

7. Create three service integration bus destinations for the event runtime messaging destinations:
   a. Click **Service integration > Buses > WbeBus > Destinations**.
   b. Click **New**.
   c. Click **Queue**, and click **Next**. In the **Identifier** field, type eventQueue, then click **Next**.
   d. Repeat this step to create two further queues, with identifiers of historyModuleQueue and durableEventQueue.
   e. Click **Finish** and **Save**.

8. Modify the provider endpoints for the connection factories to point to the servers in your event runtime cluster:
   a. Click **Resources > JMS > Connection factories**.
   b. From the **Scope** list, click **Cell=cell-name**, where **cell-name** is the name of the cell.
   c. Click **WbeConnectionFactory**
   d. In the **Provider endpoints** field, type hostname:port:BootstrapSecureMessaging,hostname:port:BootstrapSecureMessaging, where each hostname:port pair corresponds to the host name and SIB_ENDPOINT_SECURE_ADDRESS of the application servers in your event runtime cluster.
   e. Click **Apply** and **Save**.

9. Modify the provider endpoints for the JMS queue connection factory to point to the servers in your event runtime cluster:
a. Click Resources > JMS > Queue connection factories.

b. From the Scope list, click Cell=cell-name, where cell-name is the name of the cell.

c. Click WbeQueueConnectionFactory

d. In the Provider endpoints field, type hostname:port:BootstrapSecureMessaging,hostname:port:BootstrapSecureMessaging, where each hostname:port pair corresponds to the host name and SIB_ENDPOINT_SECURE_ADDRESS of the application servers in your event runtime cluster.

e. Click Apply and Save.

10. Modify the provider endpoints for the JMS topic connection factory to point to the servers in your event runtime cluster:

a. Click Resources > JMS > Topic connection factories.

b. From the Scope list, click Cell=cell-name, where cell-name is the name of the cell.

c. Click WbeTopicConnectionFactory

d. In the Provider endpoints field, type hostname:port:BootstrapSecureMessaging,hostname:port:BootstrapSecureMessaging, where each hostname:port pair corresponds to the host name and SIB_ENDPOINT_SECURE_ADDRESS of the application servers in your event runtime cluster.

e. Click Apply and Save.

11. Install the event runtime application:

a. Click Applications > New Application > New Enterprise Application.

b. Click Remote file system.

c. Click Browse and click a node or deployment manager to browse its file system. Locate the wberuntimeear application file.

d. Enter the full path of the location of the wberuntimeear application file. For example, on Windows enter C:\Program Files\IBM\ODM851\runtime\wberuntimeear.ear; and on Linux and UNIX enter /opt/ibm/ODM851/ODM/runtime/wberuntimeear.ear. Click Next.

e. In the Preparing for the application installation window, select Fast Path. Click Next.

f. In the Select installation options window, accept the default options and click Next.

12. Optional: Confirm that the EventWidgetEar is installed if you want to use Event Widgets in your cluster environment. Click Applications > Application Types > WebSphere enterprise applications. Ensure that the EventWidgetsEar application is listed and shown as started. For more information about installing the Event Widgets, see Selecting the features to install. The following Event Widgets are supported in a cluster environment:

• Event Chart Manager
• Event Chart
• Event Capture
• Event Replay
Restriction: The Event Tester widget is not supported in a cluster environment.

13. To ensure that all these changes take effect, and that the cluster can be started correctly, start the node agents on all the computers in the cluster:
   a. At a command prompt, navigate to the correct directory:
      `cd was_install_dir/bin`
   b. Run the `startNode` command. On Windows, run `startNode.bat`, on Linux and UNIX run `startNode.sh`, with the following optional parameter:
      `-profileName profile_name`
      where `profile_name` is the name of the profile.

14. Start all the application servers in the cluster:
   a. To do this step in the WebSphere Application Server administrative console, click `Servers > Clusters > WebSphere application server clusters` > `<cluster-name>` > `Cluster members`, where `<cluster-name>` is the name that you entered in step 1 on page 42.
   b. Select the application server and click `Start`.

Results

The gold topology cluster is now configured and running. If the cluster fails to start, see Starting clusters.

What to do next

Configure the technology connectors. See Administering connector application deployment in a managed server environment for the File System, HTTP, JDBC, JMS, REST, and SOAP connectors; or see Configuring technology connectors on a separate computer on page 126 for the stand-alone technology connectors.

Creating and configuring a cluster that uses WebSphere MQ as the messaging provider

You can cluster Decision Server Events to provide high availability and scalability for the event runtime, but by using WebSphere MQ as the messaging provider instead of by using WebSphere Application Server default messaging as the messaging provider.

About this task

A Decision Server Events cluster consists of a single deployment manager and one or more managed nodes. An event runtime cluster spans the managed nodes and provides high availability and scalability of the event runtime. This documentation describes how to set up a clustered event runtime for high availability and scalability, that uses WebSphere MQ as the messaging provider; but not how to configure WebSphere MQ for high availability. You create a two node cluster spanning two computers, as shown in the following diagram:
There is one cluster, contained inside a single cell.

A management profile is configured on Computer1.

There are managed nodes (custom profiles) on both Computer1 and Computer2.

Each computer has Decision Server Events and WebSphere Application Server installed.

WebSphere MQ is installed and configured for use with Decision Server Events and provides all of the JMS messaging capability for the event runtime.

This cluster topology can be extended across more computers in a similar way, in a single cluster.

To complete this task:

- Configure WebSphere MQ for use as a messaging provider. See "Configuring WebSphere MQ for use as a messaging provider" on page 48.
- Create profiles for a cluster that uses WebSphere MQ as the messaging provider. See "Creating profiles for a cluster that uses WebSphere MQ as the messaging provider" on page 48.
- Configure your cluster for Decision Server Events. See "Configuring a cluster that uses WebSphere MQ as the messaging provider" on page 50.
Configuring WebSphere MQ for use as a messaging provider

Before you can use WebSphere MQ as the messaging provider for your Decision Server Events cluster, you must do some configuration tasks in WebSphere MQ.

About this task

The WebSphere MQ administrator must do the following tasks:

Procedure

1. Create the queue manager, by using the `crtmqm` command.
2. Start the queue manager, by using the `strmqm` command.
3. If you are using WebSphere MQ V7.1, publish/subscribe must be enabled. Use the `ALTER QMGR` command for this queue manager.
4. Create the queues required, by running:
   ```
   runmqsc
   queue-manager-name <mq-install-dir>\java\bin\MQJMS_PSQ.mqsc
   ```
   where `mq-install-dir` is the installation location of WebSphere MQ.
5. If you are configuring a cluster, create the extra queues required, by running:
   ```
   runmqsc queue-manager-name <
   <InstallDir>\config\was\create_MQ_JMS_MQ_queues.mqsc
   ```
   where `<InstallDir>` is the installation location of Decision Server Events.

Results

The WebSphere MQ queue manager is created and configured.

What to do next

Create WebSphere Application Server profiles for your Decision Server Events cluster. See "Creating profiles for a cluster that uses WebSphere MQ as the messaging provider".

Creating profiles for a cluster that uses WebSphere MQ as the messaging provider

In this task, you create the WebSphere Application Server profiles augmented with Decision Server Events that are required for a cluster that uses WebSphere MQ as the messaging provider.

Before you begin

Configure WebSphere MQ for use as a messaging provider. See "Configuring a cluster that uses WebSphere MQ as the messaging provider" on page 50.

Procedure

1. Install Decision Server Events on both `Computer1` and `Computer2`.
2. On `Computer1`, create a WebSphere Application Server management profile.
   a. Navigate to the `was_install_dir\bin\ProfileManagement` directory and start the Profile Management Tool by running the `pmt.bat` or `pmt.sh` command.
   b. On the Welcome page, select `Launch Profile Management Tool` then `Create`.
   c. On the Environment Selection page, under `WebSphere Application Server`, click `Management` then `Next` then `Deployment manager`.

IBM Operational Decision Manager: Configuring Operational Decision Manager on WebSphere Application Server
Click Typical profile creation or Advanced profile creation. For more information about typical and advanced profile creation, see Creating management profiles with deployment managers.

e. Clear Launch the First steps console and click Finish.

Note: As an alternative to using the Profile Management Tool, for example where a graphical user interface is not available, navigate to the was_install_dir/bin directory and run the manageprofiles command. To create the deployment manager, use the was_install_dir/profileTemplates/management/profile template. For more details, see manageprofiles command.

3. Ensure that the deployment manager is stopped, then augment the WebSphere Application Server management profile with Decision Server Events. See “Augmenting an existing WebSphere Application Server management profile” on page 32. You must select WebSphere MQ as the messaging provider and provide the appropriate connection details.

4. Start the deployment manager on Computer1:
   a. Navigate to the was_install_dir/profiles/dmgr_profile_name/bin directory.
   b. Run the startManager.bat for Windows or the startManager.sh command for Linux and UNIX.

5. On Computer1, create a WebSphere Application Server custom profile.
   a. Navigate to the was_install_dir/bin/ProfileManagement directory and start the Profile Management Tool by running the pmt.bat or pmt.sh command.
   b. On the Welcome page, select Launch Profile Management Tool then Create.
   c. On the Environment Selection page, under WebSphere Application Server, click Custom profile then Next.
   d. Click Typical profile creation or Advanced profile creation. For more information about typical and advanced profile creation, see Creating custom profiles.
   e. On the Federation page, federate the custom profile into the cell of the deployment manager.
      - Enter the connection details for the deployment manager that you created in step 2 on page 48.
      - Clear Federate this node later.
      - Click Next then click Create.

   If federation is successful, you can skip step 7.
   f. Clear Launch the First steps console and click Finish.

Note: As an alternative to using the Profile Management Tool, for example where a graphical user interface is not available, navigate to the was_install_dir/bin directory and run the manageprofiles command. To create the custom profile, use the was_install_dir/profileTemplates/managed/profile template. For more details, see manageprofiles command.

6. Check that the system clocks on all of the computers that you are using to create the cluster are synchronized and, to ensure that any time-based logic evaluates correctly, that the clocks are set to the same time zone.

7. If you successfully federated in step 5 you can skip this step. Otherwise, on Computer1, federate the custom profile into the cell of the deployment manager:
   a. Navigate to the was_install_dir/bin directory.
   b. Run the following command:
addnode -profileName profile_name
<deployment manager host> <deployment manager port>

where profile_name is the name of the profile that you created in step 5 on page 49, where <deployment manager host> (required) is the host name of the computer where the deployment manager is running, for example localhost, and where <deployment manager port> (optional) is the SOAP port of the deployment manager, with a default of 8879.

c. Start the WebSphere Application Server administrative console for the deployment manager.
d. Log in to the WebSphere Application Server administrative console and validate that the application server has been successfully federated to the deployment manager. Click System administration > Nodes.

On Computer1, the custom profile is now federated into the cell of the deployment manager.

8. Augment the WebSphere Application Server custom profile with Decision Server Events.
a. Ensure that the node agent is stopped. To stop the node agent in the WebSphere Application Server administrative console, click System administration > Node agents. Select the node agent and click Stop.
b. Augment the custom profile with Decision Server Events. See "Augmenting an existing WebSphere Application Server custom profile" on page 38.

9. On Computer2, repeat steps 5 on page 49 to 8 to create, federate, and augment a second WebSphere Application Server custom profile.

What to do next

Now that the profiles have been created, you can configure your cluster that uses WebSphere MQ as the messaging provider. See "Configuring a cluster that uses WebSphere MQ as the messaging provider."

Configuring a cluster that uses WebSphere MQ as the messaging provider

In this task, you configure the cluster for which you have created profiles, to complete creating the cluster environment.

Before you begin

Before configuring the cluster, you must first create the profiles. See "Creating profiles for a cluster that uses WebSphere MQ as the messaging provider" on page 48.

Procedure

1. Create a WebSphere Application Server cluster and add two application servers as members of the cluster:
a. Click Servers > Clusters > WebSphere application server clusters.
b. Click New.
c. In the Cluster name field, enter a name for the cluster, for example DecisionServerCluster, then click Next.
d. In the Member name field, enter a name for the first application server that you are adding to the cluster, for example eventruntime01.
e. Select the name of the node on which the first application server runs, then click Next.
f. In the **Member name** field, enter a name for the next application server that you are adding to the cluster, for example `eventruntime02`.

g. Select the name of the node on which this application server runs. Click **Add member**, then click **Next**.

h. If you want to add more application servers to the cluster, repeat steps f. and g.

i. Click **Finish** and **Save**.

2. Create a JVM custom property that points to the Operational Decision Manager installation directory.

   **Note:** You must do this step for every application server in the cluster.

   a. Click **Servers > Server Types > WebSphere application servers > server-name > Java and Process Management > Process Definition > Java Virtual Machine > Custom properties**, where `server-name` is an application server in the cluster.

   b. Click **New**.

   c. In the **Name** field, type `wbe.home`

   d. As the value, enter the location of the home directory of your Operational Decision Manager installation.

      - For example, if you installed using the launchpad and accepted the default install location:
        - **On Windows:**
          - C:\Program Files\IBM\ODM851
        - **On Linux and UNIX:**
          - /opt/ibm/ODM851/ODM

      - For example, if you installed using Installation Manager and accepted the default install location:
        - **On Windows:**
          - C:\Program Files\IBM\ODM851
        - **On Linux and UNIX:**
          - /opt/ibm/ODM851

   e. Click **Apply** and **Save**.

3. Enable the Startup Bean service for each application server in the cluster.

   **Note:** You must do this step for every application server in the cluster.

   a. Click **Servers > Server Types > WebSphere application servers > server-name > Container Services > Startup beans service**, where `server-name` is an application server in the cluster.

   b. Select **Enable service at server startup**.

   c. Click **Apply** and **Save**.

4. Install the event runtime application:

   a. Click **Applications > New Application > New Enterprise Application**.

   b. Click **Remote file system**.

   c. Click **Browse** and click a node or deployment manager to browse its file system. Locate the `wberuntimeear` application file.

   d. Enter the full path of the location of the `wberuntimeear` application file. For example, on Windows enter `C:\Program Files\IBM\ODM851\runtime\`
wberuntimeear.ear; and on Linux and UNIX enter /opt.ibm/ODM851/ODM/runtime/wberuntimeear.ear. Click Next.

e. In the Clusters and Servers field, click WebSphere:cell=cell-
name,cluster=DecisionServerCluster, where cell-name is the name of the
cell.

f. Select both check boxes.

g. Click Apply, then Next.

h. Click Finish and Save.

5. To ensure that all these changes take effect, and that the cluster can be started
correctly, start the node agents on all the computers in the cluster:

a. At a command prompt, navigate to the correct directory:
   
   cd was_install_dir/bin

b. Run the startNode command. On Windows, run startNode.bat, on Linux
   and UNIX run startNode.sh, with the following optional parameter:

   -profileName profile_name
   
   where profile_name is the name of the profile.

6. Start all the application servers in the cluster:

a. To do this step in the WebSphere Application Server administrative
   console, click Servers > Clusters > WebSphere application server clusters >
   <cluster-name> > Cluster members, where <cluster-name> is the name that
   you entered in step 1 on page 50.

b. Select the application server and click Start.

Results

The cluster that uses WebSphere MQ as the messaging provider is now configured
and running. If the cluster fails to start, see Starting clusters.

What to do next

Configure the technology connectors. See Administering connector application
deployment in a managed server environment for the File System, HTTP, JDBC,
JMS, REST, and SOAP connectors; or see “Configuring technology connectors on a
separate computer” on page 126 for the stand-alone technology connectors.

Defining a highly available collection of catalog servers

The event runtime uses WebSphere eXtreme Scale during event processing. By
default, the WebSphere eXtreme Scale catalog service runs only within the
deployment manager. If the deployment manager is unavailable, the WebSphere
eXtreme Scale shard placement might be affected. If you define a catalog service
domain, you increase the availability of your event runtime cluster if the
deployment manager is unavailable or restarts.

About this task

Catalog service domains define a group of catalog servers that manage the
placement of shards and monitor the health of container servers in your data grid.

Procedure

Follow the procedure described in Creating catalog service domains in WebSphere
Application Server to include the catalog service in multiple node agent processes.
or in an application server that is not hosting a WebSphere eXtreme Scale application.
Chapter 2. Configuring Rule Execution Server on WebSphere Application Server

To use Rule Execution Server on a new instance of WebSphere Application Server for distributed platforms or for Linux on System z®, you follow a sequence of configuration steps. On WebSphere Application Server for distributed platforms, you can configure a cluster by using a profile template.

Operational Decision Manager 8.5.1 supports WebSphere Application Server 8.0, 8.5, and 8.5.5.

The steps or actions that are specific to Linux on System z are indicated separately.

Before you start: Opening the administration console

To configure Rule Execution Server on WebSphere Application Server, you must follow specific steps. Some of them depend on the persistence type.

Before you begin

Before you configure Rule Execution Server, make sure that you have the following software installed.
- Rule Execution Server
- WebSphere Application Server for distributed platforms or for Linux on System z
- DB2 Version 9.5 or 10
- Java™ 1.6 included in your PATH statement.

You also need the following rights:
- Access to the WebSphere Integrated Solutions Console
- DB2 Administrator authority to CREATE a database and CREATE & UPDATE tables
- Authority to create a directory structure for the Rule Execution Server files
- Authority to start and stop WebSphere Application Server

About this task

To configure Rule Execution Server on WebSphere Application Server, whether on distributed platforms or on Linux on System z, you work in the WebSphere Integrated Solutions Console.

Procedure

1. Make sure that you have a WebSphere Application Server profile.
   The predefined name of the first server profile is AppSrv01. Refer to the WebSphere Application Server documentation if necessary.
2. Start the server by clicking Start > All programs > IBM WebSphere > IBM WebSphere Application Server version_number > Profiles > AppSrv01 > Start the server
Tip: As an alternative, first open the First Steps window from the Windows Start menu, and then start the server, and later the console, from the First Steps window.

3. Start the console by clicking Start > All programs > IBM WebSphere > IBM WebSphere Application Server version_number > Profiles > AppSrv01 > Administrative console You might have to pass the security steps of your browser.

4. Log in by entering the user name and password that you defined when you created the profile.
   For example: wasadmin and wasadmin. The WebSphere Integrated Solutions Console opens.

**What to do next**

In the first four steps, you set up a dedicated database and data source. Then, you activate security, create the database schema, and deploy the management EAR file, execution unit resource archive, and optionally the EAR file for transparent decision services.

The Rule Execution Server console includes an Installation Settings wizard. If you sign in as the administrator, you can use the wizard to create the database schema and run SQL drop statements that clear any existing Rule Execution Server database.

**Note:** Specific integration extensions are available from this URL: [WebSphere Operation Decision Management Integration SupportPacs](#)

The following table summarizes the steps to configure Rule Execution Server on WebSphere Application Server, depending on the type of persistence.

<table>
<thead>
<tr>
<th>Installation Steps</th>
<th>Persistence</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Step 1: Selecting and applying the persistence type” on page 58</td>
<td>File</td>
</tr>
<tr>
<td></td>
<td>Data source</td>
</tr>
<tr>
<td></td>
<td>JDBC</td>
</tr>
<tr>
<td>“Step 2: Restricting database user permissions” on page 58</td>
<td>file</td>
</tr>
<tr>
<td></td>
<td>default persistence mode</td>
</tr>
<tr>
<td></td>
<td>MySQL</td>
</tr>
<tr>
<td>“Step 3: Setting up your database” on page 59</td>
<td>not applicable</td>
</tr>
<tr>
<td></td>
<td>file</td>
</tr>
<tr>
<td></td>
<td>default persistence mode</td>
</tr>
<tr>
<td></td>
<td>MySQL</td>
</tr>
<tr>
<td>Installation Steps</td>
<td>Persistence</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>File</td>
</tr>
<tr>
<td><strong>Step 4: Setting up a data source and connection pool</strong> on page 61</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Creating a JDBC provider</strong> on page 61</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Creating a data source and connection pool</strong> on page 62</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Creating J2C authentication data</strong> on page 63</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Setting custom properties</strong> on page 64</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Testing the connection to the database</strong> on page 65</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Step 5: Activating security on WebSphere Application Server</strong> on page 66</td>
<td><strong>Creating users and groups</strong> on page 67</td>
</tr>
<tr>
<td></td>
<td><strong>Mapping user groups to the Monitor role</strong> on page 69</td>
</tr>
<tr>
<td></td>
<td><strong>Security policies for the Rule Execution Server console</strong> on page 70</td>
</tr>
<tr>
<td><strong>Step 6: Deploying the Rule Execution Server MBean descriptors</strong> on page 70</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Step 7: Deploying the XU RAR</strong> on page 71</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Step 8: Deploying the Rule Execution Server management EAR file</strong> on page 73</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Step 9: Creating a Rule Execution Server database schema</strong> on page 74</td>
<td>Not applicable</td>
</tr>
<tr>
<td><strong>Step 10: Deploying the hosted transparent decision service EAR file</strong> on page 81</td>
<td>Optional</td>
</tr>
<tr>
<td><strong>Step 11: Setting the DecodeUrlAsUTF8 custom property</strong> on page 82</td>
<td>Required if you deployed the EAR file for hosted transparent decision services (previous step).</td>
</tr>
<tr>
<td><strong>Verifying the deployment and configuration</strong> on page 91</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Step 1: Selecting and applying the persistence type

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

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

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

Solaris users

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

Decision Warehouse

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

MySQL persistence

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

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

For more information about these settings, see the MySQL 5.0 reference manual: [5.1.6. 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.

<table>
<thead>
<tr>
<th>Database permission</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE ANY INDEX</td>
<td>Not required</td>
</tr>
<tr>
<td>DROP ANY INDEX</td>
<td>Required</td>
</tr>
<tr>
<td>CREATE ANY SEQUENCE</td>
<td>Not required</td>
</tr>
<tr>
<td>DROP ANY SEQUENCE</td>
<td>Required</td>
</tr>
<tr>
<td>SELECT ANY SEQUENCE</td>
<td>Required</td>
</tr>
<tr>
<td>CREATE ANY TABLE</td>
<td>Not required</td>
</tr>
<tr>
<td>DROP ANY TABLE</td>
<td>Required</td>
</tr>
<tr>
<td>INSERT ANY TABLE</td>
<td>Required</td>
</tr>
<tr>
<td>SELECT ANY TABLE</td>
<td>Not required</td>
</tr>
<tr>
<td>UPDATE ANY TABLE</td>
<td>Required</td>
</tr>
<tr>
<td>DELETE ANY TABLE</td>
<td>Not required</td>
</tr>
<tr>
<td>CREATE ANY TRIGGER</td>
<td>Not required</td>
</tr>
<tr>
<td>CREATE ANY VIEW</td>
<td>Not required</td>
</tr>
<tr>
<td>DROP ANY VIEW</td>
<td>Required</td>
</tr>
</tbody>
</table>

**Step 3: Setting up your database**

If you are using database persistence, you must first create an empty schema for the database that is dedicated to Rule Execution Server and establish the credentials.

**Before you begin**

Before you set up your database, make sure that the persistence is set to datasource. If you set persistence to file, you can skip all the database-related tasks and proceed to "Step 8: Deploying the Rule Execution Server management EAR file" on page 73.

**About this task**

The credentials to access the database are required to establish the data source, which you do in "Creating a data source and connection pool" on page 62.

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

**Procedure**

To create an empty Derby database:
1. Stop the application server.
2. On the `<Derby_InstallDir>/bin/` path, start the `ij.bat` command.
   For Linux, the command is `ij`.
3. Create the database and connect to it.
   For example, to create the new `c:/resdb` database as the `resdbUser` user and
   connect to it, run the following command:
   ```
   ij>connect 'jdbc:derby:c:/resdb;user=resdbUser;
   password=resdbUser;create=true';
   ```
4. Close the `ij` utility.
   ```
   ij> quit;
   ```
5. Start the application server.

**Creating a DB2 database for Linux on System z**

Customizable scripts are provided to create your DB2 database schema, but you
must first create an empty database.

**Before you begin**

The existing DB2 installation is assumed to have automatic storage and a 32 K
buffer pool with the default ID of BP32K. To create a DB2 database, you must have
DB2 administrator rights. For more information, consult your DB2 database
administrator or the DB2 documentation, in particular about creating databases
and other DB-related tasks such as [CREATE DATABASE command](#).

**Procedure**

Enter the following commands:
```
su {DB2AdminName}
db2
CREATE DATABASE {RESDBName}
```

**What to do next**

To set up the database, you can use the Rule Execution Server console or you can
run the SQL scripts that are provided in `<InstallDir>/executionserver/databases`.
A `readme` file in this directory provides more information about the scripts.

**Creating a DB2 database on z/OS**

**Before you begin**

You must first install DB2 and set the appropriate rights on the database, and then
customize JCL.
1. Check that you have DB2 for z/OS Version 9.1 or 10 installed
2. Check that you have DB2 Administrator authority to CREATE a database and
   CREATE & UPDATE tables.DB2.

For more information, consult your DB2 database administrator or the DB2
documentation, in particular about creating databases and other DB-related tasks
such as [CREATE DATABASE command](#).

**About this task**

After you have customized the JCL variables for your configuration, you can
submit the jobs to create a database for Rule Execution Server.
**Procedure**

1. Customize the JCL as indicated in the comments within each PDSE member, before you run them on the z/OS system.
   
   You can customize the JCL either by using a manual search-and-replace or by using an ISPF macro.

2. On the z/OS computer where DB2 is installed, run the following JCL scripts to create the RULEAPP tables, XOM table, EXECUTION_TRACES tables, and grant authority to the tables. This task is typically a database administrator’s task. **++DB2USER++** is the user ID to access DB2.
   
   a. `InstallDir/executionserver/jcl/HBRDSCDB.jcl`
   b. `InstallDir/executionserver/jcl/HBRDSXOM.jcl`
   c. `InstallDir/executionserver/jcl/HBRDSCTR.jcl`
   d. `InstallDir/executionserver/jcl/HBRDSGRN.jcl`

   **++DB2USER++** is the user ID to access DB2.

---

**Step 4: Setting up a data source and connection pool**

You must create a JDBC provider (data source) and connection pool for WebSphere Application Server.

**Creating a JDBC provider**

To enable Rule Execution Server, you create a JDBC provider in WebSphere Application Server, as the first step in creating a data source.

**Before you begin**

To create a JDBC provider, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 55.

**About this task**

The first step in creating a data source, based on the database schema that you have created in “Step 3: Setting up your database” on page 59, is to create a JDBC provider. You can install the JDBC provider on the cell, node, cluster, or server level. Follow these steps to install a Derby or DB2 JDBC provider at node level.

**Note:** Some drivers, such as Oracle OCI drivers, need to access additional libraries at run time (.dll or .so files). As a consequence, you must set up your working environment to access these libraries. For example, set the `PATH` and `LD_LIBRARY_PATH` environment variables.

**Procedure**

1. Click Resources > JDBC > JDBC Providers.

2. Under Scope, select the Node=xxx, Server=yyy, and then click New.

   xxx is the name of your node and yyy the name of your server.s

3. In Step 1, select the database type, provider type, and an implementation type that supports XA features.

   **Table 9. JDBC provider options for DB2 and Derby**

<table>
<thead>
<tr>
<th>Database type</th>
<th>Derby</th>
<th>DB2</th>
</tr>
</thead>
</table>

---
4. Click **Next**.
5. In **Step 2**, leave all the class path values, including the blanks, as the default values and click **Next**.
   A summary is provided in **Step 3**.
6. Check that the class path to the JAR file of your driver and the implementation class are correct.
   Default values are sufficient, except in very special cases.
7. Click **Finish**.
8. In the next panel, click **Save** to save the changes to the master configuration.
   If you work in a cluster environment, make sure that the **Synchronize changes with Nodes** check box is selected.
9. Restart your server for the changes to be taken into account.
   If you work with a DB2 database, the DB2 installation process might have installed the db2jcc_license_cisuz.jar file to a location that is not in the WebSphere Application Server path.
10. In this case, find the db2jcc_license_cisuz.jar file by running the following command.
    Before you run this command, it is best to set yourself as the root user to avoid permissions messages.
    ```bash
    find / -name db2jcc_license_cisuz.jar
    ```
11. From the WebSphere Integrated Solutions Console, open **Environment > WebSphere variables** and click **DB2UNIVERSAL_JDBC_DRIVER_PATH**.
12. Set the value to the path that the `find` command returned.
13. Click **OK** and **Save** to save the changes to the master configuration.

### Creating a data source and connection pool

Create a connection pool and a data source in WebSphere Application Server to enable Rule Execution Server.

**Before you begin**

To create a data source, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 55.

**Procedure**

1. In the WebSphere Integrated Solutions Console, open **Resources > JDBC > Data sources**.
2. Under **Scope**, select the scope that you selected for the JDBC provider in “Creating a JDBC provider” on page 61 and click **New**.
3. In **Step 1**, enter the data source and JNDI names, and then click **Next**.

---

Table 9. JDBC provider options for DB2 and Derby (continued)

<table>
<thead>
<tr>
<th>Provider type</th>
<th>Derby JDBC Provider</th>
<th>DB2 Universal JDBC Driver Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation type</td>
<td>XA data source</td>
<td>Connection pool data source</td>
</tr>
<tr>
<td>Name</td>
<td>Enter a name, for example Rule Execution Server JDBC Provider</td>
<td>Enter, for example, DB2 Universal JDBC Driver Provider for RES</td>
</tr>
</tbody>
</table>
Important: You can enter any name for the data source, but the JNDI name must be `jdbc/resdatasource`, otherwise Rule Execution Server cannot use the data source.

- For Derby databases: Rule Execution Server datasource and `jdbc/resdatasource`
- For DB2: `resdatasource` and `jdbc/resdatasource`.

4. In Step 2, choose Select an existing JDBC provider, select the JDBC provider that you created in “Creating a JDBC provider” on page 61, and click Next.

5. In Step 3, enter the specific database properties for the data source.

   The following table shows the minimum set of properties to define the supported databases. If the table does not include your driver, check the WebSphere Application Server documentation for more information.

<table>
<thead>
<tr>
<th>Database</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Universal JDBC Driver</td>
<td>• <code>databaseName</code>: Database name if <code>driverType</code> is set to 4, or a locally cataloged database name if <code>driverType</code> is set to 2, for example DSN910GP.</td>
</tr>
<tr>
<td></td>
<td>• <code>driverType</code>: 2 or 4</td>
</tr>
<tr>
<td></td>
<td>If you are working on WebSphere Application Server on Linux for System z, set the <code>driverType</code> to 4. When the <code>driverType</code> is 4, set the following properties:</td>
</tr>
<tr>
<td></td>
<td>• <code>serverName</code>: TCP/IP address or host name</td>
</tr>
<tr>
<td></td>
<td>• <code>portNumber</code>: TCP/IP port number</td>
</tr>
<tr>
<td>DB2 legacy CLI-based Type 2</td>
<td><code>databaseName</code>: for example, Sample.</td>
</tr>
<tr>
<td>Oracle JDBC Driver</td>
<td>URL: for example, <code>jdbc:oracle:oci:@sample</code></td>
</tr>
<tr>
<td>Derby</td>
<td>• <code>databaseName</code>: path to the location of the database files. This directory must not exist already. For example, <code>c:\derbydata</code>.</td>
</tr>
<tr>
<td></td>
<td>• Clear the option Use this data source in container managed persistence (CMP).</td>
</tr>
<tr>
<td></td>
<td>For more information, refer to the Derby documentation.</td>
</tr>
</tbody>
</table>

6. Clear the **Use this data source in container managed persistence (CMP)** option.

7. Click Next.

8. In Step 4, set up any necessary security aliases and then click Next. For DB2, select the DB2 administrator’s authentication alias from the Component-managed authentication alias option.

   In Step 5, a summary of the data source is provided.

9. Click Finish.

10. In the next panel, click **Save** to save the changes to the master configuration.

**Creating J2C authentication data**

In WebSphere Application Server, you can secure your enterprise information system by creating J2C authentication data.
Before you begin

To create J2C authentication data, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 55.

About this task

After you have created your data source and connection pool, you create the J2C authentication data. J2C is a secure mechanism for integrating enterprise information systems to an application server and enterprise applications.

Procedure

1. In the WebSphere Integrated Solutions Console, open Resources > JDBC > Data sources.
2. In the table, click the name of the data source that you created in “Creating a data source and connection pool” on page 62.
   Click the link, do not just select the check box.
   For example, click Rule Execution Server datasource.
3. Under Related Items, click JAAS - J2C authentication data.
4. In the next panel, click New and provide the database administrator credentials in the Alias, User ID, and Password fields.
   For Linux on System z, provide the DB2 credentials. For a Derby database, provide the following credentials:
   - Alias: ResDerbyUser
   - User ID: resdbUser
   - Password: resdbUser
5. Click Apply and Save to save directly to the master configuration.
6. Again, open Resources > JDBC > Data sources and click your data source name, Rule Execution Server datasource in this example.
   <NodeName> is the name of the WebSphere Application Server node on which you are configuring Rule Execution Server.
9. Click Apply and Save to save directly to the master configuration.

Setting custom properties

The database to which you want to connect might require that you set some custom properties.

Before you begin

To set custom properties, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 55.

About this task

Depending on the database to which you want to connect, you must define various properties. The following table presents the minimum set of properties that
are necessary to define the supported databases. If your driver is not listed, check the WebSphere Application Server documentation for more information.

Note: Some of these properties might already be defined. For example, if you are following this configuration procedure and have created a Derby database, all the properties for that database are already defined.

Table 10. Database driver properties

<table>
<thead>
<tr>
<th>Database</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Universal JDBC Driver</td>
<td>• <code>databaseName</code>: The actual database name if <code>driverType</code> is set to 4, or a locally cataloged database name if <code>driverType</code> is set to 2</td>
</tr>
<tr>
<td></td>
<td>• <code>driverType</code>: The possible values are 2 or 4. The following properties are required only if <code>driverType</code> is 4:</td>
</tr>
<tr>
<td></td>
<td>– <code>serverName</code>: The TCP/IP address or host name</td>
</tr>
<tr>
<td></td>
<td>– <code>portNumber</code>: The TCP/IP port number</td>
</tr>
<tr>
<td>DB2 Universal JDBC XA Driver</td>
<td>• <code>databaseName</code>: The locally cataloged database name</td>
</tr>
<tr>
<td></td>
<td>• <code>driverType</code>: The possible values are 2 or 4. If you are running a version of DB2 earlier than DB2 V8.1 FP6, you are restricted to using only Type 2 driver. The following properties are required only if <code>driverType</code> is 4:</td>
</tr>
<tr>
<td></td>
<td>– <code>serverName</code>: The TCP/IP address or host name</td>
</tr>
<tr>
<td></td>
<td>– <code>portNumber</code>: The TCP/IP port number</td>
</tr>
<tr>
<td>DB2 legacy CLI-based Type 2</td>
<td><code>databaseName</code>: for example, Sample</td>
</tr>
<tr>
<td>Oracle JDBC Driver</td>
<td><code>URL</code>: For example, <code>jdbc:oracle:oci:@sample</code></td>
</tr>
<tr>
<td>Derby</td>
<td><code>databaseName</code>: The path to the location of the database files. For more information, refer to the Derby documentation.</td>
</tr>
</tbody>
</table>

You can also set custom properties for any of the resources that are listed in the Custom properties page.

Procedure

1. In the WebSphere Integrated Solutions Console, open Resources > JDBC > Data sources.
2. Click the data source that you want to customize.
3. Under Additional Properties, click Custom properties.
4. Change an existing property or create one by clicking New.
   a. Click `createDatabase`.
      The General Properties page opens.
   b. Type `create` in the Value field.
      For example, for a Derby data source, if you have not created the database yet, you can set the value of the `createDatabase` property to create. The database is created at the first database connection.
5. Click OK and Save to save the changes to the master configuration.

Testing the connection to the database

After you have created a data source and connection pool, and possibly set some custom properties, you can test the connection to your database.
Before you begin

You test the connection to the database from the WebSphere Integrated Solutions Console. Log in to the console as explained in "Before you start: Opening the administration console" on page 55.

Procedure

1. In the Websphere Integrated Solutions Console, open Resources > JDBC > Data sources.
2. Select the check box next to the data source that you want to test and click Test connection.

Results

The status of the connection is indicated at the top. For example:

The test connection operation for data source Rule Execution Server datasource on server server1 at node <NodeName> was successful.

Potential connection errors:

- Java class ..com.ibm.db2.jcc is not found: Make sure that the jcc drivers class path is correctly set.
- Null Userid is not supported – SQL ERRORCODE 4461: Authentication has failed.: Make sure that the JAAS-J2C Authentication credentials are correct.

Step 5: Activating security on WebSphere Application Server

WebSphere Application Server provides security infrastructure and mechanisms to protect sensitive Java EE resources and administrative resources, and to address enterprise end-to-end security requirements on authentication, resource access control, data integrity, confidentiality, privacy, and secure interoperability.

Introduction to WebSphere Application Server security

In WebSphere Application Server, security is organized in layers, from the platform security up to the WebSphere Application Server-specific layer, based on the Java EE model, over the Java security layer.

The following diagram shows the security layers in WebSphere Application Server.
WebSphere Application Server supports the Java EE model for creating, assembling, securing, and deploying applications.

By default, the Rule Execution Server console does not require security in WebSphere Application Server. However, to activate access control for Rule Execution Server in WebSphere Application Server, follow these steps:

1. “Creating users and groups”
2. “Mapping user groups to the Monitor role” on page 69

Creating users and groups

Create users and groups and assign them roles by using a user registry with a federated repository.

Before you begin

To configure a federated repository as a user registry, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 55.

About this task

WebSphere Application Server uses various kinds of user registries: OS, LDAP, or Custom. You control access to Rule Execution Server and enforce security by defining groups and users. If no groups and users are defined yet or if you want to define new groups and users, proceed with the following steps. If suitable groups and users are already defined, skip this procedure and connect users to their appropriate roles when you deploy your applications.

The following table summarizes the main groups and their associated default user and password.
### Group Use Default user/password

<table>
<thead>
<tr>
<th>Group</th>
<th>Use</th>
<th>Default user/password</th>
</tr>
</thead>
<tbody>
<tr>
<td>resAdministrators</td>
<td>Gives a user full administrator rights: • 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</td>
<td>resAdmin - resAdmin</td>
</tr>
<tr>
<td>resDeployers</td>
<td>Gives a user the following rights: • Deploy, browse, and modify RuleApps • Test rulesets</td>
<td>resDeployer - resDeployer</td>
</tr>
<tr>
<td>resMonitors</td>
<td>Gives a user the following rights: • View RuleApps • Monitor decision history and access Decision Center reports</td>
<td>resMonitor - resMonitor</td>
</tr>
</tbody>
</table>

### Procedure

1. In the side panel, click **Security** > **Global security**.
2. Configure the repository security as follows:
   - If **Federated repositories** is already selected under **Current realm definition**, make sure that **Enable application security** is selected under Application security. If you select **Enable application security**, you must click **Apply** and **Save** to save the changes to the master configuration.
   - If **Federated repositories** is not already selected, click **Security Configuration Wizard**, and then complete the wizard as follows:
     a. In **Step 1**, to specify the level of protection, select **Enable application security** and click **Next**.
     b. In **Step 2**, select **Federated repositories** and click **Next**.
     c. In **Step 3**, type a name in the **Primary administrative user name** field and enter **websphere** in the **Password** field, and then click **Next**.
     d. In **Step 4**, review the security configuration summary and click **Finish**.
     e. Click **Save** to save the changes to the master configuration.
     f. Restart WebSphere Application Server.
        Then, you must log in to the WebSphere Integrated Solutions Console as the primary administrative user.
3. In the side panel, click **Users and Groups** > **Manage Groups**, and then click **Create**.
4. Enter **resAdministrators** as the group name, then click **Create**.
5. Click **Create Like**, create another group named **resDeployers**, and click **Create**.
6. Click **Create Like** again, enter another group named **resMonitors**, and click **Create**, then click **Close**.
7. In the side panel, open **Users and Groups** > **Manage Users** and then click **Create**.
8. Enter resAdmin as the User ID and again resAdmin as the password. Also, specify the given name and last name.

9. Click **Group Membership** and proceed as follows:
   a. Click **Search**, select the resAdministrators, resDeployers, and resMonitors groups.
   b. Click **Add**.
   c. Click **Close**, then click **Create** and **Close** again.

10. Click **Create Like** to create users with deployer and monitor roles as follows:
    a. Create another user named resDeployer with password resDeployer.
    b. Assign the user to the resDeployers and resMonitors groups.
    c. Create a user named resMonitor with password resMonitor.
    d. Assign the user to the resMonitors group.

11. Restart your application server or your deployment manager.

**What to do next**

When you create user groups, you might have to map the resAdministrators and resDeployers users to the Monitor role. In this case, see "Mapping user groups to the Monitor role."

**Mapping user groups to the Monitor role**

When you create user groups, you might have to map them to the Monitor role to give them access to the model MBeans.

**Before you begin**

To map groups to roles, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in "Before you start: Opening the administration console" on page 55.

**About this task**

To access the MBeans of the Rule Execution Server model, an application must have sufficient security credentials, restricted to the Monitor role in the WebSphere Application Server authentication system. You can give Rule Execution Server users access to the model MBeans by configuring a mapping between the resAdministrators group or the resDeployers group and the Monitor role. These groups are declared in the custom registry.

**Procedure**

1. In the WebSphere Integrated Solutions Console, open **Users and Groups** > **Administrative group roles**.
2. Click **Add**.
3. From the **Role(s)** list, select **Monitor**.
4. In the **Search string** field, type resAdministrators and click **Search**.
   An entry that begins with resAdministrators is displayed in the Available column.
5. Click the arrow to move the entry from the Available column to the Mapped to role column, and then click **OK**.
6. Optional: Do the same for resDeployers, then click **OK**.
7. In the next panel, click **Save** to save directly to the master configuration.
8. Restart your application server or your deployment manager.

**Security policies for the Rule Execution Server console**

To be able to record and manage sets of MBeans, you must override the WebSphere Application Server security policies for the Rule Execution Server console.

When the global security of WebSphere Application Server is activated, the MBean server is not accessible from the deployed application. You must override these security policies for the Rule Execution Server console so that the console can record and manage a set of MBeans.

Rule Execution Server is packaged with a specific policy file, `was.policy`, which overrides the server policies. The `was.policy` file is packaged in the META-INF directory of the `jrules-res-management-WAS<version_number>.ear` file.

---

**Step 6: Deploying the Rule Execution Server MBean descriptors**

To configure Rule Execution Server for WebSphere Application Server, you must also deploy the MBean descriptors.

**Before you begin**

Make sure that you give the application server process enough access right to read the `jrules-mbean-descriptors.jar` file. For example, change the permissions on the file by using a `chmod 777` command.

**About this task**

The Rule Execution Server architecture is based on the Java Management Extensions (JMX) API. MBeans are Java objects that are used by the JMX API. To configure Rule Execution Server for WebSphere Application Server, you must deploy the MBean descriptors, either globally for all Rule Execution Server instances or for a single Rule Execution Server instance.

**Procedure**

To deploy MBean descriptors:
1. Open the WebSphere Integrated Solutions Console.
2. In the side panel, open **Servers > Server Types > WebSphere application servers**.
3. On the Application servers page, click the name of your server.
4. Under Server Infrastructure, expand **Java and Process Management** and click **Process definition**.
   In WebSphere Application Server for z/OS, an additional layer provides three resources that can be administered: Adjunct, Control, and Servant. If you are working in that environment, select the Servant.
5. Under Additional Properties, click **Java Virtual Machine**.
6. In the **Classpath** field, add `<InstallDir>/executionserver/lib/jrules-mbean-descriptors.jar`.
7. Click **OK**, then **Save** to save the changes directly to the master configuration.
Step 7: Deploying the XU RAR

After you have deployed the MBean descriptors, you deploy the resource adapter archive (RAR) for the execution unit (XU) to WebSphere Application Server.

Before you begin

You deploy the XU resource archive from the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 55.

About this task

This phase of the Rule Execution Server configuration consists in deploying the execution unit (XU) resource adapter archive (RAR) on WebSphere Application Server and add the properties by which to identify it. The RAR file contains the XU and the persistence layer.

In some cases, because of your application constraints, you might have to deploy the XU inside the application. Choose the appropriate deployment mode of the XU: either embed it into the EAR or deploy it as a global connector. Refer to the application server documentation for instructions on packaging a connector into an EAR. In any case, be aware of the following consequences.

• When the XU is deployed as a global connector:
  – The deployed Java EE applications might use its third-party libraries (such as ASM) instead of the libraries that are deployed in the application server.
  – Use the parent last setting for the XU Java EE application if your Java EE application does not support the version of the third-party libraries that is distributed with Decision Server. If you cannot use a parent last setting, you might have to embed the XU into the EAR file that executes the rules.

• If you choose an embedded XU packaging, use the parent last setting for the code library if the version of the third-party libraries that are deployed at the level of the application-server code library is not compatible with the XU.

Procedure

To deploy the XU RAR:

1. In the WebSphere Integrated Solutions Console, open Resources > Resource Adapters, and then click Resource Adapters.
2. In the Resource adapters panel, under Scope, select the Node=xxx, Server=yyy, and then click New.
   xxx is the name of your node and yyy the name of your server.
3. Click Install RAR.
4. In the next panel, make sure that Local file system is selected, browse to the resource archive file, and click Next.
   • <ODM_InstallDir>/executionserver/applicationservers/WebSphere8/jrules-res-xu-WAS8.rar
   • <ODM_InstallDir>/executionserver/applicationservers/WebSphere85/jrules-res-xu-WAS85.rar for WebSphere Application Server 8.5 and 8.5.5.
5. In the General Properties page, enter the XU name.
   a. Set the name for the XU, such as RES XU Resource Adapter.
b. For WebSphere Application Server version 8.5 or 8.5.5, select the **Isolate this resource provider** option.

c. Click **OK**.

6. Back to the Resource adapters page, click **Save**.

7. In the next panel, select **All scopes** from the drop-down list so that the table of resource adapters shows the XU resource that you have just added.

8. Click the new resource name in the table.
   In this example, you named itRES XU Resource Adapter Click the link, do not just select the check box.

9. In the next panel, under Additional Properties, click **J2C connection factories** and **New**.

10. Enter the following values:
    - **Name**: xu_cf
    - **JNDI name**: eis/XUConnectionFactory

    **Note**: eis/XUConnectionFactory is the predefined JNDI name. If you enter a different one, you must map your user-defined name to the predefined eis/XUConnectionFactory name.

11. Click **OK** and **Save** to save the changes to the master configuration.

12. Optional: You can define more than one XU resource adapter.
    You need more XU resource adapters when you have more than one node in your environment or you want to isolate the development and testing environments in one single node. You can deploy a JCA resource adapter at any level, depending on the capability of the application server. On WebSphere Application Server, you can install the XU resource adapter at the cell, node, cluster, or server level. But you must install it at the node level before you can deploy it at other levels.
    For example, to define a XU resource adapter on the server level, follow these steps:
    a. In the side panel, open **Resources** > **Resource Adapters** and click **Resource adapters**, then click **Install RAR**.
    b. Select the scope **Node=xxx, Server=yyy**, where xxx is the name of your node, yyy is the name of your server.
    c. Click **New** and enter the name of the XU as XU.
    d. In **Archive Path**, select **${CONNECTOR_INSTALL_ROOT}/jrules-res-xu-WAS<version_number>.rar**.
    e. For WebSphere Application Server version 8.5, select **Isolate this resource provider**.
    f. Click **OK**.
    g. Repeat steps on page 71 through 11 to define the connection factory.

13. Restart the server.

**Tip:**

Whenever you install or uninstall a XU, you must restart your application server.

**What to do next**

For more information, especially for instructions about packaging a connector into an EAR or about installing extra XU resource adapters, see the WebSphere
Step 8: Deploying the Rule Execution Server management EAR file

After you have activated the security and deployed the XU RAR, you deploy the Rule Execution Server EAR to WebSphere Application Server.

Before you begin

You deploy EAR files from the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 55.

About this task

You deploy the Rule Execution Server management EAR file to WebSphere Application Server to apply the persistence type that you set in a previous step, and you map security users groups.

Procedure

1. In the side panel, open Applications > New Application and click New Enterprise Application.
2. In the next panel, make sure that Local file system is selected, browse to the archive file, and click Next.
   - `<ODM_InstallDir>/executionserver/applicationservers/WebSphere8/jrules-res-management-WAS8.ear` for WebSphere Application Server 8.5.
   - `<ODM_InstallDir>/executionserver/applicationservers/WebSphere85/jrules-res-management-WAS85.ear` for WebSphere Application Server 8.5.5.
3. Select the Detailed - Show all installation options and parameters option.
4. Expand Choose to generate default bindings and mappings, select the check box Generate Default Bindings, and click Next.
5. Scroll down the page and click Continue to accept the security warning.
   The “Select installation options” page opens.
6. In Step 1, click Next at the bottom of the console to accept the default settings.
7. In Step 2, proceed as follows:
   - If you have only one server, click Next to skip Step 2.
   - If you have more than one server, select the server to which you want to deploy the application, and then select the check box for ILOG Rule Execution Server Console and click Next.
8. Proceed as follows:
   - If you are working on WebSphere Application Server for distributed platforms, in Step 3 through Step 8, click Next to accept the default settings.
   - If you are working on WebSphere Application Server for Linux on System z, in Step 6, you map the resource references to JNDI resources.
     a. In Step 3 to Step 5, click Next to accept the default settings.
     b. In Step 6, select the check box for the eis/XUConnectionFactory resource reference.
     c. Click Browse, select the entry named xu_cf, and click Apply.
d. For the jdbc/resdatasource reference, click **Browse**, and select the entry named **resdatasource**.

e. Click **Apply**, and then click **Next**.

9. At the bottom of the Application Resource Warnings page, click **Continue**.

10. In **Step 9: Map security roles to users or groups**, proceed as follows:

   a. Select the check box next to the **resAdministrators** role.

   b. Click **Map Groups** and click **Search**.

      The groups are shown in the Available column.

   c. Click the **resAdministrators** role and click the arrow to move it to the Selected column.

   d. Click **OK** to return to the "Mapping security roles to users or groups" page.

   e. Repeat steps [10a through 10d](#) to map the roles **resDeployers** and **resMonitors** to the other groups.

      Make sure that only the check box next to the role that you are assigning is selected.

<table>
<thead>
<tr>
<th>Role</th>
<th>Mapped groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>resAdministrators</td>
<td>resAdministrators</td>
</tr>
<tr>
<td>resDeployers</td>
<td>resDeployers</td>
</tr>
<tr>
<td>resMonitors</td>
<td>resMonitors</td>
</tr>
</tbody>
</table>

11. Click **Next**.

   - On WebSphere Application Server 7, **Step 10** provides a summary.
   - On WebSphere Application Server 8, proceed as follows: In **Step 10** and **Step 11**, click **Next** to accept the default settings.

   **Step 12** provides a summary.

12. Click **Finish**.

13. After the installation has completed, click **Manage Applications** at the bottom of the page.

14. Click **ILOG Rule Execution Server**.

   Click the link, do not only select the check box.

15. In the Configuration tab, click **Manage Modules**.

16. Click **ILOG Rule Execution Server Console**.

   Click the link, do not only select the check box.

17. Under General Properties, for **Class loader order**, select **Classes loaded with local class loader first (parent last)**.

18. Click **OK**, and in the next panel, click **Save** to save directly to the master configuration.

19. In the side panel, open **Applications** > **Application Types** > **WebSphere enterprise applications**.

20. In the Enterprise Applications page, select the check box next to **ILOG Rule Execution Server** and click **Start** to start the application.

---

**Step 9: Creating a Rule Execution Server database schema**

You can create the Rule Execution Server database schema by running SQL scripts, either from the Rule Execution Server console or from the SQL tool of your database.
Creating a database schema using the Rule Execution Server console

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

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

Before you use the Installation Settings wizard, you must know the credentials of the database.

To configure the database to store managed Java XOM, you follow the procedure twice, with a different target database each time.

The Installation Settings wizard creates all the required tables for Rule Execution Server and for Decision Validation Services, even if you do not have a license for Decision Validation Services.

Note: 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 the database tables are created, you must run the SQL scripts in the database client.

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

- If RuleApp persistence and Java XOM persistence are both set to file, no wizard is presented.
- If RuleApp persistence and Java XOM persistence are both set to datasource, the process is in two steps:
  1. When you sign in to the Rule Execution Server console, the RuleApp persistence details part of the Installation Settings wizard is displayed for you to create the schema for RuleApps and the Decision Warehouse trace.
  2. After you have created the schema, the Java XOMs persistence details part of the wizard is displayed for you to configure the database for Java XOM persistence.
- If RuleApp persistence is set to file and Java XOM persistence is set to datasource, you see only Java XOMs persistence details and you cannot use the Decision Warehouse.
- If RuleApp persistence is set to datasource and Java XOM persistence is set to file or is not defined, when you sign in to the Rule Execution Server console, only RuleApp persistence details is displayed for you to create the schema for RuleApps. The Java XOMs persistence details part of the wizard is not shown.

The following table summarizes the cases.
Opening the Rule Execution Server console
To open the Rule Execution Server console and the Installation Settings wizard, you must sign in with resAdministrators rights.

Procedure
1. Start your database, if the persistence type is datasource or jdbc.
2. Open the Rule Execution Server console in a web browser by typing res at the root URL on the host computer:
   - If your browser is not running on the same host as the application server, replace localhost with the address of the computer.
   - If the web application is mapped to a host on a port that is different from the default of the server, change the port number to the host port number.
3. Sign in to the Rule Execution Server console as the administrator.
   For example, use resAdmin and resAdmin for both User ID and Password.

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

Step 1: Welcome to the Installation Settings wizard
If you open the Rule Execution Server console with datasource as the persistence setting and an empty database schema, the Installation Settings wizard opens.

The wizard can display two parts:
- If you set datasource persistence for RuleApps, it starts with RuleApp persistence details, whatever the persistence type for managed Java XOMs.
- If you set datasource persistence for both RuleApps and Java XOMs, Java XOMs persistence details is displayed after RuleApp persistence details. In this case, you go through the same steps twice.
- If you set the persistence type to file for RuleApps and to datasource for managed Java XOMs, the wizard starts with Java XOMs persistence details.

Both parts of the wizard are similar and you use them in the same way:
1. Click Next after you have read the Welcome page.
   The Welcome page displays general information:
   - Persistence details about the type of database used, including information about the driver and JDBC URL.
   - A brief description of the purpose of the Installation Settings wizard.
A diagnostic report that explains why the persistence check failed (because you have not created the database tables yet).

2. “Step 2: Choose the database schema”
3. “Step 3: Review the database schema”
4. “Step 4: Using the Installation Settings wizard report” on page 78

**Step 2: Choose the database schema**

After you have read the Welcome page, you choose the database schema. Specific settings are available for DB2. Optionally, you can select a customized SQL script.

**Procedure**

To choose the database schema:

1. In the **Database schema selected** field, select an available database schema type.
   
   A type that corresponds to the type of database that you use is selected by default, but you can choose another type from the drop-down list.
   
   If you select a db2 or db2_os390 schema, an extra field displays so that you can enter the name of the buffer pool, which is used to create the Decision Warehouse table space. This buffer pool must have a page size of 32 K. Check the DB2 documentation for information about how to create a 32 K 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.

2. Optional: If you want to use a customized SQL script, select **custom** 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 type, you confirm the creation of a schema for Rule Execution Server. You can also use SQL drop statements that flush data from an existing table and view the SQL statements.

**Before you begin**

Before you use this option, ensure that you have a backup of database resources.

**Procedure**

To confirm the creation of a schema:

1. Select from the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create SQL schema resdbUser</td>
<td>Select this option to run the SQL statement for the schema type selected in the previous step.</td>
</tr>
</tbody>
</table>
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keep drop SQL statements</td>
<td>Select this option to flush data from an existing Rule Execution Server database.</td>
</tr>
<tr>
<td>Show SQL statements</td>
<td>Select this option to display the SQL statements.</td>
</tr>
</tbody>
</table>

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

### Step 4: Using the Installation Settings wizard report

After you have confirmed the database schema, the Installation Settings wizard creates the schema. A report shows the status of the schema creation.

#### Procedure

To use the options in the report:

1. Click **Show execution details** to view the list of SQL statements executed.
2. Click **Finish**.

   If you have just worked in **RuleApps persistence details** and the persistence setting for managed Java XOMs is datasource, the **Java XOMs persistence details** part of the wizard displays now for you to repeat the procedure.

### Creating the database schema by running SQL scripts

After you create an empty database, you create the schema for the Rule Execution Server database. One way of doing so consists in running SQL scripts.

#### About this task

The script that creates the database schema is named `repository_<DatabaseName>.sql`. This procedure shows the Derby and DB2 examples.

#### Note:

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

If you use Command Editor to run the scripts, you must log in with the credentials that you use for the data source for Rule Execution Server. To access the database, the database user must have the following rights:

- A user ID
- A password
- Complete privileges on the tables and view of the schema (create, insert, delete)
- Privileges for index creation (create index)
- On Oracle, additional creation privileges: create trigger and create sequence.

Use any tool that can handle SQL to import and run the SQL scripts. The following table lists the SQL tools for each database.
The following example shows how to run the Derby SQL script to create the schema. It is assumed that the embedded version of Derby is used.

**Procedure**

1. Stop the application server.
2. Connect to the database.
   - For example, to create and connect to the database `c:/resdb` as the user `resdbUser`, use the command:
     ```
     ij>connect 'jdbc:derby:c:/resdb;user=resdbUser;password=resdbUser;';
     ```
   - If the script runs 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 necessary database schema by running the `trace_derby.sql` script. For more information, see [Additional steps to configure Decision Validation Services](#).

3. Run the script that creates the database schema.
4. Close the **ij** utility.
   ```
   ij> quit;
   ```
5. Start the application server.

**Running SQL scripts to create a DB2 schema**

If you work with a DB2 database, you can use SQL scripts from an SQL Plus client and specific constraints apply.

**About this task**

The script that creates the DB2 database schema is named `repository_db2.sql`. You can also create the database schema for Decision Warehouse by running the script `trace_db2.sql`. If you are storing the Java XOM in the database, you must create these tables by running the `xomrepository_db2.sql` script.

When you use DB2, the scripts that create the Rule Execution Server database tables are written for databases that use automatic storage.

**Important:** Users who work with DB2 on z/OS as the back-end database for the rules at run time must apply a fix to a database storage overflow problem. This fix is particularly important for the Decision Warehouse trace table, which can contain a lot of high-volume LOB data. You can find information about this fix on the [IBM support web site](#).
The following constraints apply:

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

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

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

  ```sql
  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 `tablespace`:

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

Here is an example of the `tablespace` specification in the script:

  ```sql
  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, depending on your database settings.

**Procedure**

1. Connect to the Rule Execution Server database:

   ```sql
   CONNECT TO{RESDBName}
   QUIT
   ```

2. Navigate to the directory `<ODM_InstallDir>/executionserver/databases`.

3. Run the following command to create the Rule Execution Server tables:

   ```sql
   db2 -tvf repository_db2.sql
   ```

4. Optional: Run the following command to create the Decision Warehouse tables:

   ```sql
   db2 -tvf trace_db2.sql
   ```

5. Optional: Run the following command to create the Java XOM tables:

   ```sql
   db2 -tvf xomrepository_db2.sql
   ```

6. Disconnect from the database:

   ```sql
   db2 DISCONNECT {RESDBName}
   ```

7. Exit the `{DB2AdminName}` UserID:

   ```sql
   EXIT
   ```

**Testing the connection to the database**

After you have created a data source and connection pool, and possibly set some custom properties, you can test the connection to your database.

**Before you begin**

You test the connection to the database from the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 55.

**Procedure**

1. In the WebSphere Integrated Solutions Console, open **Resources** > JDBC > **Data sources**.

2. Select the check box next to the data source that you want to test and click **Test connection**.
Results

The status of the connection is indicated at the top. For example:
The test connection operation for data source Rule Execution Server datasource
on server server1 at node <NodeName> was successful.

Potential connection errors:
- Java class ..com.ibm.db2.jcc is not found: Make sure that the jcc drivers
class path is correctly set.
- Null Userid is not supported — SQL ERRORCODE 4461: Authentication has
failed.: Make sure that the JAAS-J2C Authentication credentials are correct.

Step 10: Deploying the hosted transparent decision service EAR file

If you want to enable hosted transparent decision services, you deploy the specific
EAR file on the same node as the execution unit (XU).

Before you begin

To deploy the archive for transparent decision services, you work from the
WebSphere Integrated Solutions Console. Log in to the console as explained in
"Before you start: Opening the administration console" on page 55.

About this task

Make sure that you deploy the archive for hosted transparent decision service on
the same node as the execution unit (XU).

Procedure

1. In the side panel, click Applications > New Application and New Enterprise
Application.
2. In the next panel, make sure that Local file system is selected and browse to
the archive file, and then click Next.
   - <ODM_InstallDir>/executionserver/applicationservers/WebSphere8/
     jrules-res-htds-WAS8.rar
   - <ODM_InstallDir>/executionserver/applicationservers/WebSphere85/
     jrules-res-htds-WAS85.rar for WebSphere Application Server 8.5 and 8.5.5.
3. Select the check box Detailed - Show all installation options and parameters.
   a. Expand Choose to generate default bindings and mappings.
   b. Select the check box Generate Default Bindings.
   c. Click Next.
4. Click Continue to accept the security warning.
5. For Step 1 to Step 10, click Next to accept the default settings.
   Step 11 provides a summary.
6. Click Finish.
7. After the installation is finished, click Save to save directly to the master
configuration.
8. In the side panel, open Applications > Application Types > WebSphere
terprise applications and click jrules-res-htds.
   Click the link, do not just select the check box.
9. Click Manage Modules.
10. Click **DecisionService**.
   
   Click the link, do not just select the check box.

11. Under General Properties, for **Class loader order**, select **Classes loaded with local class loader first (parent last)**, and click **OK** and **Save**.
   
   You are back on the Manage Modules page.

12. Click **OK** again.
   
   You are back on the Enterprise Applications page. If not, open Applications > Application Types > WebSphere enterprise applications.

13. Select the check box next to **jrules-res-htds**, and then click **Start** to start the application.

**What to do next**

1. Set the web container custom property **DecodeUrlAsUTF8** to **false** to support a localized ruleset path.

2. Set the **ruleset.xmlDocumentDriverPool.maxSize** ruleset property to the appropriate value. See **Setting the ruleset.xmlDocumentDriverPool.maxSize property**.

---

**Step 11: Setting the DecodeUrlAsUTF8 custom property**

If you have deployed the hosted transparent decision service EAR file to enable those services, you must set the web container custom property **DecodeUrlAsUTF8** to **false** to support a localized ruleset path.

**Before you begin**

You set this custom property in the WebSphere Integrated Solutions Console. Log in to the console as explained in **Before you start: Opening the administration console** on page 55.

**Procedure**

To set the **DecodeUrlAsUTF8** web container custom property:

1. In the side panel, click Servers > Server Types > WebSphere application servers, and then click the server name.

2. Under Container Settings, click Web container settings > Web container.

3. Under Additional Properties, click Custom properties.

4. In the next panel, click **New** and then type **DecodeUrlAsUTF8** as the name and **false** as the value.

5. Click **Apply** and **Save** to save directly to the master configuration.

---

**Optional configuration steps**

After completing the steps to configure Rule Execution Server, you can enhance your configuration by adding support for WebSphere MQ for example, or by setting up a multiserver configuration.

**Integrating WebSphere MQ in WebSphere Application Server to support asynchronous execution**

If you use the WebSphere MQ messaging provider support in WebSphere Application Server, you can deploy Java EE applications that directly use the enterprise messaging capabilities of WebSphere MQ.
**WebSphere MQ integration overview**

The Java Message Service (JMS) API enables access to rule services. To use a message-driven rule bean, you must create the necessary resources under the WebSphere MQ JMS provider. Both the publish-and-subscribe and the point-to-point models are supported.

A Decision Server rule service can be accessed by an asynchronous invocation pattern that uses the Java Message Service (JMS) API. When a JMS message arrives, the EJB container calls a message-driven rule bean (MDB). The MDB can reside locally or remotely from the client application. In turn, the message-driven rule bean calls the rulesets that are running in the execution unit (XU). The real call to the rule engine is delegated to a simple rule session.

In WebSphere Application Server, the client application is the scenario running in WebSphere Application Server that calls the rule service, the server is the application server where Rule Execution Server is installed. Rule Execution Server is usually running remotely to the client application.

To use a Decision Server message-driven rule bean, you must create the necessary resources under the WebSphere MQ JMS provider at the proper scope for both the client and the server side to make them visible for the client application and the Decision Server MDB respectively. Decision Server implements both standards of asynchronous messaging: the Publish-and-Subscribe Model and the Point-to-Point Model. The following procedure demonstrates how to set up both a point-to-point messaging model and a publish-and-subscribe model. If you need only one of them, you can comment out the resource reference in the deployment descriptor of Decision Server MDB.

Before installation, you must create the following resources in WebSphere MQ:

- **Queue**: JRulesIn, JRulesOut
- **Topic**: JRulesTopicIn, JRulesTopicOut

Use these resources to submit rule execution requests and obtain execution results.

You must perform the following tasks to integrate WebSphere MQ in WebSphere Application Server:

**Creating the WebSphere MQ queue connection factory:**

To create the WebSphere MQ queue connection factory, you create the queue, then you configure the connection factory by following the connection factory wizard.

**Procedure**

To create a queue connection factory to connect WebSphere MQ with the Decision Server MDB:

1. Log in to the Integrated Solutions Console.
2. Open **Resources** > **JMS** and click **JMS Providers**.
3. In the panel, select the **WebSphere MQ messaging provider**.
4. Under **Additional Properties**, click **Queue connection factories**.
5. Click **New**.
6. In **Step 1: Configure basic attributes**, set the fields **Name** and **JNDI name** as follows, then Click **Next**.
Name  JRules Queue Connection Factory

JNDI name
jms/BRESQueueConnectionFactory

7. In Step 2: Select connection method, select Enter all the required information into this wizard and click Next.

8. In Step 2.1: Supply queue connection details, type the name of your queue manager or queue sharing group, then click Next.

9. In Step 2.2: Enter connection details, type the connection details to establish a connection to the queue manager or queue sharing group, then click Next.
   The default queue port is 1414.

10. In Step 3: Test connection, click Test connection.
    If your message queue is running, you see the following message:
    A connection was successfully made to WebSphere MQ.

11. Click Next.
    A summary opens showing the details of the connection factory.

12. Click Finish and then click Save to save directly to the master configuration.

Creating the WebSphere MQ input queue:

After you have created the WebSphere MQ queue connection factory, you can now create the JMS queue destination for receiving a request message. To do so, you set the scope to node or server level, you select the provider, name the queue, and finally save the configuration.

Procedure

To create the JMS queue:
1. In the Integrated Solutions Console, open Resources  >  JMS and click Queues.
2. Set the scope to either Node level or Server level, then click New.
3. In the panel, select WebSphere MQ messaging provider, then click OK.
4. In General Properties, set the fields Name, JNDI name, and Queue name as follows:
   
   Name  JRules Input Queue
   JNDI name  jms/BRESQueueIn
   Queue name  JRulesIn
5. Type in the name of your queue manager or queue sharing group name, then click OK.
6. Click Save to save directly to the master configuration.

Creating the WebSphere MQ output queue:

After you have created the input queue for request messages, you must also create the JMS queue destination for sending a response message. To do so, you set the scope to node or server level, select the provider, name the queue, and save.
Procedure

To create the output queue:
1. In the Integrated Solutions Console, open Resources > JMS and click Queues.
2. Set the scope to either Node level or Server level, then click New.
3. In the panel, select WebSphere MQ messaging provider, then click OK.
4. In General Properties, set the fields Name, JNDI name, and Queue name as follows:
   - Name: JRules Output Queue
   - JNDI name: jms/BRESQueueOut
   - Queue name: JRulesOut
5. Type in the name of your queue manager or queue sharing group name, then click OK.
6. Click Save to save directly to the master configuration.

Creating a topic connection factory:

After you have created the queue factory, input queue, and output queue, you must create a topic connection factory. To do so, you set the scope to node or server, select the provider and JNDI name, enter the connection details, test the connection, and save.

Procedure
1. In the Integrated Solutions Console, open Resources > JMS and click Topic connection factories.
2. Set the scope to either Node level or Server level, then click New.
3. In the panel, select WebSphere MQ messaging provider, then click OK.
4. In Step 1: Configure basic attributes set the fields Name, and JNDI name as follows and click Next.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>JRules Topic Connection Factory</td>
</tr>
<tr>
<td>JNDI name</td>
<td>jms/BRESTopicConnectionFactory</td>
</tr>
</tbody>
</table>

5. In Step 2: Select connection method, select Enter all the required information into this wizard and click Next.
6. In Step 2.1: Supply queue connection details, type the name of your queue manager or queue-sharing group, then click Next.
7. In Step 2.2: Enter connection details, type the connection details to establish a connection to the queue manager or queue sharing group (the default queue port is 1414), then click Next.
8. In Step 3: Test connection, click Test connection.
   If your message queue is running, you see the following message: A connection was successfully made to WebSphere MQ.
9. Click Next. A summary opens showing the details of the connection factory.
10. Click Finish, then click Save to save directly to the master configuration.
Creating the WebSphere MQ input topic:

After you have created the WebSphere MQ topic connection factory, you can create the JMS topic destination for receiving a request message. To do so, you set the scope to node or server level, select the provider, set the JNDI and input topic names, and save.

Procedure
1. In the Integrated Solutions Console, open Resources > JMS and click Topics.
2. Set the scope to either Node level or Server level, then click New.
3. In the panel, select WebSphere MQ messaging provider, then click OK.
4. In General Properties, set the fields Name, JNDI name, and Topic name as follows, then click OK.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>JRules Input Topic</td>
</tr>
<tr>
<td>JNDI name</td>
<td>jms/BRESTopicIn</td>
</tr>
<tr>
<td>Topic name</td>
<td>JRulesTopicIn</td>
</tr>
</tbody>
</table>

5. Click Save to save directly to the master configuration.

Creating the WebSphere MQ output topic:

After you have created the WebSphere MQ topic connection factory and input topic, you must also create the JMS queue destination for sending a response message. To do so, you set the scope to node or server level, select the provider, set the JNDI and output topic names, and save.

Procedure
To create the JMS output topic:
1. In the Integrated Solutions Console, open Resources > JMS and click Topics.
2. Set the scope to either Node level or Server level, then click New.
3. In the panel, select WebSphere MQ messaging provider, then click OK.
4. In General Properties, set the fields Name, JNDI name, and Topic name as follows and then click OK.

Name

    JRules Output Topic

JNDI name

    jms/BRESTopicOut

Topic name

    JRulesTopicOut

5. Click Save to save directly to the master configuration.

Creating the WebSphere MQ queue activation specification:

After you have configured WebSphere MQ queues, you create the queue activation specification.
**About this task**

The queue activation specification manages the relationship between the Decision Server message-driven rule beans (MDB) running in WebSphere Application Server and a destination in WebSphere MQ. To create the activation specification, you set the scope to node or server level, select the provider, set the specification and JNDI names, enter the connection details, and save.

**Procedure**

To create the activation specification:

1. In the Integrated Solutions Console, open **Resources > JMS** and click **Activation specifications**.
2. Set the scope to either Node level or Server level, then click **New**.
3. In the panel, select **WebSphere MQ messaging provider**, then click **OK**.
4. In Step 1: Configure basic attributes set the fields **Name**, and **JNDI name** as follows and then click Next.
   
   **Name**
   
   JRules Activation Spec
   
   **JNDI name**
   
   eis/I1rRuleExecutionEJB
   
5. In Step 1.1: Specify MDB destination data, set the field **Destination JNDI name** to jms/BRESQueueIn, set the Destination type to **Queue**, then click **Next**.
6. In Step 2: Select connection method, select **Enter all the required information into this wizard** and click **Next**.
7. In Step 2.1: Supply queue connection details, type the name of your queue manager or queue sharing group, then click **Next**.
8. In Step 2.2: Enter connection details, type the connection details to establish a connection to the queue manager or queue sharing group, then click **Next**.
   
   The default queue port is 1414.
9. In Step 3: Test connection, click **Test connection**.
   
   If your message queue is running, you see the following message:
   
   A connection was successfully made to WebSphere MQ.
10. Click **Next**.
    
    A summary opens showing the details of the connection factory.
11. Click **Finish**, then click **Save** to save directly to the master configuration.

**Creating the WebSphere MQ topic activation specification:**

After you have created the queue activation specification, you create the topic activation specification.

**About this task**

The topic activation specification manages the relationship between the Decision Server message-driven rule beans (MDB) running in WebSphere Application Server and a destination in WebSphere MQ. To create the activation specification, you set the scope to node or server level, select the provider, set the specification and JNDI names, enter the connection details, and save.
Procedure

To create the topic activation specification:
1. In the Integrated Solutions Console, open Resources > JMS and click Activation specifications.
2. Set the scope to either Node level or Server level, and then click New.
3. In the panel, select WebSphere MQ messaging provider, and then click OK.
4. In Step 1: Configure basic attributes set the fields Name, and JNDI name as follows, and then click Next.
   
   Name   JRules Topic Activation Spec
   JNDI name   eis/IlrRuleExecutionTopicEJB
5. In Step 1.1: Specify MDB destination data set the field Destination JNDI name to jms/BRESTopicIn, set the Destination type to Topic, and then click Next.
6. In Step 1.2: Configure Durable Subscription, select Nondurable subscription, and then click Next.
7. In Step 2: Select connection method, select Enter all the required information into this wizard and click Next.
8. In Step 2.1: Supply queue connection details, type the name of your queue manager or queue sharing group, and then click Next.
   The default queue port is 1414.
9. In Step 2.2: Enter connection details, type the connection details to establish a connection to the queue manager or queue sharing group, and then click Next.
10. In Step 3: Test connection, click Test connection.
    If your message queue is running, you see the following message:
    A connection was successfully made to WebSphere MQ.
11. Click Next.
    A summary opens showing the details of the connection factory.
12. Click Finish, and then click Save to save directly to the master configuration.

Installing the message-driven rule bean:

After you have created the queue and topic activation specifications, you install the Decision Server message-driven rule bean (MDB) in WebSphere Application Server as an enterprise application. To do so, you create a new enterprise application.

Procedure

To install the Decision Server message-driven rule bean:
1. Open the Integrated Solutions Console.
2. In the panel, open Applications > New Application and click New Enterprise Application.
3. In the panel, select Local file system and Browse to the following path:
   &lt;InstallDir&gt;/executionserver/applicationservers/WebSphere&lt;version_number&gt;/jrules-res-mdb-WAS&lt;version_number&gt;.jar.
4. Click Next.
5. Select the check box Detailed - Show all installation options and parameters.
6. Expand **Choose to generate default bindings and mappings** and select the check box **Generate Default Bindings**.

7. Click **Next**, then click **Continue** to accept the security warning.

8. Click **Step 5: Bind listeners for message-driven beans**.
   a. Type `jms/BRESTopicIn` as the Destination JNDI name for `IlrRuleExecutionTopicEJB`.
   b. Type `jms/BRESQueueIn` as the Destination JNDI name for `IlrRuleExecutionEJB`.
   c. Click **Next**.

9. Click **Step 6: Map resource references to resources** and then click **Next**.
   Use the default binding for the referenced resources.
   **Step 7** provides a summary.

10. Click **Finish**.

11. When the installation has completed, click **Save** directly to the master configuration.

12. In the Integrated Solutions Console, open Applications > Application Types > WebSphere enterprise applications.

13. In the Enterprise Applications page, select the check box next to `jrules-res-mdb-WAS<version_number>.jar` and click **Start** to start the application.

**Enabling server-wide Last Participant Support:**

To complete the integration of WebSphere MQ, you enable Last Participant Support.

**About this task**

To finish integrating WebSphere MQ in WebSphere Application Server for asynchronous execution, you enable Last Participant Support (LPS) so that a single one-phase commit resource is used with any number of two-phase commit resources in the same global transaction. To do so, you set the `ACCEPT_HEURISTIC_HAZARD` custom property to `true` in the Integrated Solutions Console and then restart the application server.

**Procedure**

To enable server-wide LPS:

1. Open the **Integrated Solutions Console**.
2. Click **Servers > Server Types > WebSphere application servers** and the server name.
   The properties page for the application server opens
3. Under **Container Settings**, expand **Container Services** and click **Transaction Service**.
   The properties page for the transaction service opens.
4. Under **Additional Properties**, click **Custom properties**.
5. On the Custom Properties page, click **New** and type `ACCEPT_HEURISTIC_HAZARD` as the **Name** and `TRUE` as the **Value**.
6. Click **Apply** and **Save** directly to the master configuration.
7. Restart your application server.
Configuring Rule Execution Server in 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. In the cases where you configure Rule Execution Server in a single cell, it is good practice to isolate the rulesets that you use on each server and ensure that the execution units (XUs) do not interfere with each other.

Procedure

1. Set up different data sources.
   Use unique JNDI names. For example: jdbc/resdatasourceEnv1 and jdbc/resdatasourceEnv2
2. Deploy a XU for each environment and define a J2C connection factory.
   a. In the side panel, open Resources > Resource Adapters > J2C connection factories and click the name of the connection factory xu_cf. For more information about creating a connection factory, see Step 7: Deploying the XU RAR.
   b. Modify the JNDI name to eis/XUConnectionFactoryEnv1.
      Remember: The predefined JNDI name is eis/XUConnectionFactory. When you choose a different one, you must modify the execution components that call this XU so that they use this JNDI instead of the predefined one.
   c. Under Additional Properties, click Custom properties.
   d. Click the plugins property.
   e. In the Value field, change xuName=default in the property to xuName=xuEnv1, and then click OK.
   f. Click the persistenceProperties property.
   g. In the Value field, change JNDI_NAME=jdbc/resdatasource to JNDI_NAME=jdbc/resdatasourceEnv1.
   h. Click OK and Save to save the changes to the master configuration.
   i. Repeat the entire process for XUs in other environments.
3. Deploy the Rule Execution Server console for each environment.
   a. To modify the deployment descriptor of the Rule Execution Server console EAR file: in the web.xml file, uncomment the JMX_XU_QUERY_PART parameter and specify xuName=xuEnv1.
   b. Deploy the Rule Execution Server console EAR on the server in the resource reference settings in the application server.
      1) Set the JNDI for the data source to jdbc/resdatasourceEnv1.
2) Set the JNDI name for the XU to eis/ConnectionFactoryEnv1.

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

4. Restart the node agents after you complete the configuration.

5. Call the XU instances to register the XU with the Rule Execution Server console.

Verifying the deployment and configuration

It is good practice to verify that Rule Execution Server is successfully deployed and configured by running the diagnostics.

About this task

Important:

- To let a scalable number of users access resources through the Java components, JCA assigns the task of implementing connection pooling to application server vendors.
- If the diagnostics are performed before any execution units (XU) are started, the test is passed and a message reports that no Execution Unit (XU) are initialized.
- On WebSphere Application Server, the pool size is not instantiated beforehand and therefore prevents the server diagnostics from validating a Rule Execution Server before the first execution of a rule engine. The diagnostics remain useful to validate a configuration, especially in a cluster, and to check which execution units are registered with the management model.

Procedure

1. Open the Rule Execution Server console by typing res at the root URL on the host machine:
   
   http://<machine_name>:<PORT>/res
   
   If security is enabled, the prefix is https://.
   
   To find the value of <PORT>:
   
   a. In the Integrated Solutions Console, click Servers > Server types > WebSphere application servers.
   
   b. Select the name of your server.
   
   c. Under Communications, click Ports.

2. Sign in to the Rule Execution Server console.

3. Click the Diagnostics tab.

4. Click Run Diagnostics.

Results

You see a report listing the diagnostic tests. A check mark is shown next to each test to indicate whether the test is successful. Click Expand All to show more details about the tests.
Chapter 3. Configuring Decision Server Events on WebSphere Application Server

The following sections describe how to configure Decision Server Events after installation.

Before you begin

You must have installed Decision Server Events and optionally Decision Center. See Installing.

About this task

The following components are available to you in Decision Server Events:

Table 11. Decision Server Events components.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Default installation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Capture and Replay widgets</td>
<td>In the Event Capture widget and the Event Replay widget, you capture events from a production system, and replay a sequence of one or more of them, typically on a test system.</td>
<td>Yes</td>
</tr>
<tr>
<td>Event Connectors</td>
<td>Event connectors provide data connections between the event runtime and external systems. You can configure the event connectors in Event Designer.</td>
<td>Yes</td>
</tr>
<tr>
<td>Event Designer</td>
<td>A development environment based on Eclipse to design, develop, test, deploy, and monitor event applications. Event Designer is available from the Event perspective in Eclipse.</td>
<td>Yes</td>
</tr>
<tr>
<td>Event Runtime</td>
<td>The event runtime is an execution platform that manages real-time business event coordination. The event runtime requires WebSphere eXtreme Scale and can be deployed and configured on WebSphere Application Server.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 11. Decision Server Events components. (continued)

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Default installation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Tester widget</td>
<td>The Event Tester Widget provides a way to test the event logic in a business process. This widget is aimed at testing and is used only on a test installation. Do not use it on a production installation as it might affect the performance.</td>
<td>Yes</td>
</tr>
<tr>
<td>Integration Components</td>
<td>You can install integration components into the tooling of WebSphere ESB and WebSphere Message Broker to allow these products to send and receive data from Decision Server Events.</td>
<td>No</td>
</tr>
<tr>
<td>Samples and Tutorials</td>
<td>Installs the projects for the samples and tutorials, and provides a server profile to run the samples and tutorials on WebSphere Application Server. For more information about the sample server, see Using the sample server.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

If you install Decision Center, the following chart widgets are also available to you:
- Event Chart Manager to create and edit charts.
- Event Chart to view charts. Chart data is periodically refreshed to present a real-time view of system activities.
- Event Layout to view legacy layouts.

See Accessing event widgets.

One of the key concepts in the following sections is the WebSphere Application Server profile, which you use for planning and configuring your Decision Server Events environment. A profile defines the runtime environment and includes all the files that the server processes in the runtime environment and that you can change. Because you create a profile, you also create an event runtime. Decision Server Events supports three types of profile:
- **Application server profile**: An application server profile defines a separate stand-alone WebSphere Application Server application server that has its own administrative interface and enables you to make applications available to external websites or intranet websites, depending on the applications and server configurations. This profile is ideal for single server environments. For this type of profile, you can:
  - Create a new Decision Server Events application server profile. For more information, see "Creating a new Decision Server Events application server profile" on page 98.
  - Augment an existing WebSphere Application Server application server profile with Decision Server Events. For more information, see "Augmenting an existing WebSphere Application Server application server profile with Decision Server Events" on page 102.
• **Management profile**: A management profile creates a deployment manager, which is a server that manages operations for a logical group of other servers and is the central location for administering the servers and clusters in the cell. If you are setting up a network deployment environment, you must create this profile first. For this type of profile, you can augment only an existing WebSphere Application Server management profile with Decision Server Events. For more information, see “Augmenting an existing WebSphere Application Server management profile” on page 32.

• **Custom profile**: A custom profile provides an empty node that does not contain an administrative console or servers. The typical use for a custom profile is to federate its node to a deployment manager. After federating the node, you can use the deployment manager to create a server or a cluster of servers within the node. For this type of profile, you can augment only an existing WebSphere Application Server custom profile with Decision Server Events. For more information, see “Augmenting an existing WebSphere Application Server custom profile” on page 38.

### Planning your Decision Server Events environment

There are a number of factors that you might want to consider when you decide how to configure your Decision Server Events environment. For example, you must decide whether to configure a stand-alone or clustered Decision Server Events environment.

#### About this task

- “Creating a Decision Server Events environment”
- “Choosing a Decision Server Events topology” on page 96
- “Customizing the event runtime” on page 96

### Creating a Decision Server Events environment

#### About this task

To create a Decision Server Events environment, you can either create your own profile or you can use the sample server provided (if installed). The sample server provides a preconfigured single server Decision Server Events environment by using embedded Apache Derby as the database provider and WebSphere Application Server default messaging as the messaging provider. For more information about using the sample server, see Using the sample server.

#### Procedure

In the following procedure, you create your own profile without using the preconfigured sample server.

1. Install Decision Server Events. See Installing a custom configuration or Installing using Installation Manager.
2. Create a WebSphere Application Server profile augmented with Decision Server Events. See Creating a new Decision Server Events application server profile or Augmenting an existing WebSphere Application Server application server profile with Decision Server Events.
3. Follow Accessing event widgets to access the widgets.
4. Perform any additional customization to your event runtime environment. For more information, see “Customizing the event runtime environment” on page 108.
Results

You have a running Decision Server Events environment. To verify that your environment has been configured correctly, see “Verifying your Decision Server Events configuration” on page 131.

Choosing a Decision Server Events topology

About this task

You can configure Decision Server Events in single-server or clustered environment by using the corresponding topology, depending on your requirements.

Procedure

- **Configure a single server environment**: For simplicity and ease of deployment, you might want to configure a single server Decision Server Events environment. This topology is the simplest topology to configure and manage, but it is not highly available or scalable. See “Configuring a stand-alone Decision Server Events environment” on page 98.

- **Configure a clustered environment**: If you want to configure a highly available and scalable Decision Server Events environment, you must cluster the event runtime. Depending on whether you want to use WebSphere Application Server default messaging as your messaging provider or WebSphere MQ as the messaging provider, you can select from the following two topologies:
  - **Gold Topology**: A gold topology cluster provides high availability and scalability for the event runtime, so that the topology can be configured to be resistant to server failure. Adding more servers to the cluster can improve event throughput in the event runtime, because workload is distributed between multiple cluster members. If the server fails, workload is redistributed to another cluster member and processing continues without any outage. The cluster uses WebSphere Application Server default messaging as the messaging provider for the event runtime. See “Configuring a gold topology cluster” on page 42.
  - A topology that uses WebSphere MQ as the messaging provider:
    You can cluster Decision Server Events to provide high availability and scalability for the event runtime, but by using WebSphere MQ as the messaging provider instead of by using WebSphere Application Server default messaging as the messaging provider. See “Creating and configuring a cluster that uses WebSphere MQ as the messaging provider” on page 46.

Customizing the event runtime

About this task

You must consider various separate components when customizing your event runtime.

Procedure

- **Database**: The event runtime uses a database to store runtime and configuration data. You must have a database configured for the event runtime to operate. Five database providers are supported:
  - Embedded Apache Derby. The event runtime creates a database by using the instance of Apache Derby that is embedded in WebSphere Application Server.
This option does not require any additional configuration, but is not supported for use in a production environment.

- IBM Apache Derby Network Server.
- IBM DB2 Universal Database™.
- Microsoft SQL Server.
- Oracle.

If you do not select embedded Apache Derby, you must create the database before creating your Decision Server Events profile. For more information, see “Creating the event runtime database” on page 109.

**Messaging:** Decision Server Events uses JMS messaging to receive events and send actions from the event runtime. Two messaging providers are supported:

- WebSphere Application Server default messaging.
- WebSphere MQ

If you want to ensure that no messages are lost if the server fails, you must configure a durable event destination. For more information, see “Configuring the durable event queue” on page 123. You must also configure your messaging provider to ensure that JMS messages persist and can be recovered in the event of server failure.

If your messaging provider is clustered, you must ensure that any persistent store is configured so that if a cluster member fails, another cluster member retrieves any persisted messages. Configuring a Decision Server Events cluster can provide high availability and scalability of JMS messaging. Clustering of the messaging separately from the event runtime, in a gold topology cluster, is not typically necessary unless the messaging is performance bottleneck for the system.

**Technology Connectors:** Event connectors provide connections to and from external systems by various protocols: Email (POP3, SMTP), File System, FTP, HTTP, JDBC, JMS, REST, and SOAP over HTTP. You configure the technology connectors in the Event Designer user interface. For more information, see Technology connectors.

The File System, HTTP, JDBC, JMS, REST, and SOAP connectors run as applications within WebSphere Application Server and can be deployed to a cluster for high availability and scalability.

Other technology connector protocols run as a stand-alone process and can be installed on a different computer to the event runtime. For more information, see “Configuring technology connectors on a separate computer” on page 126. These connectors cannot be configured to be highly available or scalable.

**Steps Table:** Step data is the data used to track all context information, which is used in the evaluation of event rules. For more information, see Contexts. By default, the event runtime stores step data in a table in the database you selected when configuring your Decision Server Events profile. Storing step data in the database provides persistence of context data in the case of server failure. Or, you can:

- Configure the event runtime to store step data in memory for higher performance, but no persistence. For more information, see “Configuring the event runtime to store step data in memory” on page 121.
- Configure the event runtime to store step data in ObjectGrid, which provides in-memory storage and can (if used in a clustered runtime environment) also provide failover capabilities for context data. For more information, see “Configuring the event runtime to store context data in ObjectGrid” on page 122.
Configuring a stand-alone Decision Server Events environment

For simplicity and ease of deployment, you can configure a single server Decision Server Events environment. This topology is the simplest topology to configure and manage, but it is not highly available or scalable.

About this task

To configure a stand-alone Decision Server Events environment, you can:

- Either, create a new Operational Decision Manager Decision Server Events application server profile. See "Creating a new Decision Server Events application server profile."

- Or, augment an existing WebSphere Application Server application server profile with Operational Decision Manager Decision Server Events. See "Augmenting an existing WebSphere Application Server application server profile with Decision Server Events" on page 102.

Creating a new Decision Server Events application server profile

To create a new stand-alone Decision Server Events environment, you must first create a Operational Decision Manager Decision Server Events profile. You can create a profile by using the Profile Management Tool or by using the manageprofiles command.

Before you begin

During profile creation, you are asked to select the database manager which hosts the tables for the event runtime. Unless you are using Apache Derby, create this database before starting profile creation. See "Creating the event runtime database" on page 109.

About this task

You can create a new profile as follows:

- On a 32-bit operating system, you can use the Profile Management Tool graphical user interface. The Profile Management Tool is not supported on an HP-UX operating system, nor is it on z/OS.

- On a 32-bit operating system or any other operating system, you can use the manageprofiles command.

Creating a new profile by using the Profile Management Tool

About this task

To invoke the Profile Management Tool directly:

- On Windows only, click Start > All Programs > IBM WebSphere > IBM WebSphere Application Server V8.5 > Tools > Profile Management Tool.

- On multiplatforms, switch to the <WAS_HOME>/bin/ProfileManagement directory and run the pmt script, where <WAS_HOME> is the installation location of WebSphere Application Server. On Windows run pmt.bat, and on Linux and UNIX run pmt.sh.

To create a new application server profile augmented with Operational Decision Manager Decision Server Events:
**Procedure**

1. On the Profiles page, click **Create**.
2. On the Environment Selection page, under **IBM Operational Decision Manager**, select **Application server profile augmented with IBM Operational Decision Manager Decision Server Events**.
3. On the Profile Creation Options page, you can select one of the following options:
   - **Typical profile creation**: Create a profile with default WebSphere Application Server configuration settings.
   - **Advanced profile creation**: Specify your own values or take default settings. For additional WebSphere Application Server configuration settings, see Creating application server profiles information in the WebSphere Application Server information center.
4. On the Installation Location page, enter or browse for the fully-qualified location where Decision Server Events is installed or accept the default location if correct.
5. On the Administrative Security page, select whether to enable security on WebSphere Application Server. If you do enable security, both administrative and application security for WebSphere Application Server is turned on. Supply a new user ID and password of your choice, and enter the password a second time to confirm it.
6. On the Database Configuration page,
   a. Select a database manager from the list and specify the appropriate connection details. This database manager hosts the tables for the event runtime.
      If you select embedded Apache Derby, the repository database is created for you. In all other cases, you must have already created the database for the repository.
   b. If you did not select the embedded Apache Derby database, you must provide connection details including the fully-qualified location and name of the JDBC driver JAR file for your chosen database manager. Enter the following details for the database server:
      - **Database name**: The name of the database that hosts the Decision Server Events event runtime tables.
      - **Database server host name or IP address**: The name of the computer where the database server is located. The default is localhost.
      - **Database TCP/IP service port or listener port**: The connection port number for the database manager.
      - **Fully-qualified location and name of the JDBC driver file**: Enter the file name and location.
      - **User name**: The user name that is used by Decision Server Events to connect to the database server. This user name must have administrative privileges.
      - **Password**: The password associated with the user name. Confirm the password.
   c. If you did not select the embedded Apache Derby database, click **Test Connection** to validate the connection to the database. Ensure that the connection is successful before proceeding with profile creation.
7. On the Messaging Provider Configuration page, select one of:
• **WebSphere Application Server default messaging.** Decision Server Events is configured to use the default messaging provider that is embedded in WebSphere Application Server as the JMS provider.

• **WebSphere MQ JMS messaging.** Supply the WebSphere MQ related information and Decision Server Events is configured to use WebSphere MQ as the JMS provider.

• **Do not configure a messaging provider.** No messaging provider is configured during profile creation. You must configure a single JMS provider before starting Decision Server Events.

After profile creation has completed, you can change JMS providers, or configure the JMS provider of your choice, but you must configure a single JMS provider before starting Decision Server Events.

To use, or to switch to, WebSphere Application Server default messaging as the JMS provider, see “Configuring WebSphere Application Server default messaging to be the JMS provider” on page 113.

To use, or to switch to, WebSphere MQ as the JMS provider, see “Configuring WebSphere MQ to be the JMS provider” on page 115.

8. On the Profile Creation Summary page, review the information and click **Create** to create the new profile.

**Results**

A new application server profile augmented with Operational Decision Manager Decision Server Events is created.

**Creating a new profile by using the manageprofiles command**

**Procedure**

1. Open a command prompt (Windows) or a terminal emulator (Linux and UNIX) and navigate to the `was_install_dir/bin` directory.

2. Run the `manageprofiles` command by using `manageprofiles.bat` for Windows or `manageprofiles.sh` for Linux and UNIX. Provide the following parameters:

   - `-create`
     To create the new profile.

   - `-templatePath <template_path>`
     where `<template_path>` is the location of the profile template. Provide a value for the application server profile of `was_install_dir/profileTemplates/wbe/default`.

   Optionally, you can specify the following parameters:

   - `-wbeHome installation location`
     where `installation location` is the fully-qualified path that is the installation location for Decision Server Events. You must specify this parameter if you have installed Decision Server Events in a nondefault location.

   - `-profileName profile_name`
     where `profile_name` is the name of the profile.

   - `-cellName cell_name`
     where `cell_name` is the name of the cell that is created.

   - `-nodeName node_name`
     where `node_name` is the name of the node that is created.

   - `-hostName host_name`
     where `host_name` is the host name of the computer hosting the profile.
**-serverName server_name**
where `server_name` is the server name that is created.

**-enableAdminSecurity true|false**
where `true` turns on WebSphere Application Server administrative security. If you provide this parameter, you must also provide `-adminUserName` and `-adminPassword`. The default is `false` if you do not specify this parameter.

**-adminUserName username**
where `username` is the user ID that is used to access WebSphere Application Server. Only required if `enableAdminSecurity` is enabled.

**-adminPassword password**
where `password` is the password for the user ID that is used to access WebSphere Application Server. Only required if `enableAdminSecurity` is enabled.

**-wbeDbType database_type**
where `database_type` specifies the database product. Valid values are:
- `Derby_Embedded`: the default for the application server profile
- `DB2_Universal`
- `Derby_NetworkServer`
- `Oracle`
- `MS_SQL_Server`

**-wbeDbName database_name**
where `database_name` is the name of the database. Required unless `-wbeDbType` is `Derby_Embedded`.

**-wbeDbUserId database_user_name**
where `database_user_name` is the user ID that is used to access the database server. Required unless `-wbeDbType` is `Derby_Embedded`.

**-wbeDbPassword password**
where `password` is the password for the user ID that is used to access the database server. Required unless `-wbeDbType` is `Derby_Embedded`.

**-wbeDbJDBCClasspath database_jdbc_classpath**
where `database_jdbc_classpath` specifies the path to the JDBC class path files. Required unless `-wbeDbType` is `Derby_Embedded`.

**-wbeDbHostName host_name**
where `host_name` is the host name for the database server. Required unless `-wbeDbType` is `Derby_Embedded`.

**-wbeDbServerPort port_number**
where `port_number` is the port where the TCP/IP service is assigned or the port on which the database is listening. Required unless `-wbeDbType` is `Derby_Embedded`.

**-wbeMsgingType messaging_type**
where `messaging_type` specifies which JMS provider is to be configured. Valid values are:
- `Default_Messaging`: Decision Server Events is configured to use WebSphere Application Server default messaging. This value is the default value.
- `MQ_JMS_Messaging`: Decision Server Events is configured to use WebSphere MQ
- No Messaging: No JMS provider is configured. Before executing Decision Server Events, but you must configure a JMS provider manually. See "Configuring WebSphere Application Server default messaging to be the JMS provider" on page 113 or "Configuring WebSphere MQ to be the JMS provider" on page 115.

- **wbeMqMsgingQmgrName** queue_manager_name
  where queue_manager_name is the WebSphere MQ queue manager name. Required only if **-wbeMsgingType** is MQ_JMS_Messaging.

- **wbeMqMsgingQmgrHostName** host_name
  where host_name is the host name of the WebSphere MQ queue manager. Required only if **-wbeMsgingType** is MQ_JMS_Messaging.

- **wbeMqMsgingQmgrPort** port_number
  where port_number is the port on which the WebSphere MQ queue manager is listening. Required only if **-wbeMsgingType** is MQ_JMS_Messaging.

- **wbeMqMsgingTransType** transport_type
  where transport_type is the WebSphere MQ client transport type, either BINDINGS or CLIENT. Required only if **-wbeMsgingType** is MQ_JMS_Messaging.

- **-enableTester** true|false
  where true installs and configures the Event Tester runtime. The default is false if you do not specify this parameter. This parameter is only applicable for the application server profile and cannot be used in a production environment.

- **-disableBusSecurity**
  This parameter disables bus security for the service integration bus, WbeBus, when the bus is created.

**Results**

The profile is now created.

**Examples**

The first example creates a new application server profile on Windows using embedded Apache Derby as the database provider and WebSphere Application Server default messaging as the messaging provider.

manageprofiles.bat -create -templatePath "C:\Program Files\IBM\ODM851\WAS\profileTemplates\wbe\default"

The second example creates a new application server profile on Linux and UNIX using embedded Apache Derby as the database provider and WebSphere Application Server default messaging as the messaging provider.

manageprofiles.sh -create -templatePath "/opt/ibm/ODM851/ODM/WAS/profileTemplates/wbe/default"

**Augmenting an existing WebSphere Application Server application server profile with Decision Server Events**

To create a new stand-alone Decision Server Events environment from an existing WebSphere Application Server stand-alone environment, you can augment a WebSphere Application Server application server profile with Operational Decision
Manager Decision Server Events. You can augment a profile by using the Profile Management Tool or by using the manageprofiles command.

**Before you begin**

Before you augment an WebSphere Application Server application server profile, ensure that all application servers in the profile are stopped.

During the profile creation task, you are prompted to select the database manager that hosts the tables for the event runtime. Unless you are using Apache Derby, you must create this database before starting to create the profile. For more information, see “Creating the event runtime database” on page 109.

After you have augmented the profile, review the profile security configuration to check the administrative security and application security settings. If administrative security is enabled, application security must also be enabled.

**About this task**

You can augment an existing profile as follows:

- On a 32-bit operating system, you can use the Profile Management Tool graphical user interface. The Profile Management Tool is not supported on an HP-UX operating system, nor is it on z/OS.
- On a 32-bit operating system or any other operating system, you can use the manageprofiles command.

**Augmenting an existing application server profile by using the Profile Management Tool**

**About this task**

To invoke the Profile Management Tool directly:

- On Windows only, click Start > All Programs > IBM WebSphere > IBM WebSphere Application Server V8.5 > Tools > Profile Management Tool.
- On multiplatforms, switch to the <WAS_HOME>/bin/ProfileManagement directory and run the pmt script, where <WAS_HOME> is the installation location of WebSphere Application Server. On Windows run pmt.bat, and on Linux and UNIX run pmt.sh.

To augment an existing application server profile with Operational Decision Manager Decision Server Events:

**Procedure**

1. On the Profiles page, select an existing application server profile and click Augment.
2. On the Augment Selection page, select Application server profile augmented with IBM Operational Decision Manager Decision Server Events.
3. On the Installation Location page, enter or browse for the fully-qualified location where Decision Server Events is installed or accept the default location if correct.
4. On the Database Configuration page,
   a. Select a database manager from the list and specify the appropriate connection details. This database manager hosts the tables for the event runtime.
If you select embedded Apache Derby, the repository database is created for you. In all other cases, you must have already created the database for the repository.

b. If you did not select embedded Apache Derby, you must provide connection details including the fully-qualified location and name of the JDBC driver JAR file for your chosen database manager. Enter the following details for the database server:

- **Database name**: The name of the database that hosts the Decision Server Events event runtime tables.
- **Database server host name or IP address**: The name of the computer where the database server is located. The default is localhost.
- **Database TCP/IP service port or listener port**: The connection port number for the database manager.
- **Fully-qualified location and name of the JDBC driver file**: Enter the file name and location.
- **User name**: The user name that is used by Decision Server Events to connect to the database server. This user name must have administrative privileges.
- **Password**: The password associated with the user name. Confirm the password.

c. If you did not select the embedded Apache Derby database, click **Test Connection** to validate the connection to the database. Ensure that the connection is successful before proceeding with profile creation.

5. On the Messaging Provider Configuration page, select one of:

- **WebSphere Application Server default messaging**: Decision Server Events is configured to use the default messaging provider that is embedded in WebSphere Application Server as the JMS provider.
- **WebSphere MQ JMS messaging**: Supply the WebSphere MQ related information and Decision Server Events is configured to use WebSphere MQ as the JMS provider.
- **Do not configure a messaging provider**: No messaging provider is configured during profile creation. You must configure a single JMS provider before starting Decision Server Events.

After profile augmentation has completed, you can change JMS providers, or configure the JMS provider of your choice, but you must configure a single JMS provider before starting Decision Server Events.

To use, or to switch to, WebSphere Application Server default messaging as the JMS provider, see [“Configuring WebSphere Application Server default messaging to be the JMS provider” on page 113](#)

To use, or to switch to, WebSphere MQ as the JMS provider, see [“Configuring WebSphere MQ to be the JMS provider” on page 115](#)

6. If WebSphere Application Server administrative security is turned on in the profile, the Security Configuration page is displayed. Enter the user ID and password for your application server.

7. On the Profile Augmentation Summary page, review the information and click **Augment** to augment the profile.

**Results**

The existing application server profile is augmented with Operational Decision Manager Decision Server Events.
What to do next

If administrative security is enabled for the profile, you must also enable application security for the profile.

Augmenting an existing application server profile by using the manageprofiles command

Procedure

1. Open a command prompt (Windows) or a terminal emulator (Linux and UNIX) and navigate to the was_install_dir/bin directory.
2. Run the manageprofiles command by using manageprofiles.bat for Windows or manageprofiles.sh for Linux and UNIX. Provide the following parameters:

   - **-augment**
     To augment the existing profile.

   - **-templatePath** `<template_path>`
     where `<template_path>` is the location of the profile template. Provide the following value for the application server profile: was_install_dir/profileTemplates/wbe/default.

   - **-profileName** `profile_name`
     where `profile_name` is the name of the existing application server profile that you are augmenting.

   Optionally, you can specify the following parameters:

   - **-wbeHome** `installation location`
     where `installation location` is the fully-qualified path that is the installation location for Decision Server Events. You must specify this parameter if you have installed Decision Server Events in a nondefault location.

   - **-hostName** `host_name`
     where `host_name` is the host name of the computer hosting the profile.

   - **-adminUserName** `username`
     where `username` is the user ID that is used to access WebSphere Application Server. Only required if WebSphere Application Server administrative security is turned on.

   - **-adminPassword** `password`
     where `password` is the password for the user ID that is used to access WebSphere Application Server. Only required if WebSphere Application Server administrative security is turned on.

   - **-wbeDbType** `database_type`
     where `database_type` specifies the database product. Valid values are:
     - *Derby_Embedded*: the default for the application server profile
     - *DB2_Universal*
     - *Derby_NetworkServer*
     - *Oracle*
     - *MS_SQL_Server*

   - **-wbeDbName** `database_name`
     where `database_name` is the name of the database. Required unless `-wbeDbType` is *Derby_Embedded*.

   - **-wbeDbUserId** `database_user_name`
     where `database_user_name` is the user ID that is used to access the database server. Required unless `-wbeDbType` is *Derby_Embedded*.
-wbeDbPassword  password
  where  password  is  the  password  for  the  user  ID  that  is  used  to  access  the
database  server.  Required  unless  -wbeDbType  is  Derby_Embedded.

-wbeDbJDBCClasspath  database_jdbc_classpath
  where  database_jdbc_classpath  specifies  the  path  to  the  JDBC  class  path  files.
Required  unless  -wbeDbType  is  Derby_Embedded.

-wbeDbHostName  host_name
  where  host_name  is  the  host  name  for  the  database  server.  Required  unless
-wbeDbType  is  Derby_Embedded.

-wbeDbServerPort  port_number
  where  port_number  is  the  port  where  the  TCP/IP  service  is  assigned  or  the
port  on  which  the  database  is  listening.  Required  unless  -wbeDbType  is
Derby_Embedded.

-wbeMqMsgingType  messaging_type
  where  messaging_type  specifies  which  JMS  provider  is  to  be  configured.
Valid  values  are:
  • Default_Messaging:  Decision  Server  Events  is  configured  to  use
    WebSphere  Application  Server  default  messaging.  This  value  is  the
default  value.
  • MQ_JMS_Messaging:  Decision  Server  Events  is  configured  to
    useWebSphere  MQ
  • No_Messaging:  No  JMS  provider  is  configured.  Before  executing  Decision
Server  Events,  but  you  must  configure  a  JMS  provider  manually.  See
“Configuring  WebSphere  Application  Server  default  messaging  to  be  the
JMS  provider”  on  page  113  or  “Configuring  WebSphere  MQ  to  be  the
JMS  provider”  on  page  115.

-wbeMqMsgingQmgrName  queue_manager_name
  where  queue_manager_name  is  the  WebSphere  MQ  queue  manager  name.
Required  only  if  -wbeMqMsgingType  is  MQ_JMS_Messaging.

-wbeMqMsgingQmgrHostName  host_name
  where  host_name  is  the  host  name  of  the  WebSphere  MQ  queue  manager.
Required  only  if  -wbeMqMsgingType  is  MQ_JMS_Messaging.

-wbeMqMsgingQmgrPort  port_number
  where  port_number  is  the  port  on  which  the  WebSphere  MQ  queue  manager
is  listening.  Required  only  if  -wbeMqMsgingType  is  MQ_JMS_Messaging.

-wbeMqMsgingTransType  transport_type
  where  transport_type  is  the  WebSphere  MQ  client  transport  type,  either
BINDINGS  or  CLIENT.  Required  only  if  -wbeMqMsgingType  is
MQ_JMS_Messaging.

-enableTester  true|false
  where  true  installs  and  configures  the  Event  Tester  runtime.  The  default  is
false  if  you  do  not  specify  this  parameter.  This  parameter  is  only  applicable
for  the  application  server  profile  and  cannot  be  used  in  a  production
environment.

-disableBusSecurity
  This  parameter  disables  bus  security  for  the  service  integration  bus,
WbeBus,  when  the  bus  is  created.
Results

The profile is now augmented.

Examples

The first example augments an existing application server profile called AppSrv01 on Windows using embedded Apache Derby as the database provider and WebSphere Application Server default messaging as the messaging provider.

```
manageprofiles.bat -augment
-templatePath "C:\Program Files\IBM\ODM851\WAS\profileTemplates\wbe\default"
-profileName AppSrv01
```

The second example augments an existing application server profile called AppSrv01 on Linux and UNIX using embedded Apache Derby as the database provider and WebSphere Application Server default messaging as the messaging provider.

```
manageprofiles.sh -augment
-templatePath "/opt/ibm/ODM851/ODM/WAS/profileTemplates/wbe/default"
-profileName AppSrv01
```

What to do next

If administrative security is enabled for the profile, you must also enable application security for the profile.

Unaugmenting an existing application server profile by using the manageprofiles command

About this task

To undo the augmentation of an existing profile by using the manageprofiles command:

Procedure

1. Open a command prompt (Windows) or a terminal emulator (Linux and UNIX) and navigate to the `was_install_dir/bin` directory.
2. Run the `manageprofiles` command by using `manageprofiles.bat` for Windows or `manageprofiles.sh` for Linux and UNIX. Provide the following parameters:

   - `unaugment`
     To unaugment the existing profile.

   - `templatePath <template_path>`
     where `<template_path>` is the location of the profile template. Provide the following value for the application server profile: `was_install_dir/profileTemplates/wbe/default`.

   - `profileName profile_name`
     where `profile_name` is the name of the existing application server profile that you are unaugmenting.

Results

The profile is now unaugmented. All Decision Server Events features are removed from the profile.
Examples

The first example unaugments an existing Decision Server Events application server profile called WODMSrv01 on Windows.

manageprofiles.bat -unaugment -templatePath "C:\Program Files\IBM\ODM851\WAS\profileTemplates\wbe\default" -profileName WODMSrv01

The second example unaugments an existing Decision Server Events application server profile called WODMSrv01 on Linux and UNIX.

manageprofiles.sh -unaugment -templatePath "/opt/ibm/ODM851/ODM/WAS/profileTemplates/wbe/default" -profileName WODMSrv01

Editing the setenv files

You must edit the setenv files as part of the configuration steps to complete your installation.

About this task

The setenv script files are located in subdirectories of the component_dir/config directory (setenv.sh on Linux and UNIX or setenv.bat on Windows).

Note: When you install a product fix pack after editing the setenv files, the setenv files are overwritten. If you want to keep your changes, create backups of the files before installing the fix pack.

Procedure

1. Change to the component_dir/config/was directory, and edit the setenv file, to add the appropriate values for the WebSphere Application Server instance on which Operational Decision Manager is going to run. Ask your WebSphere Application Server administrator for these values. For more information, see Environment variables.

2. Change to the component_dir/config/db directory, and edit the setenv file to add the appropriate values for the DB2 instance that is going to host the event runtime. Ask your DB2 database administrator for these values. For more information, see Environment variables.

3. In the component_dir/config/wbe directory, edit the setenv file. Update the WBE_INSTALL property so that it points to the Operational Decision Manager product installation in the read-only file system. For example:

   WBE_INSTALL=<InstallDir>

Customizing the event runtime environment

Before starting the event runtime for the first time, you might want to make configuration changes to some of the components in the runtime environment.

Configuring the event runtime database for Decision Server Events

The event runtime requires the use of a database. There are various configuration tasks that you might need to do to the database, depending on your environment.
Before you begin

The configuration tasks in this section refer to configuring the event runtime database for distributed platforms. If you are configuring the event runtime database for z/OS, instead see [Creating the event runtime database for z/OS](#).

**Creating the event runtime database**

Decision Server Events requires a database for the event runtime. This event runtime is the shared, secured location that contains assets such as data connections, business objects, events, and actions.

### Before you begin

The configuration tasks in this topic refer to configuring the event runtime database for distributed platforms. If you are configuring the event runtime database for z/OS, instead see [Creating the event runtime database for z/OS](#).

If you want to use the Apache Derby database that is embedded in WebSphere Application Server as the event runtime database, the database (called event runtime) and tables are created and configured for you during the installation process. If you want to use another supported database manager (including another installation of Apache Derby), you must create the event runtime database before you start the installer.

Apache Derby is not supported in a production environment.

If you want to use a remote DB2 database as the database for the event runtime, you must install the supplied DB2 Client Support.

1. Start the DB2 installer wizard.
2. Select Custom Installation.
3. Select the Client support feature and clear all the other features.
4. Complete the installation.

### About this task

The full list of supported database managers is available on the web at [System requirements](#).

### Procedure

When you create the event runtime database, record the following information:

- Event runtime database name of your choice
- Host name of the server where the database is located
- Port used to access the database
- Database user ID to be used to access the database
- Password associated with the database user ID

This information is used when configuring the event runtime database. See [Event runtime database settings](#).

### Note:

- To install, you must have full administrative privileges for the event runtime database. For example, for DB2, you must be authorized to CONNECT, CREATETAB, and so on. For Oracle you must have authority to ALTER
DATABASE, CREATE ANY TABLE, and so on. These full administrative privileges are only required to install Decision Server Events, not to use it. If you do not have full administrative privileges, you can still install if your Database Administrator (DBA) creates the event runtime tables in the database by using the manual instructions before you run the installer. After installation, you only need read, write, and delete access as a user.

- For all database managers except embedded Apache Derby: the database manager must be running before the installation starts and you must have available the JDBC .jar file for the database manager.

**Manually creating event runtime tables**
You can opt to tailor existing scripts and create the event runtime tables yourself, as a manual task. If you do not create these tables, they are automatically created for you during the startup of Decision Server Events.

**Before you begin**
The configuration tasks in this topic refer to configuring the event runtime database for distributed platforms. If you are configuring the event runtime database for z/OS, instead see "Creating the event runtime database for z/OS".

Before you create the event runtime tables, you must ensure that you have created the event runtime database. See "Creating the event runtime database" on page 109.

**About this task**
Decision Server Events supplies a copy of the scripts used to create the event runtime tables, in the `<InstallDir>/config/db` directory. The scripts are database dependant, one script for each supported database:

- For DB2, use `db2.sql`
- For Apache Derby, use `derbydb.sql`
- For Oracle, use `ora.sql`
- For Microsoft SQL Server, use `mssql2k.sql`

**Procedure**
Take a copy of the appropriate script, tailor it to meet your requirements, and use it to create the tables.

**Note:** You must take a copy of the script and execute it manually. Even if you edit and save the script that is in the `<InstallDir>/config/db` directory, it is not used to create the event runtime tables.

**Deploying tables in separate databases**
By default, the event runtime tables are created in a single database, but you can deploy sets of tables in separate databases.

**About this task**
The event runtime database stores assets (such as data connections, business objects, events, actions, and event rules) that are used by the event runtime to receive events and initiate the actions based on event rule groups.
The event runtime consists of several sets of tables and a number of views, which can be used to report on history. They are built by running a database manager-specific script. At installation time, you specify a single database and by default, all tables, and views are located in that database. The tables are created in this database at installation time.

To deploy sets of tables in separate databases and reference those databases:

**Procedure**

1. Create a database to hold the tables.
2. Use the appropriate CREATE statements with the SQL tool for your database manager and create the tables.

The table shows the SQL statements to create separate databases. Only partial statements are shown. For the complete syntax, see the documentation supplied with your chosen database manager.

**Note:** If you are using history, you must always ensure that the history and asset definition tables, and the resources, are created and used in the same database.

*Table 12. SQL statements to create each table.*

<table>
<thead>
<tr>
<th>Table name</th>
<th>SQL statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset definitions (maestro_asset, properties)</td>
<td>CREATE TABLE maestro_asset</td>
</tr>
<tr>
<td></td>
<td>ALTER TABLE maestro_asset</td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE properties</td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE labels</td>
</tr>
<tr>
<td>Time delays</td>
<td>CREATE TABLE time_based_asset</td>
</tr>
<tr>
<td></td>
<td>CREATE INDEX FIRETIME_IDX</td>
</tr>
<tr>
<td>Event Flows</td>
<td>CREATE TABLE steps</td>
</tr>
<tr>
<td></td>
<td>CREATE CLUSTERED INDEX STREAM_IDX</td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE csio_entry</td>
</tr>
<tr>
<td></td>
<td>CREATE INDEX csio_entry_idx on csio_entry</td>
</tr>
<tr>
<td></td>
<td>CREATE INDEX csio_entry_times</td>
</tr>
<tr>
<td>User Console</td>
<td>CREATE TABLE actions</td>
</tr>
<tr>
<td></td>
<td>ALTER TABLE actions</td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE actions_prefs</td>
</tr>
<tr>
<td></td>
<td>ALTER TABLE actions_prefs</td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE auth_users</td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE auth_groups</td>
</tr>
</tbody>
</table>
Table 12. SQL statements to create each table. (continued)

<table>
<thead>
<tr>
<th>Table name</th>
<th>SQL statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>History</td>
<td>CREATE TABLE <code>history_event</code></td>
</tr>
<tr>
<td></td>
<td>CREATE INDEX <code>history_e_idx</code></td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE <code>history_action</code></td>
</tr>
<tr>
<td></td>
<td>CREATE INDEX <code>history_a_idx</code></td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE <code>history_ent_obj</code></td>
</tr>
<tr>
<td></td>
<td>CREATE INDEX <code>history_eo_idx</code></td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE <code>history_ent_obj_values</code></td>
</tr>
<tr>
<td></td>
<td>CREATE INDEX <code>history_eo_v_idx</code></td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE <code>history_filter</code></td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE <code>history_rule</code></td>
</tr>
<tr>
<td></td>
<td>CREATE INDEX <code>history_rule_idx</code></td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE <code>history_watch_time</code></td>
</tr>
<tr>
<td></td>
<td>CREATE VIEW <code>as_director_events</code></td>
</tr>
<tr>
<td></td>
<td>CREATE VIEW <code>as_director_actions</code></td>
</tr>
<tr>
<td></td>
<td>CREATE VIEW <code>as_director_ent_objs</code></td>
</tr>
<tr>
<td></td>
<td>CREATE VIEW <code>as_director_ent_objs_values</code></td>
</tr>
<tr>
<td></td>
<td>CREATE VIEW <code>as_director_filters</code></td>
</tr>
<tr>
<td></td>
<td>CREATE VIEW <code>as_director_rules</code></td>
</tr>
<tr>
<td></td>
<td>CREATE VIEW <code>as_director_watch_times</code></td>
</tr>
</tbody>
</table>

3. Create a new datasource by using WebSphere Application Server administrative console:
   a. Click **JDBC > Data sources**.
   b. Ensure that the scope is the same as the event runtime datasource and click **New**.
   c. Enter a name in the **Datasource Name** field, for example, **Event Runtime History Datasource**.
   d. Enter a name in the **JNDI Name** field, for example, **jdbc/wbe/history**.
   e. Click **Select an existing JDBC provider** and select **Event Runtime JDBC Provider**.
   f. Enter the database-specific properties that point to the database where the tables were created in step 2 on page 111.
   g. Select an authentication alias, or create a new one with credentials for the database, then click **Finish**.

4. Set the following properties to the value that you specified in **JNDI Name**, depending on which tables you have moved:
Table 13. Properties

<table>
<thead>
<tr>
<th>Table name</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset definitions</td>
<td>as.director.common.db.asset.instance</td>
</tr>
<tr>
<td>Time delays</td>
<td>as.director.common.db.timebased.instance</td>
</tr>
<tr>
<td>Event Flows</td>
<td>as.director.common.db.eventflow.instance</td>
</tr>
<tr>
<td>User Console</td>
<td>as.director.connectors.pem.instance</td>
</tr>
<tr>
<td>History</td>
<td>as.director.common.db.history.instance</td>
</tr>
</tbody>
</table>

For more information, see “Setting properties” on page 117.

Deploying the event runtime in a case-sensitive database manager

If you are using a case-sensitive database manager, event runtime table names must be changed to be all in uppercase.

About this task

If the event runtime database is created in a case-sensitive database, references to table names that are defined as properties must be changed to be all in uppercase.

Procedure

1. For details of how to set these properties, see “Setting properties” on page 117.
2. Change the value of the as.director.common.db.asset.table property from maestro_asset (all lowercase) to MAESTRO.Asset (all uppercase).
   If you are using Properties, the properties are in the Asset Event Runtime subsection of the Common section.
3. Change the value of the as.director.common.db.timebased.table property from time_based_asset (all lowercase) to TIME_BASED.Asset (all uppercase).
   If you are using Properties, the properties are in the Server section, in the Timebased Event Runtime Settings subsection.

Results

The changes are applied the next time the wberuntimeear application is restarted.

Changing the event runtime messaging provider

If you have configured your event runtime to use one messaging provider but you want to change it to another messaging provider, or if you chose not to configure a messaging provider when creating or augmenting your Decision Server Events profile, you must do some additional configuration.

Configuring WebSphere Application Server default messaging to be the JMS provider

If you want to use WebSphere Application Server default messaging as the JMS provider for the event runtime, and you either did not select the JMS provider or you selected WebSphere MQ when you created or augmented the Decision Server Events profile, you must do some additional configuration of WebSphere Application Server and Decision Server Events.
Before you begin

The event runtime can be configured to work with only one JMS provider: either the WebSphere Application Server default messaging or WebSphere MQ, but not both.

The .sh and .bat files referenced in this topic rely on settings in the setenv file. Check and confirm your settings before doing the following steps. For more information, see “Editing the setenv files” on page 108.

Procedure

1. If, when you created or augmented the Decision Server Events profile, you chose to use WebSphere MQ but now you want to use WebSphere Application Server default messaging as the JMS provider instead, you must remove the configuration that was done by the profile:

   Table 14. Command to remove the configuration

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Location in which you must run the command</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>&lt;InstallDir&gt;\config\was\</td>
<td>configure_MQ_JMS_Messaging.bat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-undo</td>
</tr>
<tr>
<td>Linux and UNIX</td>
<td>&lt;InstallDir&gt;/config/was/</td>
<td>configure_MQ_JMS_Messaging.sh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-undo</td>
</tr>
</tbody>
</table>

   If, when you created or augmented the Decision Server Events profile, you did not select the JMS provider, you do not need to do this step.

2. To complete configuration of WebSphere Application Server default messaging as the JMS provider, run the following commands:

   Table 15. Command to complete the configuration

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Location in which you must run the command</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>&lt;InstallDir&gt;\config\was\</td>
<td>configure_bus.bat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>configure_messaging.bat</td>
</tr>
<tr>
<td>Linux and UNIX</td>
<td>&lt;InstallDir&gt;/config/was/</td>
<td>configure_bus.sh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>configure_messaging.sh</td>
</tr>
</tbody>
</table>

   The configure_bus.sh script configures the service integration bus, WbeBus. If application server administrative security is enabled, the bus is secured. To disable bus security, supply the optional argument, -disableBusSecurity.

   If the service integration bus is secured, you must specify two arguments for the configure_messaging.bat (or configure_messaging.sh) script, to supply the user ID and password that the activation specifications use to connect to the secured bus. For example:

   configure_messaging.bat <user_id> <password>

   or

   configure_messaging.sh <user_id> <password>

   If security is not enabled for the service integration bus, do not specify a user ID or password. For more information, see “Securing the service integration bus” on page 142.
3. Restart WebSphere Application Server.

**Configuring WebSphere MQ to be the JMS provider**

If you want to use WebSphere MQ as the JMS provider for the event runtime, and you either did not select the JMS provider or you selected WebSphere Application Server default messaging when you created or augmented the Decision Server Events profile, you must do some additional configuration of WebSphere MQ, WebSphere Application Server, and Decision Server Events.

**Before you begin**

The event runtime can be configured to work with only one JMS provider: either the WebSphere Application Server default messaging or WebSphere MQ, but not both.

You must have the WebSphere MQ client installed on the same computer as Decision Server Events.

**Configuring WebSphere MQ:**

**About this task**

The WebSphere MQ administrator must do the following tasks:

**Procedure**

1. Create the queue manager, by using the `crtmqm` command.
2. Start the queue manager, by using the `strmqm` command.
3. If you are using WebSphere MQ V7.1, publish/subscribe must be enabled. Use the `ALTER QMGR` command for this queue manager.
4. Create the queues required, by running:
   
   ```bash
   runmqsc queue-manager-name < mq-install-dir\java\bin\MQJMS_PSQ.mqsc
   ```
   
   where `mq-install-dir` is the installation location of WebSphere MQ.
5. If you are configuring a cluster, create the extra queues required, by running:
   
   ```bash
   runmqsc queue-manager-name <<InstallDir>\config\was\create_MQ_JMS_MQ_queues.mqsc
   ```
   
   where `<InstallDir>` is the installation location of Decision Server Events.

**Related information:**

- [WebSphere MQ V7.1 information center](#)

**Configuring Decision Server Events:**

**Before you begin**

Before you configure Decision Server Events to use WebSphere MQ as the JMS provider, you must know the appropriate queue manager name, the WebSphere MQ host name and port, and the client transport type. Obtain these values from the WebSphere MQ administrator.

The `configure_messaging` and `configure_MQ_JMS_messaging` files referenced in this topic rely on settings in the `setenv` file. Check and confirm your settings before doing the following steps. For more information, see “Editing the setenv files” on page 108.
Procedure

1. If, when you created or augmented the Decision Server Events profile, you chose to use WebSphere Application Server default messaging but now you want to use WebSphere MQ as the JMS provider instead, run the following command to remove the messaging configuration that was done by the profile:

   Table 16. Command to run.

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Location in which you must run the command</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>&lt;InstallDir&gt;\config\was\</td>
<td>configure_messaging.bat -undo</td>
</tr>
<tr>
<td>Linux, UNIX and z/OS</td>
<td>&lt;InstallDir&gt;/config/was/</td>
<td>configure_messaging.sh -undo</td>
</tr>
</tbody>
</table>

If, when you created or augmented the Decision Server Events profile, you did not select the JMS provider, you do not need to do this step.

2. Configure Decision Server Events and WebSphere Application Server:
   a. Navigate to <InstallDir>/config/was directory.
   b. Run the following script:

      ```
      configure_MQ_JMS_messaging -q queue-manager-name
      [-p queue-manager-port] [-h queue-manager-host] [-s scope]
      [-t transport] [-c channel-name]
      ```

      where
      - `queue-manager-name` is the name of the queue manager that you created when configuring your installation of WebSphere MQ
      - `queue-manager-port` is the port on which the queue manager is listening (the default value is 1414)
      - `queue-manager-host` is the host name of the computer on which the queue manager is hosted (the default value is localhost)
      - `scope` is a valid WebSphere Application Server scope
      - `transport` is BINDINGS or CLIENT
      - `channel-name` is the name of the server connection channel

      For example: `configure_MQ_JMS_messaging -q QM_PSG -p 1415 -h localhost`

      The script creates the following objects in WebSphere Application Server:

   Table 17. Connection factories.

<table>
<thead>
<tr>
<th>Connection factories</th>
<th>JNDI name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue connection factory</td>
<td>jms/WbeQueueConnectionFactory</td>
</tr>
<tr>
<td>Topic connection factory</td>
<td>jms/WbeTopicConnectionFactory</td>
</tr>
<tr>
<td>Connection factory</td>
<td>jms/WbeConnectionFactory</td>
</tr>
</tbody>
</table>

   Table 18. Topics.

<table>
<thead>
<tr>
<th>Topics</th>
<th>JNDI name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action topic</td>
<td>jms/actionTopic</td>
</tr>
<tr>
<td>Command topic</td>
<td>jms/commandTopic</td>
</tr>
<tr>
<td>CbeListener</td>
<td>topic jms/WBE/CbeListener</td>
</tr>
<tr>
<td>Durable action topic</td>
<td>jms/durableActionTopic</td>
</tr>
<tr>
<td>Durable event topic</td>
<td>jms/durableEventDestination</td>
</tr>
</tbody>
</table>
Table 18. Topics. (continued)

<table>
<thead>
<tr>
<th>Topics</th>
<th>JNDI name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event topic</td>
<td>jms/eventDestination</td>
</tr>
<tr>
<td>History topic</td>
<td>jms/historyModuleDestination</td>
</tr>
</tbody>
</table>

Table 19. Activation specifications.

<table>
<thead>
<tr>
<th>Activation specification name</th>
<th>JNDI name</th>
</tr>
</thead>
<tbody>
<tr>
<td>wbe_events</td>
<td>jca/wbe_events</td>
</tr>
<tr>
<td>wbe_events_durable</td>
<td>jca/wbe_events_durable</td>
</tr>
<tr>
<td>wbe_history</td>
<td>jca/wbe_history</td>
</tr>
<tr>
<td>wbe_reset_watch</td>
<td>jca/wbe_reset_watch</td>
</tr>
<tr>
<td>wbe_ute_receive</td>
<td>jca/wbe_ute_receive</td>
</tr>
</tbody>
</table>

c. Ensure that the names of these objects are consistent with the corresponding properties.
   For details of how to set these properties, see “Setting properties.”
   If you are using Properties, these properties are in the JMS Server Settings and the Topics subsections of the JMS section.

d. Restart WebSphere Application Server to apply the changes.

Results

Decision Server Events is now configured to use WebSphere MQ as the JMS messaging provider.

What to do next

Make sure that the queue names created by create_MQ_JMS_MQ_queues.mqsc are consistent with the queue names that are specified by your WebSphere MQ JMS event destination, durable event destination, and history module destination queue definitions. These queue definitions can be found in the WebSphere Application Server administrative console under Resources > JMS > Queues.

Important: Although the names of these resources reference topics, these resources are actually queues for clusters.

By default, the WebSphere MQ queues are called:
- eventQueue for the event destination
- durableEventQueue for the durable event destination
- historyModuleQueue for the history module destination

Setting properties

Configuration properties for the event runtime are stored in WebSphere Application Server. You might want to change properties because they determine the configurable behaviour of Decision Server Events. There are three ways to set event runtime properties.
About this task

You can use different tasks depending on the platform that you are using.

Table 20. Method by platform.

<table>
<thead>
<tr>
<th>Method</th>
<th>Platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Configuring properties by using the WebSphere Application Server administrative console&quot;</td>
<td>Multiplatforms and z/OS</td>
</tr>
<tr>
<td>&quot;Configuring properties by using the wsadmin scripting tool&quot;</td>
<td>Multiplatforms and z/OS</td>
</tr>
<tr>
<td>&quot;Configuring properties by using the Properties tool&quot; on page 119</td>
<td>Multiplatforms only</td>
</tr>
</tbody>
</table>

Configuring properties by using the WebSphere Application Server administrative console

When you configure event runtime properties (for example, if you move the event runtime to a different database manager, or you want to change to a different JMS messaging provider), you can make the changes in WebSphere Application Server administrative console. When you edit the value of a property, ensure that the value is appropriate and spelled correctly because it is not validated.

Procedure

1. Start the WebSphere Application Server administrative console. See Starting the WebSphere Application Server administrative console.

2. Log in to the WebSphere Application Server administrative console. If authentication is enabled, you must use a valid user ID and password. If authentication is not enabled, type any user ID and log in.

3. In the navigation tree of the WebSphere Application Server administrative console, click Resources > Resource Environment > Resource environment entries.

4. On the Resource environment entries page, click WbeSrv01 to open the Configuration page. The general properties for that entry are displayed.

5. On the Configuration page, click Custom properties. The properties are listed.

6. Modify the properties as required.

   Note: If the property name is prefixed by "!" (for example !as.director.common.authentication.server), the property is disabled. Edit the name field to remove the "!", then set the remaining fields.

7. Save your changes.

8. Restart the event runtime to apply your changes.

Results

When the event runtime restarts, the changes you made to the properties are applied.

Configuring properties by using the wsadmin scripting tool

When you configure event runtime properties (for example, if you move the event runtime to a different database manager, or you want to change to a different JMS messaging provider), you can make the changes by using the WebSphere
Application Server administrative scripting tool, `wsadmin`. You can modify the properties by using `wsadmin`, or you can use the supplied AdminTask object to set a single property.

**About this task**

If you want to modify multiple properties, use the `wsadmin` tool. If you want to modify a single property, use the AdminTask object as follows:

**Procedure**

1. Start `wsadmin`. For more details of the use of `wsadmin`, see [Using the wsadmin scripting tool](#).

2. Run the following command to set a single property:

   - Using Jacl:
     ```
     wsadmin
     $AdminTask wbeSetProperty {-jndiName com/ibm/wbe/wbeProperties01
     -propertyName propertyName
     -propertyValue propertyValue}
     ```
   - Using Jython:
     ```
     wsadmin -lang jython
     AdminTask.wbeSetProperty('[{-jndiName com/ibm/wbe/wbeProperties01
     -propertyName propertyName
     -propertyValue propertyValue}]')
     ```

   Where `propertyName` is the name of the property you want to change, and `propertyValue` is the value that you want to set for that property.

3. Save your changes:

   - Using Jacl:
     ```
     $AdminConfig save
     ```
   - Using Jython:
     ```
     AdminConfig.save()
     ```

4. Restart the event runtime to apply your changes.

**Results**

When the event runtime restarts, the changes you made to the properties are applied.

**Configuring properties by using the Properties tool**

Settings for system properties are stored in an encrypted properties file. The file is managed through the use of the Properties tool or the WebSphere Application Server administrative console. When using the Properties tool, properties are displayed by folder, and most folders contain subsections that consist of groups of related properties.

**About this task**

During installation, a base properties file is installed and configured to reference the event runtime database and JMS provider. You can use the Properties tool to change property settings to accommodate your environment and preferences for system operation. There is no validation of the property that you have set. By default, a field holding a property values has a length of 64 characters. When you reach the limit, press ENTER and click to the end of the field to reset the focus. Properties then gives you up to five lines of 72 characters (by repeatedly pressing ENTER). Therefore, the maximum usable length is 360 characters.
Properties provides two entry modes:

- The Configurator wizard, which is the default entry mode, presents a series of screens containing only those properties you are most likely to want to edit.
- The Full Configurator presents all properties and is appropriate for more customized installations. You can also add custom properties to describe certain environment-specific conditions.

Newly-configured property settings do not become operational until the next time you start the wberuntimeear application. You can make additional changes to property settings as required. Apply those changes by stopping this application and restarting it.

**Procedure**

1. Start Properties by invoking the `properties` script (`propertiesui.bat` on Windows, `propertiesui.sh` on Linux and UNIX).
   The syntax is:
   ```bash
   propertiesui.bat/propertiesui.sh
   [-conntype SOAP [-host host_name] [-port port_number] | RMI [-host host_name] [-port port_number] | NONE]
   [-jndiName jndi_name]
   ``
   Where `conntype` specifies the type of connection to be used, the default value being "SOAP".
   A `conntype` of "NONE" means that no server connection is made and certain operations are performed in local mode.
   `host` specifies the name of the host used for the SOAP or RMI connection, the default value being "localhost".
   `port` is the number of the port used for the SOAP or RMI connection.
   `jndiName` is the JNDI name that is in the entry for event runtime properties, as shown under the Resource environment entries section of the WebSphere Application Server administrative console.

2. Enter a user ID and password:
   - If security is enabled on WebSphere Application Server, you must enter a valid user ID and password.
   - If the user specified is not an Administrator, the user must have one of the following security roles:
     - adminsecuritymanager
     - deployer
     - configurator
   - If security is disabled on WebSphere Application Server, you can enter a user ID of your choice or leave the field blank. A password is not required.

3. Modify the existing properties as required. For information about system properties, see Properties.

4. To add a custom property:
   a. Select Full Configurator
   b. Click Edit > Add a Custom Property
   c. Click A new property and change it to the name of the custom property.
   d. Click value? and enter the value of the custom property.
e. If you want to remove a custom property, right-click the property name and click **Delete Custom Property**. The property is deleted without confirmation.

5. Click **File > Save Properties** to save your changes.
6. Restart the event runtime to apply your changes.

**Results**

When the event runtime restarts, the changes you made to the properties are applied.

**Configuring how step data is stored**

You might want to configure how step data is stored for performance reasons.

**About this task**

In Decision Server Events, you can define complex events, where one event has a dependency on another event or action. A complex event is a set of events linked through a common context ID, using the system context, or a context definition.

When a context is initiated, each activity in the process is stored as a step in the context table that was created during the installation, by using persistent storage.

You can configure how that step data is stored, if you want to configure your system for performance reasons.

**Configuring the event runtime to store step data in memory**

You might choose to store step data in memory if your contexts generally have a short life.

**About this task**

Storing a step in a database table entails a certain amount of database access and the context does not go away unless it is manually purged from the table, or the context is ended by a terminating condition in a context definition. If you know that your contexts generally have a short life and then disappear, you can store all steps in memory, avoiding the database access.

In a clustered environment, if `as.director.server.eventflow.persistentStepStore` is set to `false` and Decision Server Events is using the default memory store (as opposed to an ObjectGrid cache, which can be set by using another property), then any context table information is lost on server failover, although the event processing fails over to another server. This situation is also the case on server startup: if servers are still starting, and therefore ObjectGrid is rebalancing its partitions, it is possible for the partitions to move from server to server, and therefore context table information might be lost.

**Procedure**

To store contexts in memory instead of a database table, add the following custom properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Purpose</th>
<th>Required</th>
</tr>
</thead>
</table>

Table 21. Custom properties to be set.
Table 21. Custom properties to be set. (continued)

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>as.director.server.eventflow.persistentStepStore</td>
<td>false</td>
<td>Turns off persistent storage and causes step data to be stored in memory</td>
<td>yes</td>
</tr>
<tr>
<td>as.director.server.eventflow.cacheStepStoreTimeout.ms</td>
<td>5000 milliseconds (default)</td>
<td>Determines amount of time, in milliseconds, for which an instance of step data is kept in memory</td>
<td>no – default is 5000 ms (5 seconds)</td>
</tr>
</tbody>
</table>

See “Setting properties” on page 117 for detailed information about setting up these properties.

**Configuring the event runtime to store context data in ObjectGrid**

The context table uses the default cache. As a potentially higher-performing alternative, you can configure the context table to use ObjectGrid. You cannot use ObjectGrid as the context table on z/OS.

**Procedure**

1. To use ObjectGrid as an in-memory cache for the context table, set `as.director.server.eventflow.gridStepStore` to `true`.
2. Switch off database backing by setting `as.director.server.eventflow.persistentStepStore` to `false`.
3. Optionally, configure ObjectGrid to provide data replication capabilities:
   a. Locate `objectGridDeployment.xml` in `wberuntime.jar`, which is in `was_install_dir/profiles/profileName /installedApps/cellName/` `wberuntimeear.ear`, where `profileName` is the name of your WebSphere Application Server profile.
      `cellName` is the name of the WebSphere Application Server cell.
   b. Modify the file to include `<mapSet name="mapSet"
      numberOfPartitions="10" minSyncReplicas="0" maxSyncReplicas="1"
      maxAsyncReplicas="0" numInitialContainers="1">`

   If you have configured the event runtime in a cluster, you can configure ObjectGrid to provide data replication capabilities.

**Configuring the event runtime to record history**

If you want to record history, you must configure the event runtime to record history. Otherwise, you cannot monitor events, actions, filters, or data in real time through user-defined charts, nor can you run reports or use track events. Recording history preserves the delivery of real-time history records across system restarts.

**About this task**

You can set the event runtime history to be persistent, or non-persistent. By default, the history is non-persistent. Non-persistent history messages that are currently being processed but have not yet been written to the history database might be lost when the event runtime restarts. To ensure that all history data is preserved when the event runtime is restarted, you must enable persistent history by setting history properties.

You must configure the event runtime to record history if you are planning to:
• Use the event widgets to monitor events, actions, filters, and data in real time, either by using the Event Tester widget or through visualization of user-defined charts.
• Run reports from Administration.
• Use track events to count the number of events or keep running totals for thresholds and alerts.

History is stored in a set of history tables that were created when you installed Decision Server Events.

**Important:** If you enable persistent history, there is a significant performance impact.

**Procedure**
1. Edit the following properties:
   - as.director.server.history.enableModule - Determines whether the history module is loaded when the event runtime starts. Ensure that the property value is set to true.
   - as.director.server.history.enableRecording - Determines whether history is recorded in the history tables. Set the property value to true to record history. Set this value to false to stop recording history.
   - as.director.server.history.useJMS - Determines whether history messages that are being processed but have not been written to the history database are non-persistent or persistent. Set this value to true to ensure that all history data is preserved when the event runtime is restarted. Setting this property to true has a significant impact on performance.
   - as.director.bet.install - Set the value to true if the Event Tester widget is installed. This setting ensures that all history records required by the tester widget are captured. This property overrides the Record this data in history option, which you can set in the Event Designer editors.

See "Setting properties" on page 117 for more information. Properties are displayed in the History Settings subsection of the Server section.

2. You must restart the wberuntimeear application, by using the WebSphere Application Server Network Deployment administrative console, to ensure that changes to the properties are in effect.

3. Optional: In the Event Designer Action Editor, Event Editor, Filter Editor and Rule Editor, the Record this data in History option is selected by default. To increase product performance and reduce unused history data, you can clear this option in one or all of the editors. However, if you clear the Record this data in History setting in the editors, but the as.director.bet.install property is set to true, then the property setting overrides the editor setting, and history is recorded.

**Configuring the durable event queue**

Decision Server Events uses a file system based event queue to ensure that durable events are not lost if a server fails.

**About this task**

The default location for this property is was_install_dir/profiles/profileName/wbe/nodeName/serverName/queue_data where

profileName is the name of your WebSphere Application Server profile.
nodeName is the name of the WebSphere Application Server node.

serverName is the name of the WebSphere Application Server server.

The location of the durable event queue can be changed by using the following property:

\[
\text{as.director.server.durabledirectory}
\]

If you experience disk contention, you might want to change the location of this queue to move it to a faster disk subsystem. If you are using Decision Server Events in a cluster, ensure that each cluster member has access to the path you specify. For example, if you change the location to be a path on the E: drive every server needs access to the E: drive.

In a clustered environment, events that were stored in this queue are not failed over to other servers in the cluster. These events are only processed when the server they were being processed on is started again.

**Configuring the log size of the durable event queue**

**Procedure**

To change the size of the durable event queue log, use the following property:

<table>
<thead>
<tr>
<th>Property name</th>
<th>Default</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>as.director.server.DurableEventQueue.size</td>
<td>20 MB</td>
<td></td>
</tr>
</tbody>
</table>

The default durable event queue log size of 20 MB is suitable for most workloads. However, if you see `com.ibm.wbe.objectManager.TransactionCapacityExceededException` in your SystemOut.log file then you might benefit from increasing the default size for this property.

For details of how to set this property, see Chapter 6, “Setting properties” on page 117.

**Configuring how time-delayed event rules and actions are handled**

The evaluation of events and actions is typically done as soon as possible, but you can delay the evaluation based on user-defined criteria. For example, if a server shuts down, there might be event rules and actions scheduled for evaluation while the server is down. You can configure how the event runtime handles time-delayed event rules and actions.

**About this task**

You can tell the event runtime how to deal with queued events and actions after a shutdown. Time-delayed event rules and actions are stored as scheduled tasks in the time_based_asset table that was created in the database manager of your choice when installing. The maxtardiness property value is also evaluated when the event runtime processes any delayed event or action.
Procedure

Edit the following properties:

Table 23. Property values to set for time-delayed processing

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>For event rules:</td>
<td>The maximum amount of time, in minutes, beyond the scheduled time when a time-delayed event rule is still run.</td>
</tr>
<tr>
<td>as.director.server.timebased.event.maxtardiness.minutes</td>
<td>The default value is -1. The event runtime will evaluate all queued events or invoke all queued actions as soon as the server has started.</td>
</tr>
<tr>
<td>For actions:</td>
<td>The maximum amount of time, in minutes, beyond the scheduled time when a time-delayed action is still run.</td>
</tr>
<tr>
<td>as.director.server.timebased.action.maxtardiness.minutes</td>
<td>The default value is -1. The event runtime will evaluate all queued events or invoke all queued actions as soon as the server has started.</td>
</tr>
</tbody>
</table>

If the property has a value of 0 (zero), this is equivalent to the default value of -1.

For details of how to set these properties, see “Setting properties” on page 117.

If you are using Properties, these properties are in the Timebased Events and Actions subsection of the Server section.

Configuring multiple Decision Server Events profiles to coexist

If you have multiple profiles of Decision Server Events on one computer, you must configure the WebSphere eXtreme Scale catalog service for the profiles to coexist.

About this task

If you have a Decision Server Events profile and you have created a second profile on the same computer, you must configure the second profile for either profile to work correctly.

Procedure

1. Start the WebSphere Application Server administrative console. Ensure that you connect to the new WebSphere Application Server profile.
2. Click System administration > WebSphere eXtreme Scale > Catalog service domains.
3. Click New and enter values for the properties:
   - Name: Enter a name for the catalog service domain, for example, WODM Catalog Service.
   - Select Existing application server under Catalog Server Endpoint, then select your WebSphere Application Server server from the drop-down list.
   - Client Port: Enter a port number that is not in use. Do not use the port that is used by the catalog server on the first profile, which is 6601 by default. You might use 6602 for the second profile, for example.
   - Click OK, then click Save to save your changes.
4. Restart WebSphere Application Server for the changes to take effect.
Configuring technology connectors on a separate computer

Technology connectors can be run on the same computer as the event runtime, or they can be run on a separate computer or operating system.

About this task

If you want to run the technology connectors on a separate computer to the computer where the event runtime is installed, and you have already installed the connectors on another computer, you need to configure the connectors on the second computer to connect to the event runtime.

In the following instructions, to differentiate between the two computers involved, the computer on which the event runtime is to be installed is referred to as the Runtime computer, and computer where the connectors operate remotely is referred to as the Connectors computer.

Before installing a connector, make sure that you have a copy of Application Client for WebSphere Application Server installed on the Connectors computer. The client software must be pointing to the same instance of WebSphere Application Server on which the event runtime is deployed.

If you are planning to use WebSphere MQ as the JMS provider, you must also have the WebSphere MQ client installed.

To set up the event runtime and the technology connectors on separate computers:

Procedure

1. You must modify the environment script on the Connectors computer to point to the WebSphere Application Server on the Runtime computer:
   a. On the Connectors computer, locate a directory called config, which is in the home directory of Decision Server Events (by default on Linux and UNIX /opt.ibm/ODM851/ODM and on Windows C:\Program Files\IBM\ODM851).
   b. In the config directory, locate a script called setenv.sh (Linux and UNIX) or setenv.bat (Windows). You must set the values of the two variables in the script on the Connectors computer. For more information, see "Editing the setenv files" on page 108 and Environment variables.
   c. Edit the setenv script. On the WASADMINHOST line, enter the host name for the Runtime computer (for example, on Linux and UNIX this host name might be "WASADMINHOST=computer1.MyCompany.com").
   d. Continue to edit the setenv script. On the WASBOOTSTRAPPORT line, enter the bootstrap port number of the Runtime computer (for example, on Windows this bootstrap port number might be WASBOOTSTRAPPORT=2809).
   e. Save the setenv script
2. On the Runtime computer, if the database server definition uses localhost as the setting for hostname (as.director.common.db.default.dbhost), modify the setting of this property to point to the full hostname of the Runtime computer.

Configuring the User Console connector to use a WebSphere Application Server data source

The User Console connector accesses its tables by using data connections. If you have configured Decision Server Events to use a WebSphere Application Server data source when accessing the event runtime, you must make further
configuration changes to enable the User Console connector to function. You must do this task if you are using an embedded Apache Derby database.

**Before you begin**

You must define the User Console connector. See [User Console connector](#). If you have not configured Decision Server Events to use a WebSphere Application Server data source when accessing the event runtime, no further configuration is required.

**About this task**

If the event runtime database is an embedded Apache Derby database, you must create a separate database to host the tables for the User Console and then change some property settings so that the User Console connector accesses its tables in the separate database.

If the event runtime database is hosted by any other supported database manager (including Derby Network Server, which is included in your WebSphere Application Server installation), you can change property settings so that the User Console connector uses the Decision Server Events data connections to connect to the User Console tables. You can still, for performance reasons, access the event runtime database by using WebSphere Application Server data sources.

To change the configuration so that the User Console connector accesses its tables by using the Decision Server Events data connections, and not the WebSphere Application Server data sources:

**Procedure**

1. If the event runtime database is hosted by embedded Apache Derby, create a new database for the User Console tables. The new database can be hosted by any supported database but the instructions in this step assume that you are using Derby Network Server, which is already installed as part of WebSphere Application Server.
   a. Start Derby NetworkServer. In the `was_install_dir/derby/bin/networkServer` directory, run `startNetworkServer.bat` (Windows) or `startNetworkServer.sh` (Linux and UNIX). Derby NetworkServer starts in a command window with the following message: Server is ready to accept connections on port 1527. Port 1527 is the default port number on which Derby NetworkServer listens for connections.
   b. In the `was_install_dir/derby/bin/networkServer` directory, start the `ij` command environment by running `ij.bat` (Windows) or `ij.sh` (Linux and UNIX).
   c. In the `ij` command window, create a new database for the User Console tables by running the following command (all on one line):
      ```
      CONNECT 'jdbc:derby:<InstallDir>/derbyuserconsole;create=true';
      ```
      (on Windows) or
      ```
      CONNECT 'jdbc:derby:<InstallDir>/derby/userconsole;create=true';
      ```
      (on Linux or UNIX)
   d. In the `ij` command window, create the tables in the new database, by running the `derbydb.sql` that is located at: `<InstallDir>/config/db/derbydb.sql`. The script creates all the tables that are required of the User Console.
2. Configure the User Console to use the new database by setting the following properties (if you use Decision Server Events Properties, the properties are on the Connector page):
   a. If you are using Derby Network Server:
      
      | Property                               | Value                        |
      |----------------------------------------|------------------------------|
      | as.director.connectors.pem.dbclass     | com.ibm.wbe.db.DerbyDB       |
      | as.director.connectors.pem.instance    | `<InstallDir>/derby/userconsole` |
      | as.director.connectors.pem.dbport      | 1527                         |
      | as.director.connectors.pem.dbhost      | localhost                    |
      | as.director.connectors.pem.username    | APP                          |
      | as.director.connectors.pem.password    | derbyadmin                   |

   b. If you are using another supported database:
      
      | Property                               | Value                        |
      |----------------------------------------|------------------------------|
      | as.director.connectors.pem.dbclass     | com.ibm.wbe.db.DB2DB (DB2)   |
      |                                       | com.ibm.wbe.db.OracleDB (Oracle) |
      |                                       | com.ibm.wbe.db.MssDB (SQL Server) |
      | as.director.connectors.pem.instance    | The name of the database     |
      | as.director.connectors.pem.dbport      | The port on which the database is listening |
      | as.director.connectors.pem.dbhost      | The host name of the database |
      | as.director.connectors.pem.username    | The user ID that Decision Server Events uses to access the database |
      | as.director.connectors.pem.password    | The password associated with the user ID |

c. Enable each of the fields listed in the table in the previous step. You must enable fields as well as entering the property values.

d. Ensure that you have enabled history before you continue, otherwise no data is produced for the User Console to display.

3. Stop then restart WebSphere Application Server to apply the properties changes.

4. Ensure that database is running, then start the Connectors.

What to do next

Now you are ready to use the User Console.

Tuning your event runtime configuration

Some tuning of the system might lead to higher performance when processing events.

About this task

When considering how you tune your system, there are two main areas to consider:

- The event runtime and WebSphere Application Server
Tuning with the event runtime and WebSphere Application Server

About this task

There are several tasks that you can do to tune your system by configuring the event runtime and WebSphere Application Server. You can select from the following tasks that might improve performance:

Procedure

- Turn off logging. For more information, see Logs.
- Turn off the recording of history, if not required. See “Configuring the event runtime to record history” on page 122.
- Use connectors only if required.
  If events are already in Decision Server Events format, better performance might be achieved by avoiding the use of connectors. Send events directly to the event destinations and receive actions directly from the action destinations. For example, consider using the destinations jms/eventDestination, jms/durableEventDestination, jms/actionTopic and jms/durableActionTopic.
- Ensure that there are sufficient event rule processing threads. If you are unable to achieve full processor utilization on the Decision Server Events server, consider increasing the value of the property as.director.server.ruleProcessorInstances.
- Tune the Decision Server Events database:
  - Ensure that the database is tuned (or auto tuned) for the workload. The sizes of Log File and Buffer Pool are important. See the documentation for your database manager.
  - Consider using a remote database with fast disk subsystems for data and logs. Ideally place data and logs on separate devices.
- Tune the JVM:
  - The default heap settings (minimum 512MB, maximum 1024MB) are acceptable for many applications. The optimal tuning depends on the available free memory and the nature of the workload, but the following configurations are suitable for a wide range of system workloads. You can set these parameters by using the WebSphere Application Server administrative console (Application Servers > server1 > Java and Process Management > Process Definition > Java Virtual Machine):
    - In this configuration, a 32-bit JVM with 2 GB of free memory, the first and second parameters set the minimum and maximum heap size. The third parameter sets a generational garbage collection policy with a 1024 MB nursery heap, meaning that 1024 MB of the heap is used for short lived objects and the remainder of the heap is used for longer lived objects:
      - Initial heap size: 1280 (MB).
      - Maximum heap size: 1280 (MB).
      - Generic JVM arguments: -Xgcpolicy:gencon -Xmn1024M
    - In this configuration, a 64-bit JVM with 8 GB of free memory, you set a 4096 MB heap with a 2048 MB nursery:
      - Initial heap size: 4096 (MB).
      - Maximum heap size: 4096 (MB).
      - Generic JVM arguments: -Xgcpolicy:gencon -Xmn2048M
    - For more information, see Tuning the JVM.
Increase the concurrency:

If you are using the File System, HTTP, JDBC, JMS, or SOAP action connectors and you see messages for actionTopic or durableActionTopic, increasing the concurrency might improve the rate at which actions are processed.

1. In the navigation tree of the WebSphere Application Server administrative console, click Resources > JMS > Activation specifications, then select the activation specification that you want to modify. Activation specifications for the action connectors are:
   - File System: wbeca_file_as
   - HTTP: wbeca_http_as
   - JDBC: wbeca_jdbc_as
   - JMS: wbeca_jms_as
   - SOAP: wbeca_soap_as

2. Modify the activation specification according to the messaging provider that you are using:
   - If you are using WebSphere Application Server default messaging, modify the Maximum concurrent MDB invocations per endpoint.
   - If you are using WebSphere MQ as the messaging provider, modify the value of Maximum server sessions on the Advanced properties window.

Tuning with JMS messaging

About this task

There are several tasks that you can do to tune your system that are specific to JMS messaging. You can select from the following tasks that might improve performance:

Procedure

- For persistent messaging, consider using fast disk subsystems for data and logs. Ideally, place data and logs on separate devices.
- If you are using WebSphere MQ as the JMS provider:
  - The key tuning parameters relate to queue manager logs, channels, and listeners and queue buffer size. See Configuring and tuning WebSphere MQ for performance on Windows and UNIX.
  - Consider delivering the messages in batches from the input topic to Decision Server Events. This method is useful for non-persistent, non-durable WebSphere MQ JMS events. The batch size is configured by using the WebSphere Application Server administrative console (Servers > Application servers > server1 > Messaging > Message Listener Service > Listener Ports > wbe_events > Maximum Messages). However, if one of the messages in the batch fails, the whole batch is put back on the queue for processing.
- If you are using WebSphere Application Server default messaging as the JMS provider:
  - The key tuning parameters relate to the choice of message reliability level, activation specifications, and size of the discardable data buffer. See Tuning messaging performance with service integration technologies.
  - Activation specification: Particularly for non-durable JMS events, consider delivering the messages in batches from the input topic to Decision Server Events. This method can deliver events more efficiently. Use the WebSphere Application Server administrative console (for example, Resources > JMS > Activation specifications > wbe_events, and set Maximum batch size).
Verifying your Decision Server Events configuration

You can verify that Decision Server Events, and the event widgets, are correctly installed and configured before you use them. Check that Decision Server Events is properly installed and configured by loading the supplied verification event project and sending an event through the event runtime.

Before you begin

Install and configure Decision Server Events, including configuring a JMS messaging provider.

Ensure that you have installed Event Designer. The event runtime can be installed on a separate computer.

If you turned on WebSphere Application Server security during installation, you must configure user authentication to the event runtime before verifying the installation. See “User authentication” on page 135.

Procedure

1. On the computer on which you have installed the event runtime, ensure that WebSphere Application Server is running.
2. Check that the event runtime application has been installed and started on WebSphere Application Server:
   a. Start the WebSphere Application Server administrative console. See Starting the WebSphere Application Server administrative console
   b. Log in to the WebSphere Application Server administrative console with a user ID of your choice.
   c. In the navigation tree of the WebSphere Application Server administrative console, click Applications > Application Types > WebSphere enterprise applications. Ensure that the following application is listed and shown as started:
      • wberuntimeear
      • EventWidgetsEar

   Ensure that the following application is listed and shown as started if you have prepared the environment for testing:
      • wbtesterear
3. On the computer on which you installed the event runtime, create a directory called DecisionServerTest and make a note of the location in which you created the directory.
4. In Event Designer, right-click in the Event Explorer view then click Import. The Import wizard opens.
5. In the Import wizard, click Event Project from XML File > Next.
6. Click Browse, navigate to <InstallDir>\connectors\validation and select the event project called installValidation.xml. Click Next. Accept the default settings for the rest of the wizard and click Finish to import the event project into your workspace.
7. In the Event Explorer view, expand the event project, expand the Outgoing folder, and open the Outgoing action, shown in the following screen capture:
8. In the Action editor, click the **Connector** tab, shown in the following screen capture:

9. In the **Folder on that computer** field, replace the existing value with the complete path that points to the location of the *DecisionServerTest* directory that you created. Save your changes.

10. To deploy the event project, right-click the event project in the Event Explorer view and select **Deploy**. The Deploy wizard opens, shown in the following screen capture:
11. Select **Deploy all assets** and click **Next**.
12. Enter the connection details for your event runtime, shown in the following screen capture:

Ensure that the host name and port values are correct for the WebSphere Application Server instance in which the event runtime is deployed (the port
value is the WC_defaulthost port, which can be determined through the WebSphere Application Server administrative console). Enter the user ID and password, if security is enabled.

13. Click **Finish** to deploy the event project. A message is displayed indicating that the event project is deployed successfully. Close Event Designer.

14. **Start the stand-alone technology connectors.** See Starting and stopping connectors in the technology connector application on Windows, Linux, and UNIX

An example of a successful connector startup on Windows:

```
C:\Program Files\IBM\ODM851\connectors\bin>connectors.bat
*********************************************************
IBM Decision Server technology connectors (c)Copyright IBM Corp. 2003, 2012
Version: IBM Operational Decision Manager 8.5
Locale: en_GB
Java version: 1.6.0
*********************************************************
BEER0561I: The technology connectors are waiting for wberuntimeear.
BEER0644I: The technology connectors established communication with wberuntimeear
BEER0590I: Connector runtime checkpoint: 2009-04-14 08:58:10.718
BEER2608I: Authentication mode: None
BEER2609I: Event Runtime users group: WBEUsers
BEER2610I: Event Runtime administration group: WBEAdmins
BEER2611W: *** WARNING: SECURITY IS NOT ENABLED ***
BEER2612W: User maestro is granted full access to the event runtime
BEER0611I: Initialization is complete
BEER0670I: JNDI provider URL: corbaloc:iiop:localhost:2809
```

For more information about starting connectors, including details of the optional parameters, see Running technology connectors.

15. Open a command window (Windows) or a terminal emulator (Linux and UNIX), navigate to the `<InstallDir>/connectors/bin` directory, and send a test event to the event runtime by running a `cmdline` command:

- On Windows, run `cmdline.bat -uid userid -pwd password <test-event-file>`, where `userid` and `password` are a valid user ID and password combination that grants access to WbeBus, and `<test-event-file>` is the complete path to where the `installValidation_event.xml` file is located, for example, `cmdline.bat -uid myname -pwd mypassword "C:\Program Files\IBM\ODM851\connectors\validation\installValidation_event.xml"`.

- On Linux or UNIX, run `<InstallDir>/connectors/bin/cmdline.sh -uid userid -pwd password <test-event-file>`, where `userid` and `password` are a valid user ID and password combination that grants access to WbeBus, and `<test-event-file>` is the complete path to where the `installValidation_event.xml` file is located, for example, `<InstallDir>/connectors/bin/cmdline.sh -uid myname -pwd mypassword/opt/ibm/ODM851/ODM/connectors/validation/installValidation_event.xml`.

A new file called `ValidationTest random-number`.xml is put in the DecisionServerTest directory, where `random-number` is a system-generated number that makes the file name unique.

**Note:** In `cmdline.sh` there are relative paths (`/..`) to `setenv.sh` files. You previously modified these relative paths in the component directory, but not in the product installation directory that the relative path references (the component directory is a copy of part of the product installation directory). You can fix this issue in one of two ways:
Either, edit the setenv.sh file to include a parameter that specifies the location of the component directory.

Or, copy the connectors directory into the component directory. If you use this solution, you must change the directory path to `<InstallDir>/connectors/bin/cmdline.sh`.

**Results**

You have now verified your installation.

---

**Securing Decision Server Events**

To secure Decision Server Events, both WebSphere Application Server security and Decision Server Events security settings must be configured.

**About this task**

To configure WebSphere Application Server security, see [WebSphere Application Server Security](#).

To configure Decision Server Events security, see ["User authentication."](#).

The default and preferred security provider is WebSphere Application Server.

If you are migrating from a previous version of Decision Server Events where you used a different security provider, you can continue to use this provider, but you must complete additional steps when configuring Decision Server Events and the event widgets. See ["Securing access to charts" on page 141](#).

**Related reference:**

- [Authentication with the Administration and User Console interfaces](#)
- [User roles and administrative user roles](#)

**User authentication**

User authentication to the event runtime can be implemented for Decision Server Events by using WebSphere Application Server. This method is the default and the preferred method.

**About this task**

User authentication controls who has access to the event runtime.

For more details about user roles and administrative roles, see [User roles and administrative user roles](#).

You can permit access to the event runtime without any user authentication, though this permission does mean that anyone can access the event runtime and change assets.

If you are migrating from a previous version of Decision Server Events where you used a different security provider, you can continue to use that security provider.
**Configuring user authentication by using WebSphere Application Server**

If you want to secure access to the event runtime, configure Decision Server Events to use WebSphere Application Server to define authorized users and groups. This method is the default and the preferred method.

**About this task**

To enable security by using WebSphere Application Server, you must:

- Enable WebSphere Application Server security.
- Configure Decision Server Events to use WebSphere Application Server for user authentication.
- Map users and user groups to Decision Server Events roles.
- Configure WebSphere Application Server administrative roles.

**Procedure**

1. Start the WebSphere Application Server administrative console. For instructions, see [Starting the WebSphere Application Server administrative console](#).

2. Enable WebSphere Application Server security:
   a. In the WebSphere Application Server administrative console, click **Security** > **Global security**.
   b. Select **Enable administrative security**.
   c. Ensure **Enable application security** is selected.
   d. Ensure **Use Java 2 security to restrict application access to local resources** is selected.

   For more information, see [Enabling security](#).

3. To configure Decision Server Events to use WebSphere Application Server for user authentication, change these properties:

   **Table 26. Properties to be changed.**
   
<table>
<thead>
<tr>
<th>Property</th>
<th>Value to set</th>
</tr>
</thead>
<tbody>
<tr>
<td>as.director.common.authentication.method</td>
<td>AppServer</td>
</tr>
<tr>
<td></td>
<td>This value is the default value for Decision Server Events but no authentication or authorization takes place unless you have also enabled security in the previous step.</td>
</tr>
<tr>
<td>as.director.server.authentication.username</td>
<td>A user ID that is the primary administrative user for WebSphere Application Server.</td>
</tr>
<tr>
<td>as.director.server.authentication.password</td>
<td>The password associated with the user ID.</td>
</tr>
</tbody>
</table>

   For details of how to set these properties, see [“Setting properties” on page 117](#).

   If you are using Properties, the properties are displayed in the **Authentication** minor section of the **Common** section.

4. Set the user ID of the primary administrator:
   a. In the WebSphere Application Server administrative console, click **Security** > **Global security**.
   b. Ensure that **Available realm definitions** is set to **Local operating system**.
   c. Click **Configure**
d. In the **Primary administrative user name** field, type the user ID that is used to log in to the WebSphere Application Server administrative console to do administrative tasks after security is enabled.

e. Apply your changes.

5. Configure WebSphere Application Server administrative roles.

a. In the WebSphere Application Server administrative console, click **Security > Global security > Administrative user roles > Add**

b. In the **Search string** field, search for the user IDs to which you want to assign the roles, and click **Search**

c. In the **Available** field, select the user IDs to which you want to assign the roles and move them to the **Mapped to role** field, by using the arrows.

d. Under **Role(s)**, click either **Operator** or **Administrator**.

e. Click **OK** and **Save**.

6. Map users and user groups to Decision Server Events roles. See [Security role to user or group mapping](#).

   Various roles can be assigned to users and groups. For more information, see [User roles and administrative user roles](#).

   When you are using WebSphere Application Server as the security provider, you cannot change the role names.

   a. In the WebSphere Application Server administrative console, click **Applications > Application Types > WebSphere enterprise applications**.

   b. Click **wberuntimeear**, then click **Security role to user/group mapping**.

   c. From the list of the roles, select the roles that you want to map (assign) to users, click **Map Users**. For more information about mapping users to roles, see [Security role to user or group mapping](#).

   d. In the **Search string** field, search for the user IDs to which you want to assign the roles, and click **Search**.

   e. In the **Available** field, select the user IDs to which you want to assign the roles and move them to the **Selected** field, by using the arrows.

   f. Click **OK** and **Save**.

7. After you have made these configuration changes, restart WebSphere Application Server to ensure that the changes are in effect.

**Results**

Decision Server Events is now configured to use WebSphere Application Server to authenticate users.

**Configuring user authentication by using LDAP**

If you are migrating from a previous version of Decision Server Events where you used LDAP (Lightweight Directory Access Protocol), you can continue to use it.

**About this task**

User authentication to the event runtime can be configured by using WebSphere Application Server. This method is the default and preferred method. WebSphere Application Server supports the use of LDAP for user registries. For more information, see [Configuring Lightweight Directory Access Protocol user registries in the WebSphere Application Server information center](#) and "Configuring user authentication by using WebSphere Application Server" on page 136.
To continue using the LDAP support as provided in earlier versions of Decision Server Events, use the following steps.

**Procedure**

1. Use the appropriate LDAP administrative utility to create the following groups:

<table>
<thead>
<tr>
<th>Access level</th>
<th>Name of group to be created</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td>WBEAdmins.</td>
<td>WBEAdmins</td>
</tr>
<tr>
<td></td>
<td>If you want the admin group to reference a different name, create a custom property called <code>as.director.common.authentication.adminGroup</code> with a value of the group name.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>However</strong>, if you define this property, then the WBEAdmins group is no longer an admin group, and any users in that group no longer have administrative privileges (unless they are also a member of the new admin group).</td>
<td></td>
</tr>
<tr>
<td>User</td>
<td>Choose any name, but the name must match the value of <code>as.director.common.authentication.userGroup</code> set in Properties.</td>
<td>WBEUsers</td>
</tr>
</tbody>
</table>

2. For details of how to set these properties, see "Setting properties" on page 117.

3. Change the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>as.director.common.authentication.method</code></td>
<td>Select LDAP.</td>
</tr>
<tr>
<td><code>as.director.common.authentication.server</code></td>
<td>Replace <code>&lt;HOST-NAME&gt;</code> with the name of the server where LDAP resides. Multiple server names can be entered, separated by a comma. If the first server in the list is unavailable, authentication is attempted against the next and subsequent servers in the list.</td>
</tr>
<tr>
<td><code>as.director.common.authentication.namecontext</code></td>
<td>Remove <code>dc=&lt;MYDOMAIN&gt;</code> and replace <code>&lt;MYCOMPANY&gt;</code> with company name.</td>
</tr>
<tr>
<td><code>as.director.server.authentication.member.filter</code></td>
<td>Enable.</td>
</tr>
<tr>
<td><code>as.director.server.authentication.member.attribute</code></td>
<td>Enable.</td>
</tr>
<tr>
<td><code>as.director.server.authentication.userid.format</code></td>
<td>Only for internal use, do not change.</td>
</tr>
<tr>
<td><code>as.director.server.authentication.security.method</code></td>
<td>Set to one of:</td>
</tr>
<tr>
<td></td>
<td>• none (default)</td>
</tr>
<tr>
<td></td>
<td>• simple</td>
</tr>
<tr>
<td></td>
<td>• strong</td>
</tr>
<tr>
<td><code>as.director.common.authentication.security.protocol</code></td>
<td>Set to one of:</td>
</tr>
<tr>
<td></td>
<td>• unspecified (default)</td>
</tr>
<tr>
<td></td>
<td>• ssl</td>
</tr>
<tr>
<td><code>as.director.common.authentication.userGroup</code></td>
<td>Set the value to match the name of the user group defined in the Active Directory (the default is WBEUsers).</td>
</tr>
</tbody>
</table>
Table 28. Properties to be changed. (continued)

<table>
<thead>
<tr>
<th>Property</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>as.director.server.authentication.username</td>
<td>Replace admin with a user ID that is registered in Active Directory in the Administrator or User groups (set up in step 1 on page 138).</td>
</tr>
<tr>
<td>as.director.server.authentication.password</td>
<td>Replace admin with the password associated with the user ID.</td>
</tr>
</tbody>
</table>

If you are using Properties, the properties are in the **Authentication** minor section of the **Common** section.

**Configuring user authentication by using Microsoft Windows Active Directory**

If you are migrating from a previous version of Decision Server Events where you used Windows Active Directory, you can continue to use it.

**About this task**

On Windows, user authentication to the event runtime can be configured by using WebSphere Application Server. See “Configuring user authentication by using WebSphere Application Server” on page 136. This method is the default and the preferred method.

To continue using Windows Active Directory, use the following steps.

**Procedure**

1. In Active Directory, create the following groups:

   **Table 29. Groups to create in Active Directory.**

<table>
<thead>
<tr>
<th>Access level</th>
<th>Name of group to be created</th>
<th>For example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td>WBEAdmins.</td>
<td>WBEAdmins</td>
</tr>
</tbody>
</table>
<pre><code>   | If you want the admin group to reference a different name, create a custom property called as.director.common.authentication.adminGroup with a value of the group name. |
   | **However,** if you define this property, then the WBEAdmins group is no longer an admin group, and any users in that group no longer have administrative privileges (unless they are also a member of the new admin group). |
</code></pre>
<p>| User         | Choose any name, but the name must match the value of as.director.common.authentication.userGroup set in Properties. | WBEUsers    |</p>

2. For details of how to set these properties, see “Setting properties” on page 117.

3. Change these properties:

   **Table 30. Properties to change.**

<table>
<thead>
<tr>
<th>Property</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>as.director.common.authentication.method</td>
<td>Select Active Directory</td>
</tr>
</tbody>
</table>
Table 30. Properties to change. (continued)

<table>
<thead>
<tr>
<th>Property</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>as.director.common.authentication.server</td>
<td>Replace <code>&lt;HOST-NAME&gt;</code> with the name of the server where Active Directory resides. Multiple server names can be entered, separated by a comma. If the first server in the list is unavailable, authentication is attempted against the next and subsequent servers in the list.</td>
</tr>
<tr>
<td>as.director.common.authentication.namecontext</td>
<td>Replace <code>&lt;MYDOMAIN&gt;</code> with domain name and <code>&lt;MYCOMPANY&gt;</code> with company name.</td>
</tr>
<tr>
<td>as.director.common.authentication.userGroup</td>
<td>Set the value to match the name of the user group defined in the Active Directory (the default is WBEUsers).</td>
</tr>
<tr>
<td>as.director.server.authentication.username</td>
<td>Replace <code>admin</code> with a user ID that is registered in Active Directory in the Administrator or User groups (set up in step 1 on page 139).</td>
</tr>
<tr>
<td>as.director.server.authentication.password</td>
<td>Replace <code>admin</code> with the password associated with the user ID.</td>
</tr>
</tbody>
</table>

If you are using Properties, the properties are in the Authentication minor section of the Common section.

Configuring user authentication by using User Console

Alternatively, you can use User Console to authorize users and groups by adding them to certain event runtime tables.

About this task

If you want to provide security of access to the event runtime, the default and preferred method is to use WebSphere Application Server security. See “Configuring user authentication by using WebSphere Application Server" on page 136. The event runtime tables and the privileges of each group are described in Authentication with the Administration and User Console interfaces.

The tasks relating to user authentication by using User Console are as follows.

Procedure

1. Enable authentication by using the event runtime database:
   a. For details of how to set these properties, see “Setting properties” on page 117.
   b. Change these properties:

   Table 31. Properties to be changed.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value to set</th>
</tr>
</thead>
<tbody>
<tr>
<td>as.director.common.authentication.method</td>
<td>Database</td>
</tr>
<tr>
<td>as.director.server.authentication.username</td>
<td>user ID</td>
</tr>
<tr>
<td>as.director.server.authentication.password</td>
<td>password of your choice</td>
</tr>
</tbody>
</table>

   If you are using Properties, the properties are in the Authentication subsection of the Common section.
c. Stop and restart the wberuntimeear application in the WebSphere Application Server administrative console. The next time a Decision Server Events user interface or the event runtime is started, a valid user ID and password must be provided.

2. Define new groups:
   a. Start User Console by logging from the common login panel at:
      http://server:port/wbe/common/login.jsp
      where server is the name of the WebSphere Application Server server where Decision Server Events is installed and port is the port number of WC_defaulthost of this server (typically 9080). This value can be determined from the table displayed when you use the Application Server Network Deployment administrative console, click Servers, then Application servers, click the name of the server, and then on Ports.
   b. Move the mouse pointer over the User Accounts button and select Groups from the menu to display the Groups window.
   c. Click the Add a New Group button to display the New Group window. Complete the fields and click Add. Repeat for each group you want.
   d. Click Save configuration to save your changes.

3. Define new users:
   a. Start User Console by logging in from the common login panel at:
      http://server:port/wbe/common/login.jsp
   b. Move the mouse pointer over the User Accounts button and select Users from the menu to display the Users window.
   c. Click the Add a New User button to display the New User window. Complete the fields, including the groups to which the user belongs, and click Add. Repeat for each user you want.
   d. Click Save configuration to save your changes.

4. Add a user to or remove a user from a group:
   a. Start User Console by logging in from the common login panel at:
      http://server:port/wbe/common/login.jsp
   b. Move the mouse pointer over the User Accounts button and select Users from the menu to display the Users window.
   c. Click the Edit button for the relevant user, to display the Edit User window.
   d. Select or clear a group as required, and click Update to complete the edit.
   e. Click Save configuration to save your changes.

5. Disable user authentication:
   If you no longer want to do user authentication, or if Active Directory or LDAP is not available to support authentication, then you can disable the authentication.
   Change the as.director.common.authentication.method property to none. For details of how to set this property, see “Setting properties” on page 117.
   If you are using Properties, the property is in the Common section, in the Authentication subsection. Disabling user authentication means that anyone can access the event runtime and change assets.

Securing access to charts
By using WebSphere Application Server security, you can ensure that only authenticated users can access charts through the use of the chart widgets.
About this task

To ensure that a user must authenticate before accessing chart widgets, follow these steps:

Procedure

1. Ensure that WebSphere Application Server security is enabled. See “Configuring user authentication by using WebSphere Application Server” on page 136.
2. Assign the user to the 'WBEDashboardUsers' role. See “Configuring user authentication by using WebSphere Application Server” on page 136.

Securing the service integration bus

If you are using WebSphere Application Server default messaging as the JMS provider for the event runtime, you must secure the service integration bus to prevent unauthorized access.

Before you begin

Note: This information applies only if you are using default messaging as the JMS provider for the event runtime. If you are using WebSphere MQ as the JMS provider, see “Configuring WebSphere MQ to be the JMS provider” on page 115 for more information.

You must configure the event runtime to use default messaging as the JMS provider:

- Either, when a management or custom profile is created or when the profile is augmented with Decision Server Events.
- Or, at a later time by using the procedure described in “Configuring WebSphere Application Server default messaging to be the JMS provider” on page 113.

When the event runtime is configured to use default messaging as the JMS provider, the service integration bus WbeBus and a number of messaging destinations and activation specifications are configured. If WebSphere Application Server administrative security is enabled when WbeBus is configured, the bus is secured and all authenticated users are authorized to connect to the bus. You can change the configuration of the bus to restrict access to specific user IDs. For more information about service integration bus security, see Securing Service integration.

About this task

Use the WebSphere Application Server administrative console.

Procedure

1. Ensure that security is enabled for the service integration bus WbeBus. For more information, see Securing buses.
2. Review the users and groups in the bus connector role.
   a. Remove the special groups AllAuthenticated and Everyone if these groups are in the role.
   b. Restrict access to only those users and groups that are used to connect to the bus.

For more information, see Administering the bus connector role.
Note: Depending on your requirements, you can add a single user ID to the bus connector role that is shared by the components of the event runtime that connect to the bus.

3. Review the authentication aliases used by the activation specifications to connect to WbeBus. For a list of activation specifications, see Activation specifications. For more information about configuring activation specifications, see Configuring security for message-driven beans that use activation specifications.

   If the service integration bus is secured when the activation specifications are configured, the activation specifications are configured to use a single authentication alias, WbeBusAlias. This alias must be configured with a user ID and password in the bus connector role.

4. Review the authentication credentials used by the event runtime and connector components to connect to the JMS provider, which are set by using the following event runtime properties:

   as.director.server.jms.username
   as.director.server.jms.password

   These properties must be configured with a user ID and password in the bus connector role. For more information about configuring event runtime properties, see “Setting properties” on page 117.
Chapter 4. Configuring the Decision Center consoles on WebSphere Application Server

To be able to use the Decision Center consoles on an instance of WebSphere Application Server, you deploy the provided archives and go through a series of configuration steps.

To configure a cluster, use a profile template. For more information, see "Configuring the Decision Center consoles by using profile templates" on page 20.

Operational Decision Manager 8.5.1 supports WebSphere Application Server 8.0, 8.5, and 8.5.5.

Configuring the Decision Center consoles on WebSphere Application Server for distributed platforms

To be able to use the Decision Center consoles on an instance of WebSphere Application Server, you deploy the provided archives for this server and perform a series of configuration steps.

Before you start: Opening the administration console

To configure the Decision Center consoles on WebSphere Application Server, you must follow specific steps. Some are mandatory, while others are optional.

Before you begin

This guide assumes that you use a profile with administrative security enabled.

About this task

Decision Center includes two consoles: Enterprise and Business. Both consoles work on WebSphere Application Server 8.0 and 8.5.

To configure Decision Center on WebSphere Application Server, you work in the WebSphere Integrated Solutions Console.

Procedure

1. Make sure that you have a WebSphere Application Server profile.
   The predefined name of the first server profile is AppSrv01. Refer to the WebSphere Application Server documentation if necessary.

2. Start the server by clicking Start > All programs > IBM WebSphere > IBM WebSphere Application Server version_number > Profiles > AppSrv01 > Start the server
   Tip: As an alternative, first open the First Steps window from the Windows Start menu, and then start the server, and later the console, from the First Steps window.
3. Start the console by clicking **Start > All programs > IBM WebSphere > IBM WebSphere Application Server version_number > Profiles > AppSrv01 > Administrative console** You might have to pass the security steps of your browser.

4. Log in by entering the user name and password that you defined when you created the profile.
   For example: wasadmin and wasadmin. The WebSphere Integrated Solutions Console opens.

**What to do next**

If you have rule projects that were created with a previous product version, refer to the Migrating topics on how to upgrade the Rule Team Server database schema.

**Note:** A specific integration extension for the IBM Process Server platform is available. For more information, see IBM Operational Decision Manager Integration SupportPacs.

The following table summarizes the steps to configure Decision Center on WebSphere Application Server.

<table>
<thead>
<tr>
<th>Step</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1: Restricting database user permissions</strong></td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Step 2: Configuring database connectivity</strong>&lt;br&gt;on page 148</td>
<td>&quot;Creating a DB2 database on z/OS&quot;&lt;br&gt;on page 148</td>
</tr>
<tr>
<td></td>
<td>Creating a JDBC provider</td>
</tr>
<tr>
<td></td>
<td>&quot;Creating a data source and a connection pool&quot;&lt;br&gt;on page 149</td>
</tr>
<tr>
<td></td>
<td>&quot;Creating J2C authentication data&quot;&lt;br&gt;on page 150</td>
</tr>
<tr>
<td></td>
<td>&quot;Changing the custom properties of your data source&quot;&lt;br&gt;on page 151</td>
</tr>
<tr>
<td></td>
<td>&quot;Connecting to the database&quot;&lt;br&gt;on page 152</td>
</tr>
<tr>
<td><strong>Step 3: Configuring security</strong>&lt;br&gt;on page 152</td>
<td>&quot;Creating users and groups&quot;&lt;br&gt;on page 152</td>
</tr>
<tr>
<td></td>
<td>&quot;Updating the security policies&quot;&lt;br&gt;on page 155</td>
</tr>
<tr>
<td><strong>Step 4: Deploying the Decision Center EAR file</strong>&lt;br&gt;on page 156</td>
<td>&quot;Declaring custom groups&quot;&lt;br&gt;on page 156</td>
</tr>
<tr>
<td></td>
<td>&quot;Deploying the EAR file&quot;&lt;br&gt;on page 157</td>
</tr>
<tr>
<td></td>
<td>&quot;Changing the class loading sequence&quot;&lt;br&gt;on page 159</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Database permission</th>
<th>Use Decision Center</th>
<th>Create the database schema by using the Decision Center console or Ant tasks</th>
<th>Modify the database schema by using the Decision Center console or Ant tasks</th>
<th>Migrate the database schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER TABLE</td>
<td>Not required</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>CREATE INDEX</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>CREATE ROLE</td>
<td>Not required</td>
<td>Not required</td>
<td>Not required</td>
<td>Required</td>
</tr>
<tr>
<td>CREATE SEQUENCE</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>CREATE TABLE</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>CREATE VIEW</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>DROP INDEX</td>
<td>Not required</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>DROP SEQUENCE</td>
<td>Not required</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>DROP TABLE</td>
<td>Not required</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
</tr>
</tbody>
</table>
### Operation

<table>
<thead>
<tr>
<th>Database permission</th>
<th>Use Decision Center</th>
<th>Create the database schema by using the Decision Center console or Ant tasks</th>
<th>Modify the database schema by using the Decision Center console or Ant tasks</th>
<th>Migrate the database schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>DROP VIEW</td>
<td>Not required</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>INSERT TABLE</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>SELECT SEQUENCE</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>SELECT TABLE</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>UPDATE TABLE</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
</tbody>
</table>

### Step 2: Configuring database connectivity

In this set of tasks, you create a JDBC provider, a data source, and a connection pool, and do other optional tasks. You then establish and test the database connection.

**Creating a DB2 database on z/OS**

If you want to use a DB2 database on z/OS, you must create the database schema by running the provided JCL scripts on the z/OS system where DB2 is installed.

**Before you begin**

- Before you create a database, check that you have DB2 for z/OS Version 9.1 or 10 installed, and that you have DB2 Administrator authority to CREATE a database and CREATE & UPDATE tables.
- Before you run JCL scripts on the z/OS system, you must customize the variables as indicated in the comments within each PDSE member. You can customize JCL scripts either by using a manual search and replace or by using an ISPF macro.

**About this task**

After you have customized the JCL variables for your configuration, you can submit the jobs to create a database for Decision Center.

**Procedure**

On the z/OS computer where DB2 is installed, run the following JCL scripts to create the database tables and grant authority to the tables. This is typically a database administrator's task.

1. `InstallDir/teamserver/jcl/HBRDCCDB.jcl`
2. `InstallDir/teamserver/jcl/HBRDCGNR.jcl`

   `++DB2USER++` is the user ID to access DB2.

**Creating a JDBC provider**

Before you can create a data source and connection for Decision Center, you first create a JDBC provider in WebSphere Application Server.
Before you begin

To create a JDBC provider, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 145.

Procedure

1. In the side panel of the console, open Resources > JDBC, and click JDBC Providers.
2. In Scope, select Node=x, Server=y.
   x is the name of your node and y the name of your server.
3. Click New.
4. In Step 1, select the database type, provider type, and a non-XA implementation type.

Table 32. JDBC provider options for DB2 and Derby

<table>
<thead>
<tr>
<th>Database type</th>
<th>Derby</th>
<th>DB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider type</td>
<td>Derby JDBC Provider</td>
<td>DB2 Universal JDBC Driver Provider</td>
</tr>
<tr>
<td>Implementation type</td>
<td>Connection pool data source</td>
<td>Connection pool data source</td>
</tr>
<tr>
<td>Name</td>
<td>Enter a name, for example Rule Execution Server JDBC Provider</td>
<td>Enter, for example, DB2 Universal JDBC Driver Provider for DC</td>
</tr>
</tbody>
</table>

5. Click Next.
6. In Step 2l, confirm or modify the database class path and click Apply, and then click Next.
   Step 3 is a summary.
7. Click Finish.
8. Back on the JDBC providers page, click Save to save the changes directly to the master configuration.

Creating a data source and a connection pool

You must create a data source and a connection pool in WebSphere Application Server V8.0 and V8.5 to enable Decision Center.

Before you begin

To create a data source, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 145.

About this task

If WebSphere Application Server is used in cluster mode, you must define the data source at node level in the cluster (as opposed to cluster level).

Procedure

To create a data source and a connection pool:
1. In the side panel of the console, open Resources > JDBC, and click Data sources.
2. In the **Scope** section, select the scope that you selected for the JDBC provider in “Creating a JDBC provider” on page 148, and click **New**.

3. In **Step 1**, enter a name for the data source in the **Data source name** field and `jdbc/ilogDataSource` in the **JNDI name** field, and click **Next**.

   **Important:** You can enter any name for the data source, but the JNDI name must be `jdbc/resdatasource`, otherwise Rule Execution Server cannot use the data source. However, if you need several data sources for your installation, you must specify multiple JNDI names. In this case, to start the console, you need a URL in this form:
   `<hostName>:<portNumber>/teamserver?datasource=<jndiName>`

4. In **Step 2**, select **Select an existing JDBC provider**, and then select the non-XA JDBC provider that you created in “Creating a JDBC provider” on page 148, and click **Next**.

5. In **Step 3**, enter database properties for the data source and click **Next**.

   The following table shows the minimum set of properties that are required to define the supported databases. If the table does not include your driver, check the WebSphere Application Server documentation for more information.

<table>
<thead>
<tr>
<th>Database</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Universal JDBC Driver</td>
<td>• databaseName: Database name if <code>driverType</code> is set to 4, or a locally cataloged database name if <code>driverType</code> is set to 2</td>
</tr>
<tr>
<td></td>
<td>• <code>driverType</code>: 2 or 4</td>
</tr>
<tr>
<td></td>
<td>For <code>driverType</code> 4:</td>
</tr>
<tr>
<td></td>
<td>• <code>serverName</code>: TCP/IP address or host name</td>
</tr>
<tr>
<td></td>
<td>• <code>portNumber</code>: TCP/IP port number</td>
</tr>
<tr>
<td>DB2 legacy CLI-based</td>
<td><code>databaseName</code>: for example, Sample.</td>
</tr>
<tr>
<td>Type 2</td>
<td></td>
</tr>
<tr>
<td>Oracle JDBC Driver</td>
<td><code>URL</code>: for example, <code>jdbc:oracle:oci:@sample</code></td>
</tr>
<tr>
<td>Derby</td>
<td>• <code>databaseName</code>: path to the location of the database files. This directory must not exist already. For example, <code>c:\derbydata</code>.</td>
</tr>
<tr>
<td></td>
<td>• Clear the option <strong>Use this data source in container managed persistence (CMP)</strong>.</td>
</tr>
<tr>
<td></td>
<td>For more information, refer to the Derby documentation.</td>
</tr>
</tbody>
</table>

   **Note:** Decision Center may encounter various database dead-locks if the transaction isolation level is not set to **READ-COMMITTED**. Therefore, you might need to configure data source isolation to provide better concurrency experience. For more information see the following technote: [http://www-01.ibm.com/support/docview.wss?uid=swg21224492](http://www-01.ibm.com/support/docview.wss?uid=swg21224492)

6. In **Step 4**, select the default values in the list and click **Next**. **Step 5** displays a summary of your settings.

7. Click **Finish**.

   The connection pool is created and is associated with the data source.

8. Click **Save** to save the changes directly to the master configuration.

### Creating J2C authentication data

In WebSphere Application Server, you can secure your enterprise information system by creating J2C authentication data.
Before you begin

To create J2C authentication data, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 145.

About this task

After you have created your data source and connection pool, you optionally create the J2C authentication data. J2C is a secure mechanism for integrating enterprise information systems to an application server and enterprise applications.

This procedure uses rtsAdmin as the user ID and rtsAdmin as the password. Your user ID and password might be different, depending on the user or schema that is used to execute the Rule Execution Server SQL script. If you follow the procedure as described, tables are created under the rtsAdmin schema.

For a Derby database, if the connect command does not contain the user ID, the user ID and password are the default values APP and APP. If you create the tables under the APP schema, you do not need to do this procedure.

Procedure

1. In the side panel of the console, open Resources > JDBC and Data sources.
2. Click the name of your data source.
   Click the link, do not just select the check box.
3. Under Related Items, click JAAS - J2C authentication data.
4. On the Data sources page, click New and set the fields Alias, User ID, and Password. For example, this procedure uses the following values:
   • Alias: RtsDerbyUser
   • User ID: rtsAdmin
   • Password: rtsAdmin
5. Click Apply and then click Save to save directly to the master configuration.
6. Open Resources > JDBC and Data sources and click the name of your data source.
7. Under Security settings, for Component-managed authentication alias, select the <NodeName>/RtsDerbyUser alias, where <NodeName> is the name of the WebSphere Application Server node on which you are configuring Rule Execution Server.
8. For Container-managed authentication alias, select <NodeName>/RtsDerbyUser.
9. Click Apply and then click Save to save directly to the master configuration.

Changing the custom properties of your data source

Before you connect to the database, you can change the custom properties of your data source.

Before you begin

To change custom properties, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 145.
Procedure

To change the custom properties of your data source:
1. In the side panel of the console, open Resources > JDBC, and click Data sources.
2. Click the name of the data source that you want to customize.
3. Under Additional Properties, click Custom properties. From here, you can make custom changes to the data source.

Example

For example, for a Derby data source, if you have not created the database, you can set the createDatabase property to create. The first database connection creates the database.

To create the database:
1. Click createDatabase. The General Properties page opens.
2. Type create in the Value field.
3. Click OK, and then click Save to save the changes to the master configuration.

Connecting to the database

After you have created a data source and connection pool, you connect to the data source to test the connection to the database.

Procedure

To test the connection to your database:
1. In the Integrated Solutions Console, open Resources > JDBC, and click Data sources.
2. Select the check box next to the data source that you want to test, and click Test connection.
   The status of the connection is indicated at the top in a message such as the following one:
   The test connection operation for data source <data_source> on server <server_name> at node <node_name> was successful.

Step 3: Configuring security

Access to Decision Center is managed by the application server security. To access Decision Center in WebSphere Application Server, you must define a user registry. You can also manage the security policies.

Creating users and groups

Security relies on users and user groups. You define the groups to which the users belong in the application server. For this purpose, you configure a federated repository as a user registry.

Before you begin

To create users and groups, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in "Before you start: Opening the administration console" on page 145.
About this task

At this stage of the configuration, creating users and groups is optional:

- If you already have suitable groups and users defined, skip this procedure and connect users to their appropriate roles when you deploy your applications.
- If you do not have groups and users defined or if you want to define new groups and users, do the following steps.

Information about users and groups resides in a user registry. In WebSphere Application Server, a user registry authenticates a user and retrieves information about users and groups to perform security-related functions, including authentication and authorization. WebSphere Application Server supports the following types of user registries:

- Federated repository
- Local operating system
- Stand-alone Lightweight Directory Access Protocol (LDAP) registry
- Stand-alone custom registry

A federated user repository serves as the active user registry.

The following rules apply:

- Every user of Decision Center must belong to at least one of the mandatory groups rtsAdministrator, rtsConfigManager, rtsInstaller, or rtsUser. Membership in these groups determines the parts of Decision Center that a user can access.
- You must create all these groups in the application server.
- For testing purposes, you must create a default user and password for each of the mandatory groups.
- If you want to perform the Decision Center permissions tutorial in your own installation, you must create the custom groups Validator and Eligibility.

The following table summarizes the groups, their uses, and the default user names and passwords. For more information about Decision Center groups, see Groups.

<table>
<thead>
<tr>
<th>Group</th>
<th>Use</th>
<th>Default user name and password</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtsAdministrator</td>
<td>Mandatory, gives the user administrator access.</td>
<td>rtsAdmin, rtsAdmin</td>
</tr>
<tr>
<td>rtsConfigManager</td>
<td>Mandatory, gives the user configuration manager access.</td>
<td>rtsConfig, rtsConfig</td>
</tr>
<tr>
<td>rtsUser</td>
<td>Mandatory, gives a user standard access.</td>
<td>rtsUser1, rtsUser1</td>
</tr>
<tr>
<td>rtsInstaller</td>
<td>Mandatory, gives the user access to the Installation Manager.</td>
<td>rtsAdmin, rtsAdmin</td>
</tr>
<tr>
<td>Validator</td>
<td>Optional custom group, used in the Decision Center permissions tutorial.</td>
<td>Val, Val</td>
</tr>
<tr>
<td>Eligibility</td>
<td>Optional custom group, used in the Decision Center permissions tutorial.</td>
<td>Eli, Eli</td>
</tr>
</tbody>
</table>
Procedure

To configure a federated repository:

1. In the side panel of the console, open Security > Global security.
2. Configure the repository security as follows:
   - If Federated repositories is already selected under Current realm definition, make sure that Enable application security is selected. If you select Enable application security, you must click Apply and then click Save to save the changes to the master configuration.
   - If Federated repositories is not already selected, click Security Configuration Wizard and complete the wizard:
     a. In Step 1, select Enable application security and click Next
     b. In Step 2, select Federated repositories and click Next.
     c. In Step 3, type the name in the Primary administrative user name field, type websphere in the Password field, and click Next.
     d. In Step 4, review the security configuration summary and click Finish.
     e. Click Save to save the changes to the master configuration.
     f. Restart WebSphere Application Server, and then log in to the console as the primary administrative user.
3. In the side panel, open Users and Groups > Manage Groups.
4. Click Create, enter rtsUser as the group name, and click Create.
5. Click Create Like, create another group named rtsAdministrator, and click Create.
6. Click Create Like, create another group named rtsConfigManager, and click Create.
7. Click Create Like, create another group named rtsInstaller, and click Create.
8. Click Close.
   On the Manage Groups page, the table shows the groups that you have created.
9. If you want to follow the Decision Center permissions tutorial in your own installation, create two more groups: Validator and Eligibility.
10. In the side panel, open Users and Groups > Manage Users.
11. Click Create.
12. Enter rtsUser1 as the User ID and assign this new user to a group.
    a. Click Group Membership.
    b. On the Group Membership page, click Search.
    c. In the Available column, select the rtsUser group and click Add.
    d. Click Close.
13. Enter a given name and surname for rtsUser1, then enter the rtsUser1 password, and click Create.
14. Click Close.
15. Click Create Like and repeat steps 12 through 14 to create more users and assign them roles as show in the following table. The Val and Eli users are optional.
Table 33. Decision Center users and groups

<table>
<thead>
<tr>
<th>User ID and password</th>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtsUser1 - rtsUser1</td>
<td>rtsUser</td>
</tr>
<tr>
<td>rtsAdmin - rtsAdmin</td>
<td>rtsAdministrator, rtsInstaller</td>
</tr>
<tr>
<td>rtsConfig - rtsConfig</td>
<td>rtsConfigManager</td>
</tr>
<tr>
<td>Val - Val</td>
<td>rtsUser, Eligibility, Validator</td>
</tr>
<tr>
<td>Eli - Eli</td>
<td>rtsUser, Eligibility</td>
</tr>
</tbody>
</table>

Back on the Manager Users page, the table shows your users.

16. Restart your application server.

Updating the security policies
If you enable Java 2 security on WebSphere Application Server, you must override the global security policies of the application server so that the deployed application can access the MBean server.

If you enable Java 2 security in the Global Security window of the WebSphere Integrated Solutions Console, you must update the was.policy file that is packaged in the Decision Center EAR file.

The was.policy file is in the META-INF directory of the jrules-teamserver-WAS<version_number>.ear file, in the <ODM_InstallDir>/teamserver/applicationservers/WebSphere<version_number> folder.

Note: The EAR file is a compressed file. You must open it to extract the files that must be changed, and then replace the files in the EAR. You can use Ant commands to repackage the EAR file, as explained in “Repackaging the Decision Center archive” on page 174.

Update the was.policy file to give read and write permissions on each directory that contains published RuleDocs. For example:

permissions java.io.FilePermission "<path to my ruledoc folder>${/}-", "read, write, delete";

If you use Java 2 security but do not update the was.policy file, users cannot synchronize RuleDocs to the file system.

Setting custom properties for Decision Center security
To ensure that the Decision Center consoles work correctly, set the InvalidateOnUnauthorizedSessionRequestException and setContextRootForFormLogin properties in WebSphere Application Server.

Before you begin
To set custom properties for security, you work in the WebSphere Integrated Solutions Console. Make sure that you have a WebSphere Application Server profile start the server, and then open the console and log in by using the user ID and password that you defined in the profile.

About this task
To prevent potential session and authentication errors, set the following custom properties in the WebSphere Integrated Solutions Console.
<table>
<thead>
<tr>
<th>Location in the administrative console</th>
<th>Property and value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application servers &gt; server1 &gt; Session management &gt; Custom properties</td>
<td>InvalidateOnUnauthorized SessionRequestException = true</td>
<td>If the same user ID accesses the Business console and the Enterprise console at the same time, this property prevents authentication errors. For more information, see the following technote: <a href="http://www.ibm.com/support/docview.wss?uid=swg21609826">http://www.ibm.com/support/docview.wss?uid=swg21609826</a></td>
</tr>
<tr>
<td>Global Security &gt; Custom properties</td>
<td>com.ibm.websphere.security. setContextRootForFormLogin = true</td>
<td>If the Business console and the Enterprise console are used on the same application server, this property prevents a WebSphere Application Server cookie from pointing to the incorrect application. For more information, see the following technote: <a href="http://www.ibm.com/support/docview.wss?uid=swg1PM58885">http://www.ibm.com/support/docview.wss?uid=swg1PM58885</a></td>
</tr>
</tbody>
</table>

**Step 4: Deploying the Decision Center EAR file**

Before you deploy the Decision Center EAR file, you can declare custom groups, and after you deploy it, you must change the class loader sequence.

**Declaring custom groups**

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

**About this task**

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

You must add any custom groups that you declared in "Creating users and groups" on page 152. For example, add the Validator and Eligibility groups if you created them to follow the Decision Center tutorials.

You add custom groups by editing the deployment descriptor files in 
<ODM InstallDir>/teamserver/applicationservers/WebSphere<version_number>/jrules-teamserver-WAS<version_number>.ear.

**Tip:** To use the Decision Center permissions mechanism, you must upload groups to the database. For more information, see [Completing the configuration from the Decision Center Enterprise console](http://www.ibm.com/support/docview.wss?uid=swg21609826).

**Procedure**

To add your custom groups to the Decision Center deployment descriptor:

1. Before you modify the EAR file, create a back-up copy.
   - <ODM InstallDir>/teamserver/applicationservers/WebSphere8/jrules-teamserver-WAS8.ear
   - <ODM InstallDir>/teamserver/applicationservers/WebSphere85/jrules-teamserver-WAS85.ear
2. Uncompress the EAR file to extract the files that you must change.

3. In the SECURITY - ROLE section of the teamserver.war/WEB-INF/web.xml file, add each custom group as a role. For example:

   ```
   <security-role>
     <role-name>Validator</role-name>
   </security-role>
   ```

4. Similarly, add the groups in the decisioncenter.war/WEB-INF/web.xml file.

5. Add each custom group to the META-INF/application.xml file.

6. Repackage the EAR.

   You can use Ant commands, as explained in “Repackaging the Decision Center archive” on page 174.

**Deploying the EAR file**

After you have added custom groups, you deploy the Decision Center EAR file.

**Before you begin**

To deploy the EAR file, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 145.

**About this task**

After you have added your custom groups to the deployment descriptors in the Decision Center EAR, you deploy the EAR file on WebSphere Application Server to be able to map security roles to users and groups.

**Important:**

When you deploy the Decision Center EAR file, the process sets the persistence locale. After you have saved a rule to the database, you are no longer allowed to change the persistence locale. If you want to install Decision Center in a language other than English, take note of the instructions in Set the persistence locale.

If you redeploy the Decision Center EAR file, the redeployment has the following consequences:

- The class loading sequence is lost. See “Changing the class loading sequence” on page 159. WebSphere Application Server reverts to the default parent first setting.
- All users, such as rtsUser1 and rtsAdmin, lose their role, even though they belong to the correct group. When you sign in to the Decision Center console, a message is displayed, such as rtsUser1 does not have the correct role.

**Procedure**

To deploy the EAR file:

1. In the side panel of the console, click Applications > New Application and then New Enterprise Application.

2. Click Browse and navigate to the Decision Center EAR file for your version of WebSphere Application Server.

   ```
   <ODM_InstallDir>/teamservver/applicationservers/WebSphere8/jrules-teamservver-WAS8.ear
   ```
3. Click Next.

4. Select Detailed - Show all installation options and parameters.

5. Expand Choose to generate default bindings and mappings and select the Generate Default Bindings check box, then click Next.

6. If a security warning is displayed, click Continue.

7. On the Install New Application page, in Step 1, click Next to accept the default settings.

8. In Step 2, select the target server and the WAR files and click Next. Keep the default setting if you have only one server.

9. For Step 3 to Step 8, click Next to accept the default settings.

10. In Step 9, map security roles to users and groups.

   The application server uses the roles that are defined in the deployment descriptors. You must map these roles to the groups found in the security settings.

   a. Select a check box next to a role in the table and click Map groups.

   b. Click Search in the middle of the page to display the groups.

   c. Map the group to the role that you are editing by moving that role to the Selected column.

   d. Click OK and repeat Steps 10a to 10d for all the roles.

   e. In addition, if you want to follow the Decision Center permissions tutorial in your own installation, create two custom groups: Validator and Eligibility.

After you have completed the assignments, they are shown as in the following table. If you have groups or users, you can map them to the existing role instead.

Table 34. Mapping user groups to roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Mapped groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtsUser</td>
<td>rtsUser</td>
</tr>
<tr>
<td>rtsAdministrator</td>
<td>rtsAdministrator</td>
</tr>
<tr>
<td>rtsConfigManager</td>
<td>rtsConfigManager</td>
</tr>
<tr>
<td>rtsInstaller</td>
<td>rtsInstaller</td>
</tr>
<tr>
<td>Validator</td>
<td>Validator</td>
</tr>
<tr>
<td>Eligibility</td>
<td>Eligibility</td>
</tr>
</tbody>
</table>

11. If your application server is WebSphere Application Server V8.0, V8.5, or 8.5.5, click Next for Step 10 and Step 11 to accept the default settings.

Step 12 displays a summary.

**Note:** If WebSphere eXtreme Scale is also installed, more settings are displayed as Step 12 in the side panel of the WebSphere Integrated Solutions Console. For more information on administering WebSphere eXtreme Scale with WebSphere Application Server, see Administering in the WebSphere eXtreme Scale information center.

12. Click Finish.

13. Click Save to save your workspace changes to the master configuration.
Changing the class loading sequence
After you deploy the EAR file, you must set the class loading sequence to parent last.

Before you begin
To change the class loading sequence, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 145.

About this task
The Decision Center application does not support the default parent first configuration.

Procedure
To change the class loading sequence:
1. In the console, open Applications > Application Types > WebSphere enterprise applications.
2. Click ILOG Rule Team Server or the name you specified for the Decision Center application.
3. Under Modules, click Manage Modules.
4. Click teamserver.
5. In the Class loader order menu, select Classes loaded with local class loader first (parent last), and then click OK.
6. If you are configuring WebSphere Application Server V8.0, V8.5 or V8.5.5s, click decisioncenter and repeat the same action.
7. Click Save to save the changes directly to the master configuration.
8. In the side panel, open Applications > Application Types > WebSphere enterprise applications.
9. Select the check box next to ILOG Rule Team Server and click Start to start the application.

What to do next
Verify the deployment of Decision Center.

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 the configuration of the Decision Center consoles**

You complete the configuration either from the Decision Center console or by running Ant tasks.

To use the built-in governance mode, you must run an Ant task.

**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>
<thead>
<tr>
<th>Table 35. Actions in the Installation Settings wizard</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action</strong></td>
</tr>
<tr>
<td>Configure the database</td>
</tr>
<tr>
<td>Set up message files</td>
</tr>
<tr>
<td>Set up groups</td>
</tr>
<tr>
<td>Change the persistence locale</td>
</tr>
</tbody>
</table>
Table 35. Actions in the Installation Settings wizard (continued)

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change configuration parameters</td>
<td>Optional. You change some configuration parameters when you customize Decision Center. For more information, see Step 5: Set configuration parameters.</td>
</tr>
</tbody>
</table>

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/defaultextensionmessages_<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.
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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Used to</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>teamserver.&lt;extractorValidator&gt;.class</code></td>
<td>Specify a class of ruleset extractor validators to use for the extractorValidator 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.</td>
</tr>
</tbody>
</table>
| `teamserver.build.path` | Define the location of the IRL cache in the file system. Compute the path as follows:
  - Use this property with the name of the user who started the server as the root for the cache (`<build.path>_username`).
  - If this property is not defined, use the system property `java.io.tmpdir` and add rtscache. For example, `<temp dir>/rtscache_username`).
  - If the system property is not defined, use the server directory and add rtscache. For example, `<server dir>/rtscache_username`). |
| `teamserver.brl.verbalizers` | Specify the list of locales for which a BAL verbalizer is defined. |
| `teamserver.brl.verbalizer.<locale>` | Specify the verbalizer class for the locale. The class must implement the `IlrVerbalizer` 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 appropriate environment variables.

**Before you begin**

When preparing to run Ant tasks, make sure that the following conditions are met.

• 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.

To test your current version of Ant, type the following command in a Windows Command Prompt or UNIX shell:

```bash
ant -version
```

You can download Ant from [the Apache web site](#) or you can use the Ant 1.7.1 distribution that is packaged at `<ODM_InstallDir>/shared/tools/ant`, where `<ODM_InstallDir>` is your Operational Decision Manager installation directory.

• Make sure that any environment variables required by WebSphere Application Server are correctly set up. To do so, use the `setupCmdLine.bat` or `setupCmdLine.sh` script in the `<WAS_Install_Dir>/profiles/<profile_name>/bin` directory. This script sets up a full WebSphere Application Server environment, including the WAS_HOME, WAS_LOGGING, and WAS_CLASSPATH variables. For more information:

  - [WebSphere Application Server 8.0 information center](#)
  - [WebSphere Application Server 8.5 information center](#)

**About this task**

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

**Procedure**

To set up your environment to use Ant:
1. Set the ANT_HOME environment variable to `<ODM_InstallDir>/shared/tools/ant`.

2. Add the directory `<ODM_InstallDir>/shared/tools/ant/bin` to your PATH environment variable.

**Results**

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

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

**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.

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`.

**Example**

```
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 computer. 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. If these parameters are properly defined in this file, you do not have to include them in your Ant task command. 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=9080
server.url=${protocol}://${server.host}:${server.port}/teamserver

datasourceName=jdbc/ilogDataSource

outputFile=output.sql

languagePackPath = .
languagePackOutputPath = ./generated

defaultLocale =
selector =
branch =
override = false
```

---

166 IBM Operational Decision Manager: Configuring Operational Decision Manager on WebSphere Application Server
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).
- `[ -DdbSchemaName=<database schema name>]`
  You can use this optional parameter to specify the database schema name in which the Decision Center tables are stored. If you do not specify the parameter, Decision Center uses the database user name as the schema name. However, some databases allow some users to access several schemas, and the default schema is not always named the same as the user.
- `[ -DoutputFile=<SQL file>]`
  The name of the file that stores the generated SQL script. If this parameter is not specified, the task creates a file named `output.sql` in the directory that is defined as `basedir` in the `build.xml` file.

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

Results

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

Executing the database schema script:

You execute the database schema script.

Procedure

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

- `-Dserver.url=<server url>`
- `-DdatasourceName=<data source name>`
- `[ -Dfile=<SQL file>]`
  The name of the file to execute, which corresponds to the script that you...
created. If you do not specify this parameter, the task attempts to execute a file named output.sql in the directory that is defined as basedir in the build.xml file.

ant execute-schema -Dfile=my_sql_file.sql

**Uploading the database schema extension:**

You upload the database schema extension.

**Procedure**

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

- `Dserver.url=<server url>`
- `DdatasourceName=<data source name>`
- `DextensionModel=<model file>`
  - The model description (.brmx extension).
- `DextensionData=<data file>`
  - The model data description (.brdx extension).

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

**Results**

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

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

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

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

**Before you begin**

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

**About this task**

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

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

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

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

-DServer.url=<server url>
-DDatasourceName=<data source name>
-DDataSourcesName=<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:

```bash
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:

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

The default messages file is provided in: `<ODM_InstallDir>/teamserver/bin/defaultextensionmessages_<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>`

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

Configuring the users of the Decision Center Business console:

If you use the built-in governance mode, you can configure the users of the Business console by running an Ant task.

You must maintain the list of users to keep it consistent with the users that are defined in your application server.

The default way to configure the users is to add them to the Decision Center database by running the `upload-users` Ant command. The command uploads the names that are in an XML file.

Use the following format for the names:
For an example of a user file, see InstallDir/teamserver/tutorials/fileusers.xml.

Specify the following parameters for the upload-users command:
- -DuserFilePath=the path to the XML file
- -Dserver.url=the URL of the server
- -DrtsAdmin.login=the user name to log in to the console
- -DrtsAdmin.password=the password to log in to the console
- -DdatasourceName=the data source used

Example
ant upload-users -DuserFilePath=C:\my_path\my_users.xml -Dserver.url=my_url
- -DrtsAdmin.login=rtsAdmin -DrtsAdmin.password=rtsAdmin
- -DdatasourceName=my_datasource

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`.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>teamserver.&lt;extractorValidator&gt;.class</code></td>
<td>Specify a ruleset extractor validator class to use for the extractorValidator 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.</td>
</tr>
</tbody>
</table>
| `teamserver.build.path` | Define where the cache of the IRL is located on the file system. The path is computed as follows:

1. First, use this property with the name of the user who started the server as the root for the cache:
   `<build.path>_<username>`.
2. If the path is not defined, use the system property `java.io.tmpdir` and add `rtscache`. For example,
   `<temp_dir>/rtscache_<username>`.
3. If the system property is not defined, use the server directory and add `rtscache`. For example,
   `<server_dir>/rtscache_<username>`.
| `teamserver.brl.verbalizers` | Specify the list of locales for which a BAL verbalizer is defined. |
| `teamserver.brl.verbalizer.<locale>` | Specify the verbalizer class for the specified locale. The class must implement the `IlrVerbalizer` 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 repackage the Decision Center archive by using an Ant task.

When you add new .jar files to the Decision Center archive, you must repackage 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 repackage 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).
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 36. Search parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>search.SearchProvider</td>
<td>This parameter takes one of the following values:</td>
</tr>
<tr>
<td></td>
<td>• SolrEmbedded: Use this value to select the embedded Solr search engine.</td>
</tr>
<tr>
<td></td>
<td>• SolrRemote: Use this value to select a remote instance of the Solr search</td>
</tr>
<tr>
<td></td>
<td>engine.</td>
</tr>
<tr>
<td>search.SolrEmbeddedDataDir</td>
<td>Use this optional parameter to direct the index of the embedded version</td>
</tr>
<tr>
<td></td>
<td>(SolrEmbedded) to a specific directory on the Decision Center.</td>
</tr>
<tr>
<td>search.SolrRemoteUrl</td>
<td>Use this parameter with SolrRemote to provide the URL of the remote Solr</td>
</tr>
<tr>
<td></td>
<td>search engine.</td>
</tr>
</tbody>
</table>

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 37. Configuration table for search properties

<table>
<thead>
<tr>
<th>Solr server</th>
<th>Parameter settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>External server</td>
<td>Set the preferences.properties file as follows:</td>
</tr>
<tr>
<td></td>
<td>search.SearchProvider=SolrRemote</td>
</tr>
<tr>
<td>Embedded server</td>
<td>Set the preferences.properties file as follows:</td>
</tr>
<tr>
<td></td>
<td>search.SearchProvider=SolrEmbedded</td>
</tr>
<tr>
<td></td>
<td>search.SolrEmbeddedDataDir=c:/temp/DC-SearchIndex</td>
</tr>
</tbody>
</table>

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

Optionally, you can deploy, configure, and test Decision Validation Services to complement your Decision Center configuration on WebSphere Application Server.

Before you start

Before you can deploy and configure Decision Validation Services, you must install Rule Execution Server and its console. You go through a sequence of steps to ensure a successful configuration.
Before you deploy and configure Decision Validation Services, install the following software:

1. Optionally, install Rule Designer.
2. Install Rule Execution Server and Decision Center, and configure both applications.

**Note:** On WebSphere Application Server, Decision Validation Services uses the default Work manager `wm/default` to run simulations and tests in managed threads. Therefore, make sure that it is defined in the application server where you configure Decision Validation Services. For more information about configuring a work manager by using the WebSphere Application Server administration console, see the topic "Configuring work managers" in the WebSphere Application Server information center.

### What steps to follow

The following table summarizes the steps to configure Decision Validation Services.

**Note:** You can also perform the configuration by following the steps in Configuring Rule Execution Server on WebSphere Application Server for z/OS using wsadmin scripts.

<table>
<thead>
<tr>
<th>Step</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Step 1: Creating Decision Warehouse database resources”&lt;br&gt;Tip: If you have used the Rule Execution Server console to create database resources (see Populating a database using the Rule Execution Server console), this step is not necessary.</td>
<td>Optional</td>
</tr>
<tr>
<td>“Step 2: Deploying the Decision Validation Services archive for the first time” on page 179&lt;br&gt;This archive is the default SSP archive packaged with the installer.</td>
<td>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 which case the SSP archive does not have to be repackaged.</td>
</tr>
<tr>
<td>“Step 3: Checking the availability of Decision Validation Services” on page 180</td>
<td>Optional</td>
</tr>
<tr>
<td>“Step 4: Packaging Decision Validation Services by using Ant” on page 181</td>
<td>Optional</td>
</tr>
<tr>
<td>“Step 5: Redeploying the Decision Validation Services archive” on page 182</td>
<td>Checkmark</td>
</tr>
</tbody>
</table>

### 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:

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

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

  ```sql
  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:

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

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

  ```sql
  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:

<table>
<thead>
<tr>
<th>Database</th>
<th>Database tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM DB2</td>
<td>DB2 command line processor</td>
</tr>
<tr>
<td>Derby</td>
<td>ij command line processor</td>
</tr>
<tr>
<td>MySQL</td>
<td>mysql command line processor</td>
</tr>
<tr>
<td>Oracle</td>
<td>sqlplus command line processor</td>
</tr>
<tr>
<td>Postgre SQL</td>
<td>Postgre SQL command line tool</td>
</tr>
<tr>
<td>SQL Server</td>
<td>Query Tool</td>
</tr>
<tr>
<td>Sybase</td>
<td>isql command line processor</td>
</tr>
</tbody>
</table>

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

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

The default CLOB size might not be sufficient for the FULL_EXECUTION_TRACE field in the EXECUTION_TRACES table. You might need a size qualifier if SQL raises exceptions with the <lob-Value> reason code.

Step 2: Deploying the Decision Validation Services archive for the first time
After you have Decision Warehouse database resources available, you deploy the Decision Validation Services archive.

Before you begin
To deploy a Decision Validation Services archive, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 145.

About this task
You can follow this procedure to deploy the default Scenario Service Provider (SSP) archive that is packaged with the installer or for any subsequent deployment of a repackaged archive. Deploy the default SSP EAR file if you want to check the availability of the feature. You can also deploy the default SSP EAR file and use it to test your rules if you have an XML XOM. An XML XOM is included in the ruleset archive inside a RuleApp. Therefore, you do not have to repackage the SSP EAR file to include the XOM.

Important: You must deploy the Decision Validation Services archive on the same server as the XU. Moreover, for the testing and simulation services to work, by default the SSP application is expected to be collocated with the Rule Execution Server console. If a Rule Execution Server console instance is not also deployed on the same server as the SSP, you must implement the IlrSSPResolver interface as explained in “Distributing Decision Validation Services tests to multiple servers” on page 183.

If you are installing Decision Validation Services on z/OS, you can use customizable installation scripts to perform this task. For more information, refer to Configuring Rule Execution Server on WebSphere Application Server for z/OS using wsadmin scripts.

Procedure
To deploy an SSP archive:
1. In the side panel of the WebSphere Integrated Solutions Console, open Applications and click Websphere Enterprise Applications.
2. Click Install.
3. Make sure that **Local file system** is selected, click **Browse** and navigate to one of the following files, and then click **Next**.
   - For WebSphere Application Server 8.0: `<ODM_InstallDir>/executionserver/applicationservers/WebSphere8/jrules-ssp-WAS8.ear`
   - For WebSphere Application Server 8.5 or 8.5.5: `<ODM_InstallDir>/executionserver/applicationservers/WebSphere85/jrules-ssp-WAS85.ear`

4. Select the check box **Detailed - Show all installation options and parameters**.

5. Expand **Choose to generate default bindings and mappings**, elect the check box **Generate Default Bindings**, and click **Next**.

6. Click **Continue** to accept the security warning.

7. In **Step 1** through **Step 8** click **Next** to accept the default settings.

8. In **Step 9**, if you have activated security, select **resAdministrators** and click **Map groups**.

9. Click **Search**.

10. Click the **resAdministrators** group under **Available** and then click the arrows to move it to the **Selected** column.

11. Click **OK** to return to the **Map security roles to users or groups** page.

12. Repeat steps 8 to 11 for the **resDeployers** group, selecting **resDeployers**.

13. Click **Next** and then click **Finish**.

14. After the installation has completed click **Save** directly to the master configuration.

15. In the side panel, open **Applications > Application Types > WebSphere enterprise applications**, and click **jrules-ssp-WAS8** or **jrules-ssp-WAS85**.

16. Click **Manage Modules**.

17. Click **Scenario Service Provider**.

18. Under General Properties, for **Class loader order** select **Classes loaded with local class loader first (parent last)** and click **OK**.

19. Click **OK** again and then click **Save** to save directly to the master configuration.

20. In the side panel, open **Applications > Application Types > WebSphere enterprise applications**.

21. In the Enterprise Applications page, select the check box next to **jrules-ssp-WAS8** or **jrules-ssp-WAS85** and click **Start** to start the application.

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

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

To check the availability of Decision Validation Services:

1. Enter the URL `http://<host>:<port>/testing` in a web browser.

2. Log in to the (Scenario Service Provider) SSP application in one of the Rule Execution Server roles.
   - The application displays a home page, which contains information about the SSP server.

   **Version**
   - The version of Decision Server used.

   **Patch level**
The patch level of Decision Server used.

License information

The type of license of this version.

RuleSession

The rule session type (POJO or J2SE).

DAO Factory Class

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

Job store class

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

Job pool size

The size of the pool for asynchronous execution.

Started since

The time and date when the SSP started.

Jobs currently running

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

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

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

Step 4: Packaging Decision Validation Services by using Ant

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

About this task

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

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

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

**Step 5: Redeploying the Decision Validation Services archive**

How to redeploy the Decision Validation Services archive.

**Before you begin**

To redeploy a Decision Validation Services archive, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in **“Before you start: Opening the administration console” on page 145**.

**About this task**

You follow this procedure when you must redeploy the default Scenario Service Provider (SSP) archive that is packaged with the installer or any subsequent repackaged archive. Redeploy the default SSP EAR file if you want to check the availability of the feature. You can also redeploy the default SSP EAR file and use it to test your rules if you have an XML XOM. An XML XOM is included in the ruleset archive inside a RuleApp. Therefore, you do not have to repackage the SSP EAR file to include the XOM.

**Important:** The archive must be redeployed on the same server as the XU.

**Procedure**

To redeploy an SSP archive that was already deployed earlier:

1. In the side panel of the console, open **Applications** and click **Websphere Enterprise Applications**.
2. Select the deployed SSP application and click **Uninstall**.
3. Click **OK** to confirm uninstallation of the SSP application.
4. Click **Save** to save the changes directly to the master configuration.
5. Follow **“Step 2: Deploying the Decision Validation Services archive for the first time” on page 179** again to deploy the new SSP archive.
Distributing Decision Validation Services tests to multiple servers
When you configure the Rule Execution Server environment on a domain with multiple servers, you can define which Rule Execution Server instances are used to execute Decision Validation Services tests.

Before you begin
1. Package all your executable object models (XOMs) into the ssp.war archive. For more information, see Making the XOM accessible by repackaging the SSP.
2. Make sure that the Scenario Service Provider (SSP) archive file (ssp.war) is deployed on each server along with a XU connection factory and the appropriate data source definition.

About this task
To define which Rule Execution Server instances are used to execute Decision Validation Services tests, you create a custom SSP resolver.

Procedure
1. Implement the IlrSSPResolver interface.
   For a simple implementation, you can use the server name to return a specific server URL. For example, you can have two servers defined in Decision Center:
   • testingServer1: http://host1:9080/res
   • testingServer2: http://host1:9080/res
   And your SSP resolver can evaluate the SSP URL as follows:
   
   ```java
   if (serverName.equals("testingServer1")) {
       return (new URL("http://server1:9080/testing"));
   }
   else if (serverName.equals("testingServer2")) {
       return (new URL("http://server2:9080/testing"));
   }
   else
       return (new URL("http://host1:9080/testing"));
   ```
2. Add your class to the teamserver.war archive.
3. Set the teamserver.defaultSSPResolver.class property to that class.

Results
Next time you run a DVS test, the execution will be directed according to the server that you choose.

What to do next
For a better implementation, you can set a dependency with the project that is being tested so that you can distribute the DVS execution according to that project.

```java
ManagerBean bean = ManagerBean.getInstance();
IlrSession session = bean.getSession();
IlrRuleProject project = session.getWorkingBaseline().getProject();
String pname = project.getName();
```
Configuring the Decision Center consoles on WebSphere Application Server for Linux on System z

To be able to use the Decision Center consoles on an instance of WebSphere Application Server for Linux on System z, you set up the persistence for the resources, create a JDBC provider, deploy the provided archives for this server, and activate security.

Before you start

To configure Decision Center console on WebSphere Application Server for Linux on System z, you must follow specific steps. Some are mandatory, while others are optional.

A specific integration extension for the IBM Process Server platform is available. For more information, refer to this page: Integration SupportPacs

To understand the tasks you must complete, refer to “What steps to follow.”

You finish the configuration by Completing the configuration using the Decision Center console or Completing the configuration using Ant tasks

After your configuration is finished, Decision Center is ready to use (see Opening the Decision Center Enterprise console). The first time you open Decision Center, it does not contain a rule project. You must publish a project from Rule Designer (see Publishing a project).

This guide assumes that you are using a profile that has administrative security enabled.

Software Prerequisites

You must have installed the following software before you configure Decision Center:

- WebSphere Application Server for zLinux V8.0
- DB2 Version 9.5 or 10
- Java 1.6 included in your PATH statement.

You also need the following rights:

- Access to the WebSphere Application Server Integrated Solutions Console
- DB2 Administrator authority to CREATE a database and CREATE & UPDATE tables
- Authority to start and stop the WebSphere Application Server

What steps to follow

The following table summarizes the steps that you follow to configure Decision Center on WebSphere Application Server.

<table>
<thead>
<tr>
<th>Step</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1: Enforcing database user permissions</td>
<td>✔️</td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Database permission</th>
<th>Use Decision Center</th>
<th>Create the database schema by using the Decision Center console or Ant tasks</th>
<th>Modify the database schema by using the Decision Center console or Ant tasks</th>
<th>Migrate the database schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER TABLE</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>CREATE INDEX</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>CREATE ROLE</td>
<td>Not required</td>
<td>Not required</td>
<td>Not required</td>
<td>Required</td>
</tr>
<tr>
<td>CREATE SEQUENCE</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>CREATE TABLE</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>CREATE VIEW</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>DROP INDEX</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>DROP SEQUENCE</td>
<td>Not required</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>DROP TABLE</td>
<td>Not required</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>DROP VIEW</td>
<td>Not required</td>
<td>Not required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>INSERT TABLE</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>SELECT SEQUENCE</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>SELECT TABLE</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>UPDATE TABLE</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
</tbody>
</table>

**Step 2: Setting up Java 2 connection security**

How to create J2C authentication data in WebSphere Application Server.

**About this task**

If you completed this task as part of the Rule Execution Server installation, you do not need to do it a second time. The procedure is included here solely for completeness.

**Procedure**

To set up connection security:

1. From the WebSphere Application Server Integrated Solutions Console, open Security > Global Security.
2. Expand Java Authentication and Authorization Service and select J2C authentication data.

3. Click New and provide the DB2 Administrator’s credentials in the User ID and Password fields.

4. Click Apply and Save directly to the master configuration.

**Step 3: Configuring security on WebSphere Application Server**

Decision Center access is managed by the application server security. To access Decision Center in WebSphere Application Server, you must define a user registry.

**Defining a user registry**

How to define a file-based federated user repository as a user registry.

Information about users and groups resides in a user registry. In WebSphere Application Server, a user registry authenticates a user and retrieves information about users and groups to perform security-related functions, including authentication and authorization. WebSphere Application Server supports the following types of user registries:

- Federated repository
- Local operating system
- Standalone Lightweight Directory Access Protocol (LDAP) registry
- Standalone custom registry

A federated user repository serves as the active user registry.

- If you do not have groups or users defined or if you want to define new groups or users, go through the steps described in "Creating users and groups" on page 152. The following rules apply:
  - Every user of Decision Center must belong to at least one of the mandatory groups rtsAdministrator, rtsConfigManager, rtsInstaller, or rtsUser. Adherence to these groups determines the parts of Decision Center that a user can access.
  - You must create all these groups in the application server.
  - For testing purposes, also create a default user and password for each of these groups.
  - In addition, if you want to perform the Decision Center permissions tutorial in your own installation, you must create the two custom groups Validator and Eligibility.

- If you already have groups or users defined that can match the expected role, you will proceed to connect them to the roles during the application deployment.

The following table summarizes groups, their uses, and their default users and passwords.

<table>
<thead>
<tr>
<th><strong>Group</strong></th>
<th><strong>Use</strong></th>
<th><strong>Default user and password</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>rtsAdministrator</td>
<td>Mandatory, gives the user administrator access.</td>
<td>rtsAdmin, rtsAdmin</td>
</tr>
<tr>
<td>rtsConfigManager</td>
<td>Mandatory, gives the user configuration manager access.</td>
<td>rtsConfig, rtsConfig</td>
</tr>
<tr>
<td>rtsUser</td>
<td>Mandatory, gives a user standard access.</td>
<td>rtsUser1, rtsUser1</td>
</tr>
</tbody>
</table>
Creating users and groups
Security relies on users and user groups. You define the groups to which the users belong in the application server. For this purpose, you configure a federated repository as a user registry.

Before you begin
At this stage of the configuration, creating users and groups is optional:
- If you already have suitable groups and users defined, skip this procedure and connect users to their appropriate roles when you deploy your applications.
- If you do not have groups and users defined or if you want to define new groups and users, proceed with the following steps.

About this task
This step explains how to configure a federated user repository as the active user registry in the WebSphere Application Server Integrated Solutions Console.

Note: See Groups for information on the Decision Center groups.

Procedure
To configure a federated repository:
1. In the Integrated Solutions Console, open Security > Global security.
2. Configure the repository security as follows:
   - If Federated repositories is already selected under Current realm definition, make sure Enable application security is selected. If you select Enable application security, you must click Apply and Save to save the changes to the master configuration.
   - If it is not already selected, click Security Configuration Wizard. Complete the wizard as follows
     a. In Step 1, to specify the level of protection, select Enable application security and click Next.
     b. In Step 2, select Federated repositories and click Next.
     c. In Step 3, type the name in the Primary administrative user name field and websphere in the Password field, then click Next.
     d. In Step 4, review the security configuration summary and click Finish.
     e. Click Save to save the changes to the master configuration.
     f. Restart WebSphere Application Server. You then have to log into the Integrated Solutions Console as the primary administrative user.
3. In the side panel, open **Users and Groups** > **Manage Groups**, and click **Create**.
4. Enter *rtsUser* as the group name and click **Create**.
5. Click **Create Like** and create another group named *rtsAdministrator* and click **Create**.
6. Click **Create Like** and create another group named *rtsConfigManager* and click **Create**.
7. Click **Create Like** again and enter another group named *rtsInstaller* and click **Create**, then click **Close**.
8. If you want to perform the Decision Center permissions tutorial in your own installation, create two additional groups: **Validator** and **Eligibility**.
9. In the side panel, open **Users and Groups** > **Manage Users**, and click **Create**.
10. Enter *rtsUser1* as the User ID with a password *rtsUser1*. Also specify the given name and last name.
11. Click **Group Membership** and then click **Search**. Select the *rtsUser* group and click **Add**. Click **Close** and then click **Create** and **Close** again.
12. Create another user named *rtsAdmin* with password *rtsAdmin*. Repeat the previous step and assign the user to the *rtsAdministrator* and *rtsInstaller* groups.
   Then create a user named *rtsConfig* with password *rtsConfig*. Repeat the previous step and assign the user to *rtsConfigManager* group.
13. Skip this step if you do not intend to perform the Decision Center permissions tutorial. Create a user named *Val* with password *Val* and add the user to the *rtsUser*, **Validator**, and **Eligibility** groups. Then create another user named *Eli* with password *Eli* and add the user to the *rtsUser* and **Eligibility** groups.
14. Restart your application server.

**Security policies for the Decision Center console**

When you enable Java 2 Security on WebSphere Application Server, you must override the server policies.

After the global security of WebSphere Application Server is activated, the MBean server cannot be accessed from the deployed application.

These security policies must be overridden for the Decision Center console because the console needs to record and manage a set of MBeans. Decision Center is packaged with a specific policy file that overrides the server policies.

This was.policy file is packaged in the EAR files, in the META-INF directory.

When you enable Java 2 Security on WebSphere Application Server, you must update the was.policy file to give read and write permissions on each directory where RuleDoc is published, as in the following example:

```java
permission java.io.FilePermission "<path to my ruledoc folder>${/}-", "read, write, delete";
```

If you do not update the was.policy file, users cannot synchronize RuleDoc to the file system.
Setting custom properties for Decision Center security
To ensure that the Decision Center consoles work correctly, set the
InvalidateOnUnauthorizedSessionRequestException and
setContextRootForFormLogin properties in WebSphere Application Server.

Before you begin
To set custom properties for security, you work in the WebSphere Integrated
Solutions Console. Make sure that you have a WebSphere Application Server
profile start the server, and then open the console and log in by using the user ID
and password that you defined in the profile.

About this task
To prevent potential session and authentication errors, set the following custom
properties in the WebSphere Integrated Solutions Console.

<table>
<thead>
<tr>
<th>Location in the administrative console</th>
<th>Property and value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application servers &gt; server1 &gt;</td>
<td>InvalidateOnUnauthorizedSessionRequestException = true</td>
<td></td>
</tr>
<tr>
<td>Session management &gt; Custom properties</td>
<td></td>
<td>If the same user ID accesses the Business console and the Enterprise console at the same time, this property prevents authentication errors. For more information, see the following technote: <a href="http://www.ibm.com/support/docview.wss?uid=swg21609826">http://www.ibm.com/support/docview.wss?uid=swg21609826</a>.</td>
</tr>
<tr>
<td>Global Security &gt; Custom properties</td>
<td>com.ibm.websphere.security.setContextRootForFormLogin = true</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the Business console and the Enterprise console are used on the same application server, this property prevents a WebSphere Application Server cookie from pointing to the incorrect application. For more information, see the following technote: <a href="http://www.ibm.com/support/docview.wss?uid=swg1PM58885">http://www.ibm.com/support/docview.wss?uid=swg1PM58885</a>.</td>
</tr>
</tbody>
</table>

Step 4: Creating the DB2 database
An empty DB2 database is required for Decision Center.

About this task
You must create an empty database for Decision Center.

Procedure
To create the DB2 database:
1. If the Decision Center database does not exist, you must create it. You need
   DB2 Administrator permission to create the database using the following commands:
   ```
   su {DB2AdminName}
   db2
   CREATE DATABASE {RTSDBName}
   ```
2. After you have created the empty database, you can close the DB2 command line client:
   ```
   quit
   exit
   ```

Creating the DB2 database on z/OS

Before you begin

Before you perform this step, check that you have DB2 for z/OS Version 9.1 or 10 installed, and that you have DB2 Administrator authority to CREATE a database and CREATE & UPDATE tables.

About this task

Note:

You must customize the JCL as required, as indicated in the comments within each PDSE member, before you run them on the z/OS system. You can customize the JCL either by using a manual search and replace, or by using an ISPF macro.

After you have customized the JCL variables for your configuration, you can submit the jobs to create a database for Decision Center.

Procedure

On the z/OS computer where DB2 is installed, run the following JCL scripts to create the database tables and grant authority to the tables. The database administrator typically performs this task.

1. `InstallDir/teamserver/jcl/HBRDCCDB.jcl`
2. `InstallDir/teamserver/jcl/HBRDCGNR.jcl`
   ```
   ++DB2USER++ is the user ID to access DB2.
   ```

Step 5: Creating a JDBC provider

How to create a JDBC provider in WebSphere Application Server to enable Decision Center.

About this task

The first step in creating a data source is to create a JDBC provider.

Procedure

1. Log in to the Integrated Solutions Console.
2. In the side menu, open Resources > JDBC and click JDBC Providers.
3. In Scope, select Node=xxx, Server=yyy, where xxx is the name of your node and yyy the name of your server, and click New.
4. In Step 1 select the database type, provider type and implementation type:
   - Database type
     - DB2
   - Provider type
     - DB2 Universal JDBC Driver Provider
Implementation type

Select, then type a name (for example, “DB2 Universal JDBC Driver Provider for Decision Center”).

5. Click Next.
6. In Step 2, leave all class path values, including the blanks, as the default values and click Next. A summary is provided in Step 3.
7. Check that the class path to the JAR of your driver and the implementation class are correct.
   Default values are usually sufficient.
8. Click Finish and Save to save the changes directly to the master configuration.
9. The DB2 installation might have installed the file db2jcc_license_cisuz.jar in a location that is not in the WebSphere path. Use the following command to find the db2jcc_license_cisuz.jar file:
   find / -name db2jcc_license_cisuz.jar
   Note: It is best to do this as the root user to avoid permissions messages.
10. From the WebSphere Integrated Solutions Console, open Environment > WebSphere variables and click DB2UNIVERSAL_JDBC_DRIVER_PATH.
11. Set the value to the path discovered by the find command.
12. Click OK and Save to save the changes to the master configuration.

Step 6: Creating a data source and connection pool

Creating a data source assumes that your database is already running.

Creating a data source and connection pool
You must create a connection pool and a data source in WebSphere Application Server to enable Decision Center.

About this task

After you have created your JDBC provider, you can create a data source and connection pool.

Note:

If WebSphere Application Server is used in cluster mode, you must define the data source at node level in the cluster (as opposed to cluster level).

Procedure

To create a data source and a connection pool:
1. In the Integrated Solutions Console, open Resources > JDBC and Data sources.
2. In the Scope side panel, select the scope that you selected for the JDBC provider in “Step 5: Creating a JDBC provider” on page 191, and then click New.
3. In Step 1, enter a name for the data source in the Data source name field and jdbc/<your_data_source_name> in the JNDI name field, and click Next.
   Important: The name of the data source is not important here, but in the JNDI name, you must reuse the data source name that you entered. For example, if
you entered ilogDataSource as the data source name, set the JNDI name to jdbc/ilogDataSource. Otherwise, Decision Center cannot use the data source.

4. In **Step 2** choose **Select an existing JDBC provider**, then select the JDBC provider that you created in "Step 5: Creating a JDBC provider" on page 191 and click **Next**.

5. In **Step 3**, set the specific database properties for the data source:
   a. Enter the following values:
      
      **Driver type**
      
      4

      **Database name**
      
      Type your database name, for example RTSDB.

      **Server name**
      
      Type your server name.

      **Port number**
      
      Type your port number.

      **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'. For more information see the following technote: [http://www-01.ibm.com/support/docview.wss?uid=swg21224492](http://www-01.ibm.com/support/docview.wss?uid=swg21224492)

   b. Clear the option **Use this data source in container managed persistence (CMP)**.

   c. Click **Next**.

6. In **Step 4**, select the DB2 Administrator’s authentication alias from the **Component-managed authentication alias** option, and then click **Next**.

   In **Step 5**, a summary of the data source is provided.

7. Click **Finish** and **Save** to save the changes to the master configuration.

**Testing the connection to the database**

How to test the connection to your database.

**About this task**

After you have created a data source and connection pool, and set the custom properties, test the connection to your database.

**Procedure**

To test the connection to your database:

1. In the Integrated Solutions Console, open **Resources** > **JDBC** and click **Data sources**.

2. Select the check box next to the data source you want to test and click **Test connection**.

   The status of the connection is indicated at the top of the screen.

   Potential connection errors might be:

   - **Java class ..com.ibm.db2.jcc is not found**
     
     Check that the jcc drivers classpath is correctly set.
Step 7: Deploying the Decision Center EAR on WebSphere Application Server

Deploying the Decision Center EAR on WebSphere Application Server V8.0 requires you to declare custom groups, deploy the EAR, then change the Class Loader sequence.

Declaring custom groups
Before you can deploy the EAR file, you must declare your custom groups in the deployment descriptor.

About this task
The Decision Center EAR file references the basic groups rtsUser, rtsConfigManager, rtsAdministrator, and rtsInstaller.

However, you must add any custom groups that you declared previously, as described in “Creating users and groups” on page 188. Also, add the Validator and Eligibility groups that you created for the Decision Center tutorials. You add custom groups by editing the deployment descriptor files in the EAR file 

\[<++JRULESHOME++>\]jrules-teamserver-WAS<version_number>.ear\] before deploying, where \[<++JRULESHOME++>\] is your z/OS zFS JRules installation directory.

Tip:
- You might want to make a copy of the EAR file before you modify it.
- To use the Decision Center permissions mechanism, you upload groups to the database. For more information, see Completing the configuration using the Decision Center console.

Procedure
To add your custom groups to the Decision Center deployment descriptor:

1. Add your custom group as a role in the SECURITY - ROLE section of \[teamserver.war/WEB-INF/web.xml\] of \[jrules-teamserver-WAS<version_number>.ear\] in the EAR file. For example:

   <security-role>
   <role-name>my_custom_group</role-name>
   </security-role>

   ...

2. Repeat the same action in the \[decisioncenter.war/WEB-INF/web.xml\] file.

3. Add your custom group to \[META-INF/application.xml\] in the EAR file.

   ...

   <security-role>
   <role-name>my_custom_group</role-name>
   </security-role>

Changing the class loading sequence
After you have deployed the EAR file, you must set the class loading sequence to parent last.
About this task

After you have deployed the Decision Center EAR file, set the class loading sequence. The Decision Center application does not support the default parent first configuration.

Procedure

To change the class loading sequence:

1. In the panel, open Applications > Application Types > WebSphere enterprise applications.
2. Click ILOG Decision Center or the name that you specified for the Decision Center application.
3. In the Modules section, click Manage Modules.
4. Click teamserver, select Class loader order > Classes loaded with local class loader first (parent last), then click OK.
5. If you are configuring on WebSphere Application Server V8.0, click decisioncenter and repeat the same action.
6. Click Save to save the changes directly to the master configuration.
7. In the panel, open Applications > Application Types > WebSphere enterprise applications.
8. Select the check box next to ILOG Decision Center and click Start to start the application.

What to do next

You can now verify the deployment of Decision Center. See Step 5: Verifying the deployment of the Decision Center console.

Deploying the EAR

How to deploy the Decision Center EAR on WebSphere Application Server.

About this task

After you have added your custom groups to the deployment descriptors in the Decision Center EAR, you deploy the EAR on WebSphere Application Server.

Important:

- When you deploy the Decision Center EAR, the process sets the persistence locale. After you save a rule to the database, you no longer need to change the persistence locale. If you want to install Decision Center in a language other than English, take note of the instructions provided in Step 4: Set Persistence Locale.
- If you redeploy the Decision Center EAR, the redeployment has the following consequences:
  - The class loading sequence is lost (see Changing the class loading sequence on page 194). Decision Center reverts to the default parent first setting.
  - All users (such as rtsUser1, rtsAdmin) lose their role, even though they belong to the correct group. You notice this when you sign in to the Decision Center console. A message is displayed, for example, rtsUser1 does not have the correct role.
**Procedure**

To deploy the EAR:

1. In the side panel of the WebSphere Integrated Solutions Console, click **Applications > New Application** and then **New Enterprise Application**.

2. Click **Browse** and navigate to the following Decision Center EAR file, and then click **Next**.
   
   
   
   `<++JRULESHOME++>/jrules-teamserver-WAS8.ear`

   `<++JRULESHOME++>` is your z/OS zFS Operational Decision Manager installation directory.

3. Select the **Detailed - Show all installation options and parameters** check box.

4. Expand **Choose to generate default bindings and mappings** and select the check box **Generate Default Bindings**, and then click **Next**.

5. Click **Continue** to accept the security warning.

6. In **Step 1**, click **Next** to accept the default settings.

7. In **Step 2**, select the target server and click **Next**.

   Keep the default setting if you only have one server.

8. For **Step 3** to **Step 8**, click **Next** to accept the default settings.

9. Click **Step 9** to map security roles to users and groups.

   The application server uses the roles that are defined in the deployment descriptors. You must map these to the groups found in the security settings.

   a. Select a check box next to a role in the table and click **Map groups**.

   b. Click **Search** in the middle of the page to display the groups.

   c. Map the group to the role that you are editing by moving it to the **Selected** column.

   d. Click **OK** and repeat for all the roles.

   After you have completed the assignments, they are shown as in the following table. If you have groups or users, you can map those to the existing role instead.

<table>
<thead>
<tr>
<th>Role</th>
<th>Special subjects</th>
<th>Mapped groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtsUser</td>
<td>None</td>
<td>rtsUser</td>
</tr>
<tr>
<td>rtsAdministrator</td>
<td>None</td>
<td>rtsAdministrator</td>
</tr>
<tr>
<td>rtsConfigManager</td>
<td>None</td>
<td>rtsConfigManager</td>
</tr>
<tr>
<td>rtsInstaller</td>
<td>None</td>
<td>rtsInstaller</td>
</tr>
<tr>
<td>Validator</td>
<td>None</td>
<td>Validator</td>
</tr>
<tr>
<td>Eligibility</td>
<td>None</td>
<td>Eligibility</td>
</tr>
</tbody>
</table>

10. Click **Next**.

    A summary is displayed.

11. Click **Finish**.

12. After the installation has completed, click **Save** to save your workspace changes to the master configuration.

**Verifying the deployment of the Decision Center Enterprise console**

After you have deployed Decision Center on your application server, verify the deployment.
Procedure

To verify the deployment of Decision Center:

1. Enter the default URL to access Decision Center in a browser.
   - The default URL to access Decision Center is http://localhost:<PORT_NUMBER>/teamserver.
   - On WebSphere Application Server V8.0, each console has a different default URL:
     - For the Enterprise Console: http://localhost:<PORT_NUMBER>/teamserver
     - For the Business Console: http://localhost:<PORT_NUMBER>/decisioncenter
   - If your browser is not running on the same host as the application server, replace localhost with the address of the host.
   - If your web application is mapped to a host on a port that is different from the port number shown, change the port number to your host port number.

2. Select the help option to view the help.

Step 8: Completing the configuration

Explains how to complete the configuration. You can perform this step using the Decision Center console or 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 38. Actions in the Installation Settings wizard

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the database</td>
<td>Mandatory when you complete the configuration with a database on a distributed platform. For more information, see Step 1: Configure the database.</td>
</tr>
</tbody>
</table>
Table 38. Actions in the Installation Settings wizard (continued)

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set up message files</td>
<td>Mandatory during the installation only if you have some custom rule model extension files. For more information, see Step 2: Set up message files.</td>
</tr>
<tr>
<td>Set up groups</td>
<td>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.</td>
</tr>
<tr>
<td>Change the persistence locale</td>
<td>Mandatory if the persistence locale is different from the locale en_US. For more information, see Step 4: Set the persistence locale.</td>
</tr>
<tr>
<td>Change configuration parameters</td>
<td>Optional. You change some configuration parameters when you customize Decision Center. For more information, see Step 5: Set configuration parameters.</td>
</tr>
</tbody>
</table>

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](https://www.ibm.com/support/pages/ibm-operational-decision-manager).

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](https://www.ibm.com/support/pages/ibm-operational-decision-manager) 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/ defaultextensionmessages_<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:

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

**Step 3: Set up groups:**

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

**Before you begin**

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

About this task

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

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

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

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

Procedure

To set up groups:
1. Click New.
2. Type the group name.
3. Click Apply.
4. Repeat steps 1 on page 163 to 3 on page 163 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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Used to</th>
</tr>
</thead>
<tbody>
<tr>
<td>teamserver.&lt;extractorValidator&gt;.class</td>
<td>Specify a class of ruleset extractor validators to use for the extractorValidator 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.</td>
</tr>
</tbody>
</table>
Parameter | Used to
---|---
teamserver.build.path | Define the location of the IRL cache in the file system. Compute the path as follows:
- Use this property with the name of the user who started the server as the root for the cache (<build.path>_<username>).
- If this property is not defined, use the system property java.io.tmpdir and add rtscache. For example, <temp dir>/rtscache_<username>.
- If the system property is not defined, use the server directory and add rtscache. For example, <server dir>/rtscache_<username>.
teamserver.brl.verbalizers | Specify the list of locales for which a BAL verbalizer is defined.
teamserver.brl.verbalizer.<locale> | Specify the verbalizer class for the locale. The class must implement the ilog.rules.vocabulary.verbalization.IlrVerbalizer 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 appropriate environment variables.

**Before you begin**
When preparing to run Ant tasks, make sure that the following conditions are met:
- 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.
To test your current version of Ant, type the following command in a Windows Command Prompt or UNIX shell:

```
ant -version
```

You can download Ant from [the Apache web site](http://ant.apache.org) or you can use the Ant 1.7.1 distribution that is packaged at `<ODM_InstallDir>/shared/tools/ant`, where `<ODM_InstallDir>` is your Operational Decision Manager installation directory.

- Make sure that any environment variables required by WebSphere Application Server are correctly set up. To do so, use the `setupCmdLine.bat` or `setupCmdLine.sh` script in the `<WAS_Install_Dir>/profiles/<profile_name>/bin` directory. This script sets up a full WebSphere Application Server environment, including the `WAS_HOME`, `WAS_LOGGING`, and `WAS_CLASSPATH` variables.
  - WebSphere Application Server 8.0 information center
  - WebSphere Application Server 8.5 information center

**About this task**

Communication between the Ant tasks and Decision Center supports the HTTP or HTTPS communication protocols. For more information, see [Ant task communication protocol](http://ant.apache.org/manual/)

**Procedure**

To set up your environment to use Ant:

1. Set the `ANT_HOME` environment variable to `<ODM_InstallDir>/shared/tools/ant`.
2. Add the directory `<ODM_InstallDir>/shared/tools/ant/bin` to your `PATH` environment variable.

**Results**

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

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

**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.

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`.

**Example**

```
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 computer. 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. If these parameters are properly defined in this file, you do not have to include them in your Ant task command. The content of the `teamserver-anttasks.properties` file is as follows:

```properties
# Default properties
# ------------------------------------
rtsAdmin.login=rtsAdmin
rtsAdmin.password=rtsAdmin
protocol=http
server.host=localhost
server.port=9080
server.url=${protocol}://${server.host}:${server.port}/teamserver
datasourceName=jdbc/ilogDataSource
outputFile=output.sql
languagePackPath = .
languagePackOutputPath = ./generated

Creating the database schema:

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

Creating schemas with the set-extensions Ant task:

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

About this task

Extensions to the Decision Center rule model are stored in two XML files.
- Model description: This file usually has the file name extension `.brmx`.
- Initialization of enumerations and hierarchies: This file usually has file name extension `.brdx`.

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

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

Alternatively, you can create the database schema step by step, which is useful if you want to look at the generated SQL schema. See "Creating the schema using a step-by-step sequence" on page 167.

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>
-ددatasourceName=<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 EligibilityValidator"

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>
-ددatasourceName=<data source name>
-DextensionModel=<model file>
  The model description (.brmx extension).
-DextensionData=<data file>
  The model data description (.brdx extension).
[-DdbName=<database schema name>]
  You can use this optional parameter to specify the database schema name in which the Decision Center tables are stored. If you do not specify the parameter, Decision Center uses the database user name as the schema name. However, some databases allow some users to access several schemas, and the default schema is not always named the same as the user.
[-DoutputFile=<SQL file>]
  The name of the file that stores the generated SQL script. If this parameter is not specified, the task creates a file named output.sql in the directory that is defined as basedir in the build.xml file.

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

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

Executing the database schema script:

You execute the database schema script.

Procedure

To execute the SQL script that you created, run the `execute-schema` Ant task with these parameters:
- `-Dserver.url=<server url>`
- `-DdatasourceName=<data source name>`
- `-Dfile=<SQL file>`

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

```bash
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).

```bash
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:

```
ant upload-roles -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:

```
ant gen-drop-schema -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/defaultheadertitlesmessages_<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>
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.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>teamserver.&lt;extractorValidator&gt;.class</td>
<td>Specify a ruleset extractor validator class to use for the extractorValidator name. The class must implement the IlrExtractorValidator interface. After this class is defined, specify this name as the extractor validator to use when defining a ruleset extractor.</td>
</tr>
</tbody>
</table>
| teamserver.build.path                       | Define where the cache of the IRL is located on the file system. The path is computed as follows:  
1. First, use this property with the name of the user who started the server as the root for the cache: `<build.path>_<username>`.  
2. If the path is not defined, use the system property `java.io.tmpdir` and add rtscache. For example, `<temp_dir>/rtscache_<username>`.  
3. If the system property is not defined, use the server directory and add rtscache. For example, `<server_dir>/rtscache_<username>`. |
| teamserver.brl.verbalizers                  | Specify the list of locales for which a BAL verbalizer is defined.                                                                                                                                                                                                   |
| teamserver.brl.verbalizer.<locale>          | Specify the verbalizer class for the specified locale. The class must implement the IlrVerbalizer 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>
• [-Uusername]
• -Dkey=<parameter key>

Repackaging the Decision Center archive:

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

When you add new .jar files to the Decision Center archive, you must repackage 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 repackage 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” on page 175.

You can set three parameters for the search engine:

Table 39. Search parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>search.SearchProvider</td>
<td>This parameter takes one of the following values:</td>
</tr>
<tr>
<td></td>
<td>• SolrEmbedded: Use this value to select the embedded Solr search engine.</td>
</tr>
<tr>
<td></td>
<td>• SolrRemote: Use this value to select a remote instance of the Solr search engine.</td>
</tr>
<tr>
<td>search.SolrEmbeddedDataDir</td>
<td>Use this optional parameter to direct the index of the embedded version (SolrEmbedded) to a specific directory on the Decision Center.</td>
</tr>
<tr>
<td>search.SolrRemoteUrl</td>
<td>Use this parameter with SolrRemote to provide the URL of the remote Solr search engine.</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Solr server</th>
<th>Parameter settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>External server</td>
<td>Set the preferences.properties file as follows:</td>
</tr>
<tr>
<td></td>
<td>search.SearchProvider=SolrRemote</td>
</tr>
<tr>
<td>Embedded server</td>
<td>Set the preferences.properties file as follows:</td>
</tr>
<tr>
<td></td>
<td>search.SearchProvider=SolrEmbedded</td>
</tr>
<tr>
<td></td>
<td>search.SolrEmbeddedDataDir=c:/temp/DC-SearchIndex</td>
</tr>
</tbody>
</table>

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

Optionally, you can deploy, configure, and test Decision Validation Services to complement your Decision Center configuration on WebSphere Application Server.

Before you start

Before you can deploy and configure Decision Validation Services, you must install Rule Execution Server and its console. You go through a sequence of steps to ensure a successful configuration.

Before you deploy and configure Decision Validation Services, install the following software:

1. Optionally, install Rule Designer.
2. Install Rule Execution Server and Decision Center, and configure both applications.

Note: On WebSphere Application Server, Decision Validation Services uses the default Work manager wm/default to run simulations and tests in managed threads. Therefore, make sure that it is defined in the application server where you configure Decision Validation Services. For more information about configuring a work manager by using the WebSphere Application Server administration console, see the topic "Configuring work managers" in the WebSphere Application Server information center.

What steps to follow

The following table summarizes the steps to configure Decision Validation Services.
Note: You can also perform the configuration by following the steps in Configuring Rule Execution Server on WebSphere Application Server for z/OS using wsadmin scripts.

<table>
<thead>
<tr>
<th>Step</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Step 1: Creating Decision Warehouse database resources” on page 177</td>
<td>Optional</td>
</tr>
<tr>
<td>Tip:</td>
<td></td>
</tr>
<tr>
<td>If you have used the Rule Execution Server console to create database resources (see Populating a database using the Rule Execution Server console), this step is not necessary.</td>
<td></td>
</tr>
<tr>
<td>“Step 2: Deploying the Decision Validation Services archive for the first time” on page 179</td>
<td>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 which case the SSP archive does not have to be repackaged.</td>
</tr>
<tr>
<td>This archive is the default SSP archive packaged with the installer.</td>
<td></td>
</tr>
<tr>
<td>“Step 3: Checking the availability of Decision Validation Services” on page 180</td>
<td>Optional</td>
</tr>
<tr>
<td>“Step 4: Packaging Decision Validation Services by using Ant” on page 181</td>
<td>Optional</td>
</tr>
<tr>
<td>“Step 5: Redeploying the Decision Validation Services archive” on page 182</td>
<td></td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Database</th>
<th>Database tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM DB2</td>
<td>DB2 command line processor</td>
</tr>
<tr>
<td>Derby</td>
<td>ij command line processor</td>
</tr>
<tr>
<td>MySQL</td>
<td>mysql command line processor</td>
</tr>
<tr>
<td>Oracle</td>
<td>sqlplus command line processor</td>
</tr>
<tr>
<td>Postgre SQL</td>
<td>Postgre SQL command line tool</td>
</tr>
<tr>
<td>SQL Server</td>
<td>Query Tool</td>
</tr>
<tr>
<td>Sybase</td>
<td>isql command line processor</td>
</tr>
</tbody>
</table>

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

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

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

The default CLOB size might not be sufficient for the FULL_EXECUTION_TRACE field in the EXECUTION_TRACES table. You might need a size qualifier if SQL raises exceptions with the `<Lob-Value>` reason code.
Step 2: Deploying the Decision Validation Services archive for the first time

After you have Decision Warehouse database resources available, you deploy the Decision Validation Services archive.

Before you begin

To deploy a Decision Validation Services archive, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 145.

About this task

You can follow this procedure to deploy the default Scenario Service Provider (SSP) archive that is packaged with the installer or for any subsequent deployment of a repackaged archive. Deploy the default SSP EAR file if you want to check the availability of the feature. You can also deploy the default SSP EAR file and use it to test your rules if you have an XML XOM. An XML XOM is included in the ruleset archive inside a RuleApp. Therefore, you do not have to repackage the SSP EAR file to include the XOM.

Important: You must deploy the Decision Validation Services archive on the same server as the XU. Moreover, for the testing and simulation services to work, by default the SSP application is expected to be collocated with the Rule Execution Server console. If a Rule Execution Server console instance is not also deployed on the same server as the SSP, you must implement the IlrSSPResolver interface as explained in “Distributing Decision Validation Services tests to multiple servers” on page 183.

If you are installing Decision Validation Services on z/OS, you can use customizable installation scripts to perform this task. For more information, refer to Configuring Rule Execution Server on WebSphere Application Server for z/OS using wsadmin scripts.

Procedure

To deploy an SSP archive:

1. In the side panel of the WebSphere Integrated Solutions Console, open Applications and click Websphere Enterprise Applications.
2. Click Install.
3. Make sure that Local file system is selected, click Browse and navigate to one of the following files, and then click Next.
   - For WebSphere Application Server 8.0: <ODM_InstallDir>/executionserver/applicationservers/WebSphere8/jrules-ssp-WAS8.ear
   - For WebSphere Application Server 8.5 or 8.5.5: <ODM_InstallDir>/executionserver/applicationservers/WebSphere85/jrules-ssp-WAS85.ear
4. Select the check box Detailed - Show all installation options and parameters.
5. Expand Choose to generate default bindings and mappings, elect the check box Generate Default Bindings, and click Next.
6. Click Continue to accept the security warning.
7. In Step 1 through Step 8 click Next to accept the default settings.
8. In Step 9, if you have activated security, select resAdministrators and click Map groups.
9. Click Search.
10. Click the resAdministrators group under Available and then click the arrows to move it to the Selected column.
11. Click OK to return to the Map security roles to users or groups page.
12. Repeat steps 8 on page 180 to 11 on page 180 for the resDeployers group, selecting resDeployers.
13. Click Next and then click Finish.
14. After the installation has completed click Save directly to the master configuration.
15. In the side panel, open Applications > Application Types > WebSphere enterprise applications, and click jrules-ssp-WAS8 or jrules-ssp-WAS85.
16. Click Manage Modules.
17. Click Scenario Service Provider.
18. Under General Properties, for Class loader order select Classes loaded with local class loader first (parent last) and click OK.
19. Click OK again and then click Save to save directly to the master configuration.
20. In the side panel, open Applications > Application Types > WebSphere enterprise applications.
21. In the Enterprise Applications page, select the check box next to jrules-ssp-WAS8 or jrules-ssp-WAS85 and click Start to start the application.

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

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

To check the availability of Decision Validation Services:
1. Enter the URL http://<host>:<port>/testing in a web browser.
2. Log in to the (Scenario Service Provider) SSP application in one of the Rule Execution Server roles.
   The application displays a home page, which contains information about the SSP server.

**Version**

The version of Decision Server used.

**Patch level**

The patch level of Decision Server used.

**License information**

The type of license of this version.

**RuleSession**

The rule session type (POJO or J2SE).

**DAO Factory Class**

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

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

**Job pool size**

The size of the pool for asynchronous execution.

**Started since**

The time and date when the SSP started.

**Jobs currently running**

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

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

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

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

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

**About this task**

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

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

**Procedure**

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

Step 5: Redeploying the Decision Validation Services archive

How to redeploy the Decision Validation Services archive.

Before you begin

To redeploy a Decision Validation Services archive, you work in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 145.

About this task

You follow this procedure when you must redeploy the default Scenario Service Provider (SSP) archive that is packaged with the installer or any subsequent repackaged archive. Redeploy the default SSP EAR file if you want to check the availability of the feature. You can also redeploy the default SSP EAR file and use it to test your rules if you have an XML XOM. An XML XOM is included in the ruleset archive inside a RuleApp. Therefore, you do not have to repackgage the SSP EAR file to include the XOM.

Important: The archive must be redeployed on the same server as the XU.

Procedure

To redeploy an SSP archive that was already deployed earlier:
1. In the side panel of the console, open Applications and click Websphere Enterprise Applications.
2. Select the deployed SSP application and click Uninstall.
3. Click OK to confirm uninstallation of the SSP application.
4. Click Save to save the changes directly to the master configuration.
5. Follow “Step 2: Deploying the Decision Validation Services archive for the first time” on page 179 again to deploy the new SSP archive.
Chapter 5. 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:


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.
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
access rights
  for MBean descriptors  70
Ant tasks
  configuring Decision Center  165, 202
  configuring users  171
  creating the database schema  168, 205
defining and uploading message files  171, 208
executing the database schema  168, 206
gen-create-schema  168, 205
gen-drop-schema  170, 207
print-config-param  173, 209
remove-config-param  173, 209
removing a database schema  170, 207
repackage-war  174, 211
repackaging the Decision Center archive  174, 211
set-config-param  173, 209
setting configuration parameters  173, 209
setting the persistence locale  172, 209
setting up the environment  165, 202
ssp-setup  181, 218
upload-extensions  169, 206
upload-messages  171, 208
upload-roles  169, 207
upload-users  171
uploading groups to the database  169, 207
uploading the database schema extension  169, 206
asynchronous execution
  pool size  180, 217
  support in Decision Server  83

C
cells
  Rule Execution Server configuration in a cluster environment  90
WebSphere Application Server clusters  5
changing, class loading sequence
  Decision Center  195
class loading sequence, changing
  Decision Center  159, 195
custom groups, declaring
  Decision Center  156, 194
custom properties
  Decision Center security  155, 190
setting for data sources on WebSphere Application Server  164
configuring
  data source  191
  creating a JDBC provider  191
executing the database schema  168, 206
figuration of Decision Center
  declaring custom groups  156
  defining user registry  187
on WebSphere Application Server overview  184
using federated repository  188
configuring Decision Centers
  on WebSphere Application Server overview  145
configuring Decision Validation Services  181, 218
diagnostics  91
evironments  90
on WebSphere Application Server  55
creating connection pools  62
creating data sources  62
deploying the hosted transparent decision service EAR file  81
federated repository  67
J2C authentication data  64
JDBC providers  61
mapping the admin group to the Monitor role  69
security  66
WebSphere Application Server security policies  70
connection factories
  for WebSphere MQ input and output queues  83
  for WebSphere MQ input and output topics  85
current pool
  creating  149, 192
connection pools
  Rule Execution Server  144
  WebSphere Application Server  62
custom groups, declaring
  Decision Center  156, 194
custom properties
  Decision Center security  155, 190
setting for data sources on WebSphere Application Server  64
data sources
  on WebSphere Application Server
    creating  62
  setting custom properties on
    WebSphere Application Server  64
database
  connecting  193
testing connection  152
database resources
  for Decision Validation Services  177, 214
database schema
  creating  168, 205
  executing  168, 206
  for Rule Execution Server
    choosing, using Installation Settings wizard  77
    reviewing, using Installation Settings wizard  77
  removing  170, 207
  uploading extension  169, 206
databases
  creating schema by running SQL scripts  78
  from the Installation Settings wizard  75
permissions
  Decision Center  147, 185
testing the connection
  WebSphere Application Server  66, 80
user permissions  58
DB2
  creating a database for Rule Execution Server configuration  59
  creating a database on z/OS  148
  creating JDBC providers
    for WebSphere Application Server  61
DB2 database
  creating a database schema by running SQL scripts  79
Decision Center
  configuring the database  161, 198
  configuring with Ant tasks  165, 202
  Installation Settings wizard  160, 197
  opening  222
  reprockaging the archive  174, 211
  security
    custom properties  155, 190
    setting up configuration parameters  164, 201
    setting up groups  162, 199
    setting up message files  162, 199
    setting up persistence locale  163, 200
    verifying deployment  159
  WebSphere Application Server clusters removing  31
Decision Center console
  configuring
    completing the configuration  160
IBM Operational Decision Manager: Configuring Operational Decision Manager on WebSphere Application Server

Decision Server Rules
WebSphere Application Server clusters
removing 19
Decision Validation Services
checking availability 180, 217
configuring
on WebSphere Application Server 176, 213
deploying the archive for the first time 179, 216
distributing tests to multiple servers 183
packaging using Ant 181, 218
redeploying 182, 219
Decision Warehouse
perspective setting for execution trace 58
declaring
custom groups in Decision Center 156
declaring, custom groups
Decision Center 194
defining, user registry
Decision Center 152, 187
deploying
Decision Validation Services
archive 179, 216
hosted transparent decision service EAR file
on WebSphere Application Server 81
jrules-teamserver-WAS8.ear 195
management EAR
on WebSphere Application Server 73
XU RAR
on WebSphere Application Server 71
deploying, Decision Center EAR
Decision Center 157, 195
Derby database
creating a database schema
by running SQL scripts 78
Derby databases 59
creating JDBC providers
for WebSphere Application Server 61
diagnostics
in the Rule Execution Server console running 91
dropping a database schema
See removing

E
ear, deploying
Decision Center 157, 195
environment variables
for Ant 165, 202
environments
Rule Execution Server
configuration 90
execution unit (XU)
deploying the XU RAR 71

F
failover
class configuration 1
gold topology 5
federated repositories
Rule Execution Server
on WebSphere Application Server 67
def federated repository, using
Decision Center 152, 188
installing Decision Center (continued)
deploying the Decision Center EAR 157, 195
restricting user permissions 147, 185
security policies 155, 189
using federated repository 152
installing Rule Execution Server
on WebSphere Application Server
testing the database connection 66, 80

G
gold topology
on WebSphere Application Server 5
groups
Rule Execution Server
on WebSphere Application Server 67
setting up 162, 199
uploading to the database 169, 207

H
hosted transparent decision services
deploying EAR
on WebSphere Application Server 81

I
input/output queues
creating the connection factory in
WebSphere MQ 83
input/output queues, WebSphere MQ activation specification 87
input/output topics, WebSphere MQ activation specification 87
creating the connection factory 85
Installation Settings wizard 160, 197
choosing the database schema 77
Configure Database 161, 198
report 78
reviewing the database schema 77
Rule Execution Server on WebSphere Application Server 55
Set Configuration Parameters 164, 201
Set Persistence Locale 163, 200
Set up Groups 162, 199
Set up Message File 162, 199
to create a Rule Execution Server database schema 75
installing
creating to database 193
creating a connection pool 149, 192
creating a data source 149, 192
creating a JDBC provider 149
database, connecting to 193
database, testing connection 152
installing, Decision Center class loading sequence, changing 159, 195
declaring custom groups 194
defining user registry 152

J
J2C authentication data
Rule Execution Server
on WebSphere Application Server 64
JBoss application server
verifying deployment of Decision Center
159
JDBC provider,
creating 191
JDBC provider, creating 149
JDBC providers
WebSphere Application Server
DB2 61
JMX (Java Management Extensions)
limitations with Rule Execution Server console instances 5
jrules-teamserver-WAS8.ear
deploying 195

L
Last Participant Support (LPS)
enabling in WebSphere Application Server 89
Linux on System z
configuring Rule Execution Server on WebSphere Application Server 55

M
management EAR, deploying
on WebSphere Application Server 73
mapping
the admin group to the Monitor role,
in WebSphere Application Server 69
MBeans
and WebSphere MQ 83
descriptors for WebSphere Application Server 7.0 70
security policy for the Rule Execution Server console 70
message files
defining and uploading 171, 208
setting up 162, 199
message-driven rule beans
and WebSphere MQ installation 88
queue activation specification 87
topic activation specification 87
MQ
Integrating WebSphere MQ in
WebSphere Application Server 83
O
opening
Decision Center  222
Rule Execution Server console 76

P
packaging
Decision Validation Services 181, 218
parameters
See configuration parameters
database access 58
Decision Center installation 147, 185
for MBean descriptors 70
persistence
and Installation Settings wizard 75
configuring in Rule Execution
Server 58
persistence locale
setting 172, 209
setting up 163, 200
Profile Management Tool 13, 14
Profile Management Tools  16, 25, 26, 29
profile templates
configure WebSphere Application
Server 32
to configure Decision Center 20
to configure WebSphere Application
Server 8
properties
custom, setting 64
publishing rule projects
from Decision Center 221

Q
queues, WebSphere MQ
for request messages 84
for response messages 85

R
redeploying
Decision Validation Services 182, 219
removing
database schema 170, 207
roles
in WebSphere Application Server 69
Rule Execution Server
configuring for WebSphere
Application Server for Linux on
System z  55
configuring for WebSphere
Application Server on distributed
platforms 55
configuring on Rule Execution Server
setting custom properties 64
configuring on WebSphere Application
Server
deploying the management
EAR 73
empty database schema and
credentials 59
diagnostics 91
Rule Execution Server (continued)
distributing DVS tests to multiple
instances 183
on WebSphere Application Server
deploying the XU RAR 71
setting up the persistence mode 58
Rule Execution Server configuration
creating a database schema
by running SQL scripts 78
from the Installation Settings
wizard 75
Rule Execution Server console
none or more than one
JMX limitations 5
opening 76
security policies
WebSphere Application Server 70
rule projects
publishing
from Decision Center 221
rule sessions
and WebSphere MQ integration 83

S
Scenario Service Provider (SSP)
application collocated with the Rule
Execution Server console 179, 216
Rule Execution Server console
collocation with the SSP
application 179, 216
ssp-setup Ant task 181, 218
security
custom properties 155, 190
Decision Center 155, 190
enabled when configuring Decision
Center 145, 184
J2C authentication data 64
policies for the Rule Execution Server
console
WebSphere Application Server 70
Rule Execution Server
on WebSphere Application
Server 66
security policies
Decision Center 155, 189
SQL scripts
to create a database schema 78
to remove a database schema 170, 207
ssp-setup, Ant task 181, 218

T
topics, WebSphere MQ
for request messages 86
for response messages 86
transaction management
in WebSphere Application Server, Last
Participant Support 89

U
user permissions 58
user registry, defining
Decision Center 152, 187
users
Rule Execution Server
on WebSphere Application
Server 67

W
WebLogic Server application server
verifying deployment of Decision
Center 159
WebSphere Application Server
cluster configuration 13, 14, 16, 25,
26, 29
removing applications 18, 30
clusters, configuring 1, 13, 14, 16, 18,
25, 26, 29, 30
configuring Rule Execution Server
database, empty 59
security, Rule Execution Server 66
setting up the Ant tasks
environment 165, 202
WebSphere MQ
input queue, creating 84
input topic, creating 86
integrating in WebSphere Application
Server 83
Last Participant Support, enabling 89
message-driven rule bean,
installing 88
output queue, creating 85
output topic, creating 86
queue activation specification,
creating 87
queue connection factory, creating 83
topic activation specification,
creating 87
topic connection factory, creating 85
workload balancing
cluster configuration 1
gold topology 5

X
XU RAR, deploying
on WebSphere Application Server 71
XUConnectionFactory
default JNDI name 71, 90

Z
z/OS
Decision Center consoles
creating a database 148