

IBM Operational Decision Manager
Version 8 Release 6

*Configuring Operational Decision
Manager on WebSphere Application
Server*

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 173.

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

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

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

Contents

Chapter 1. Clustering servers for high availability and scalability 1

WebSphere Application Server clustering and the ideal topology	1
Operational Decision Manager cell	4
Decision Server ideal topology cell	5
Operational Decision Manager profile templates	7
Configuring Rule Execution Server by using profile templates	8
Augmenting a deployment manager profile with Rule Execution Server	8
Running the configureDSCluster script to create a cluster	9
Running the addNodeToDSCluster script	12
Unaugmenting Rule Execution Server from existing profiles	14
Uninstalling Rule Execution Server from a cluster	14
Configuring the Decision Center consoles by using profile templates.	16
Augmenting a deployment manager profile with Decision Center	16
Running the configureDCCluster script to create a cluster	17
Running the addNodeToDCCluster script	19
Unaugmenting Decision Center from an existing cluster	21
Uninstalling Decision Center from a cluster.	22
Configuring the event runtime by using profile templates	23
Augmenting an existing WebSphere Application Server management profile	23
Augmenting an existing management profile by using the Profile Management Tool	24
Augmenting an existing management profile by using the manageprofiles command	25
Unaugmenting an existing profile by using the manageprofiles command	28
Augmenting an existing WebSphere Application Server custom profile	28
Augmenting an existing custom profile by using the Profile Management Tool	29
Augmenting an existing custom profile by using the manageprofiles command	29
Unaugmenting an existing profile by using the manageprofiles command	30
Creating profiles for an ideal topology cluster	31
Configuring a cluster by using the ideal topology	33
Creating and configuring a cluster that uses WebSphere MQ as the messaging provider	37
Configuring WebSphere MQ for use as a messaging provider.	39
Creating profiles for a cluster that uses WebSphere MQ as the messaging provider	39
Configuring a cluster that uses WebSphere MQ as the messaging provider	41

Defining a highly available collection of catalog servers	43
---	----

Chapter 2. Configuring Rule Execution Server on WebSphere Application Server 45

Before you start: Opening the administration console	45
Step 1: Selecting and applying the persistence type	48
Step 2: Restricting database user permissions	48
Step 3: Setting up your database	49
Creating a DB2 database for Linux on System z	50
Creating a DB2 database on z/OS	50
Step 4: Setting up a data source and connection pool	51
Creating a JDBC provider	51
Creating a data source and connection pool	52
Creating J2C authentication data	53
Setting custom properties.	54
Testing the connection to the database	55
Step 5: Activating security on WebSphere Application Server	56
Introduction to WebSphere Application Server security.	56
Creating users and groups	57
Mapping user groups to the Monitor role	59
Security policies for the Rule Execution Server console	60
Step 6: Deploying the Rule Execution Server MBean descriptors	60
Step 7: Deploying the XU RAR	61
Step 8: Deploying the Rule Execution Server management EAR file	63
Step 9: Creating a Rule Execution Server database schema	65
Creating a database schema using the Rule Execution Server console	65
Installation Settings wizard overview.	65
Opening the Rule Execution Server console.	66
Step 1: Welcome to the Installation Settings wizard	66
Step 2: Choose the database schema	67
Step 3: Review the database schema	67
Step 4: Using the Installation Settings wizard report	68
Creating the database schema by running SQL scripts	68
Running SQL scripts to create a DB2 schema	69
Testing the connection to the database	70
Step 10: Deploying the hosted transparent decision service EAR file	71
Step 11: Setting the DecodeUrlAsUTF8 custom property	72
Optional configuration steps.	72
Repackaging the Rule Execution Server archive using Ant	72

Integrating WebSphere MQ in WebSphere Application Server to support asynchronous execution	73
WebSphere MQ integration overview	73
Creating the WebSphere MQ queue connection factory	74
Creating the WebSphere MQ input queue	75
Creating the WebSphere MQ output queue	75
Creating a topic connection factory	76
Creating the WebSphere MQ input topic	76
Creating the WebSphere MQ output topic	77
Creating the WebSphere MQ queue activation specification	77
Creating the WebSphere MQ topic activation specification	78
Installing the message-driven rule bean	79
Enabling server-wide Last Participant Support	80
Configuring Rule Execution Server in different environments	80
Verifying the deployment and configuration	81

Chapter 3. Configuring Decision Server Events on WebSphere Application Server

Server	83
Planning your Decision Server Events environment	85
Creating a Decision Server Events environment	85
Choosing a Decision Server Events topology	86
Customizing the event runtime	86
Editing the setenv files	87
Customizing the event runtime environment	88
Configuring the event runtime database for Decision Server Events	88
Creating the event runtime database	88
Manually creating event runtime tables	89
Deploying tables in separate databases	90
Deploying the event runtime in a case-sensitive database manager	93
Changing the event runtime messaging provider	93
Configuring WebSphere Application Server default messaging to be the JMS provider	93
Configuring WebSphere MQ to be the JMS provider	95
Configuring WebSphere MQ	95
Configuring Decision Server Events	95
Setting properties	97
Configuring properties by using the WebSphere Application Server administrative console	98
Configuring properties by using the wsadmin scripting tool	98
Configuring properties by using the Properties tool	99
Configuring how step data is stored	101
Configuring the event runtime to store step data in memory	101
Configuring the event runtime to store context data in ObjectGrid	102
Configuring the event runtime to record history	102
Configuring the durable event queue	103

Configuring the log size of the durable event queue	104
Configuring how time-delayed event rules and actions are handled	104
Configuring multiple Decision Server Events profiles to coexist	105
Configuring technology connectors on a separate computer	106
Configuring the User Console connector to use a WebSphere Application Server data source	106
Tuning your event runtime configuration	108
Tuning with the event runtime and WebSphere Application Server	109
Tuning with JMS messaging	110
Verifying your Decision Server Events configuration	111
Securing Decision Server Events	115
User authentication	115
Configuring user authentication by using WebSphere Application Server	116
Configuring user authentication by using LDAP	117
Configuring user authentication by using Microsoft Windows Active Directory	119
Configuring user authentication by using User Console	120
Securing the service integration bus	121

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

Before you start: Opening the administration console	125
Step 1: Restricting database user permissions	127
Step 2: Configuring database connectivity	128
Creating the DB2 database	128
Creating the DB2 database on z/OS	129
Creating a JDBC provider	129
Creating a data source and a connection pool	130
Creating J2C authentication data	131
Changing the custom properties of your data source	132
Testing the connection to the database	133
Step 3: Configuring security	133
User registry	134
Creating users and groups	135
Security policies	136
Setting custom properties for Decision Center security	137
Step 4: Deploying the Decision Center EAR file	137
Declaring custom groups	137
Deploying the EAR file	138
Changing module properties	140
Step 5: Verifying the deployment of the Decision Center consoles	141
Step 6: Completing the configuration of the Decision Center consoles	141
Completing the configuration from the Decision Center Enterprise console	142
Installation Settings wizard overview	142

Step 1: Configure the database	143
Step 2: Set up message files	143
Step 3: Set up groups.	144
Step 4: Set the persistence locale	145
Step 5: Set configuration parameters.	145
Completing the configuration by using Ant tasks	146
Setting up the Ant tasks environment	147
Creating the database schema	148
Creating schemas with the set-extensions Ant task	148
Creating the schema using a step-by-step sequence	149
Removing a database schema	151
Defining and uploading message files	152
Configuring the users of the Decision Center Business console	153
Setting the persistence locale	153
Adding or removing configuration parameters	154
Repackaging the Decision Center archive	156
Configuring the search function of the Decision Center Business console	156
Using a remote Solr search engine	156
Setting parameters for the Solr search engine	157
Additional steps to configure testing and simulation	158

Before you start	158
Step 1: Creating Decision Warehouse database resources	159
Step 2: Creating Decision Runner database resources	161
Step 3: Deploying the testing and simulation archive	161
Step 4: Checking the availability of the testing and simulation services	163
Step 5: Using an Ant task to package the SSP archive	164
Step 6: Redeploying a testing and simulation archive	165
Distributing rule testing to multiple servers	166
Tuning the Decision Runner web application	167

Chapter 5. Verifying your configuration of Decision Center . . . 169

Publishing a project to Decision Center.	169
Opening the Decision Center consoles	170

Notices	173
Trademarks	175

Index	177
------------------------	------------

Chapter 1. Clustering 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 ideal topology

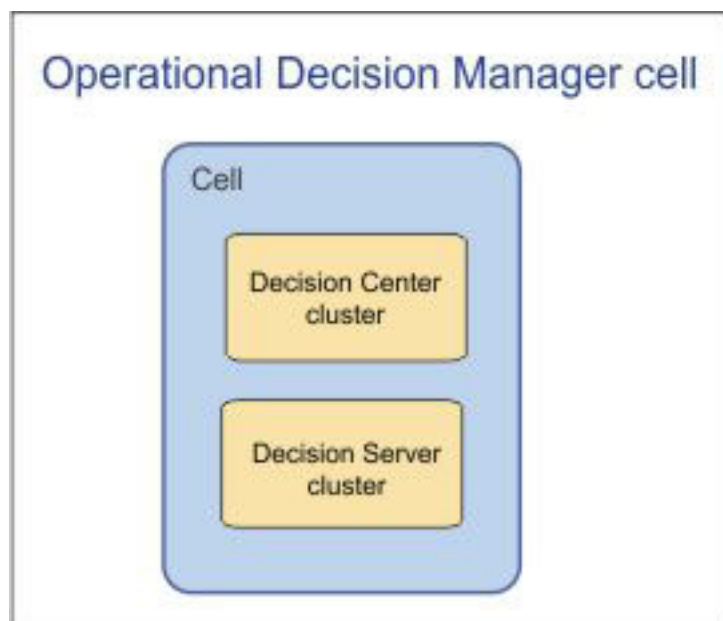
Planning a clustered topology with WebSphere Application Server Network Deployment is complicated because many combinations are possible but only a subset suits your needs. An ideal topology helps you to make the right choices.

An *ideal 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 ideal 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.

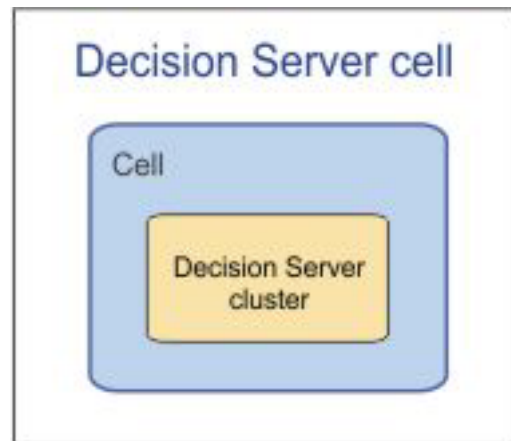
Operational Decision Manager cell

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, a staged environment, or a mix of both.

In practice, your cluster does not have to replicate exactly the ideal topology. You can implement cell management centralization (sharing) or take a more isolated approach (staging); whatever suits your needs. For example, in a shared environment, you can still choose to separate the production environments for authoring and for execution.

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.

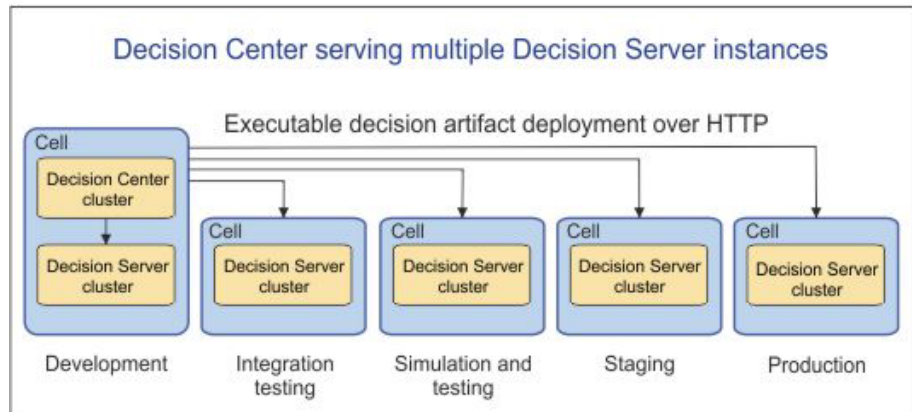
A shared environment has the following advantages:

- 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:

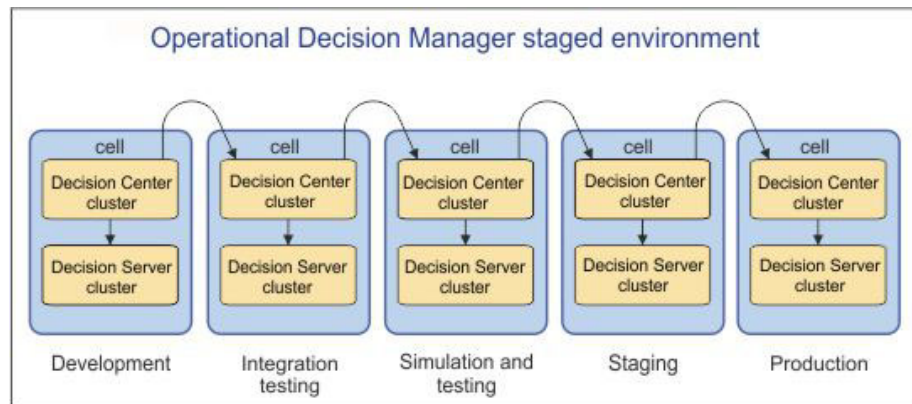
- Provides full isolation between development lifecycle stages.
- Isolates authoring and execution by stage and cell.
- Offers the ability to customize Decision Center in each cell, including security.
- Supports high availability for clustered Decision Center and Decision Server instances.

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.



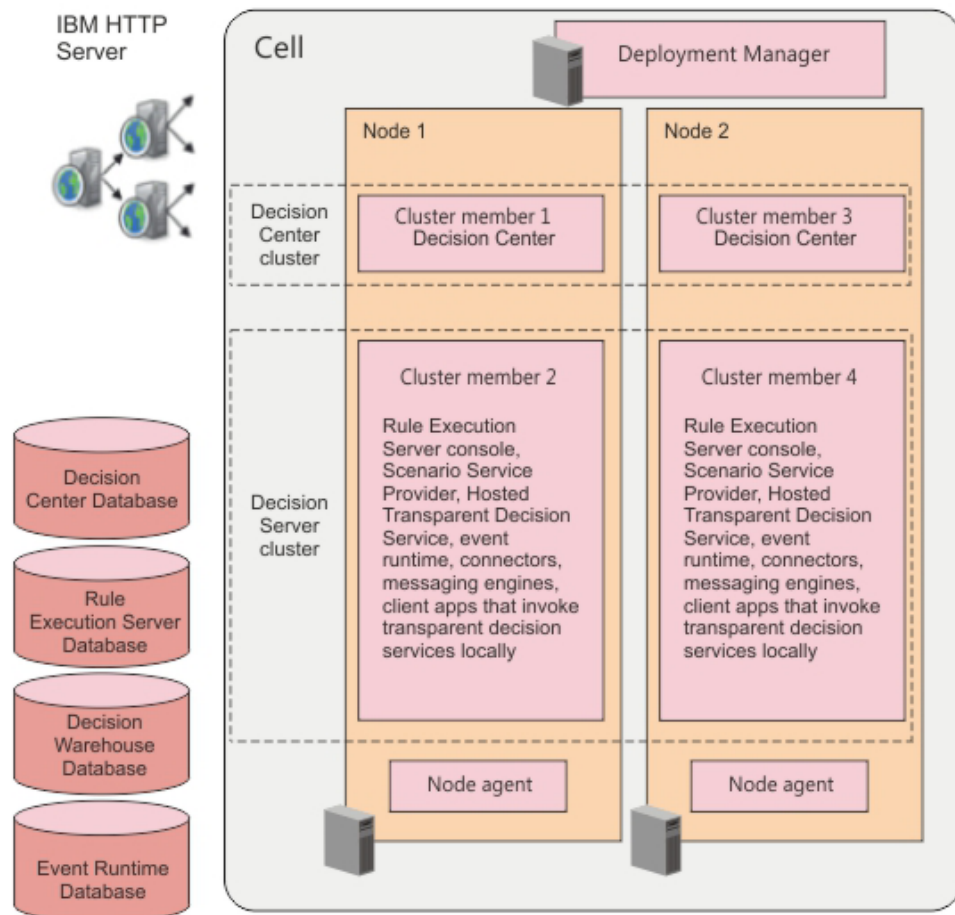
You can configure the Operational Decision Manager cell and the Decision Server cell by using the provided profile template scripts, or manually by following the documented steps.

Operational Decision Manager cell

The Operational Decision Manager ideal 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 to the cluster.
- 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).

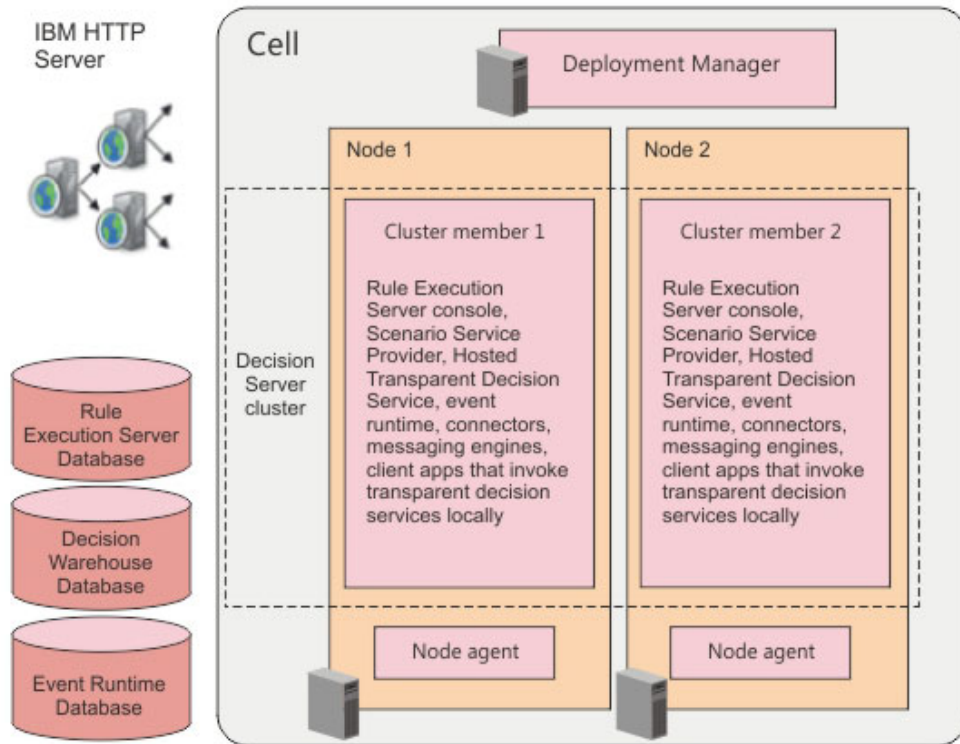
Rule and event runtimes are collocated in the same JVMs. To enable a decision service that is triggered by an event detection, you must place the two engines in the same JVM for a local invocation.

Decision Server ideal topology cell

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

Typically, you use the Decision Server cell to host your 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.



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 to the cluster.
- 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.

Known limitations in clustered environments

No Rule Execution Server console deployed in the cluster

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.

One Rule Execution Server console deployed in the cluster

For the testing and simulation services to work, you must point to a single Scenario Service Provider (SSP) server. Set the **LOCAL_DEPLOYMENT_ONLY** parameter in the `web.xml` file of the SSP server to true for a single Rule Execution Server and to false for a clustered Rule Execution Server.

More than one Rule Execution Server console deployed in the cluster

- Some administrative interactions from Rule Execution Server components to the Rule Execution Server console might not work properly. There is no notification between each MBean of the Rule Execution Server console. A change in a Rule Execution Server console can remain invisible until you log off and log on again. You can also force a read from the persistence layer, by clicking the **Update RuleApps** icon.
- Avoid the use of management sessions (`ilog.rules.session.IlrManagementSession` API) or interceptors (`IlrSessionInterceptor` implementations), which might fail on an error such as `java.lang.IllegalStateException` or `IlrSessionInterceptorException`.
- For the testing and simulation services to work, you must point to a single Scenario Service Provider (SSP) server. Set the **LOCAL_DEPLOYMENT_ONLY** parameter in the `web.xml` file of the SSP server to true for a single Rule Execution Server and to false for a clustered Rule Execution Server.
- Use the Rule Execution Server console only for active/passive failover. When a Rule Execution Server console becomes unavailable, follow specific steps to use another Rule Execution Server console: Synchronize the other consoles manually, by clicking the **Update RuleApps** icon of each console, which forces a read from the persistence layer, and target a different Scenario Service Provider (SSP) manually.
- Deploy the Managed XOM to each Rule Execution Server console in the cluster individually. That way, you ensure that each Rule Execution Server is registered as a target for XOM deployment in the `deployment.xml` resource file. The `deployment.xml` file holds a record of the known deployments of the XOM for a particular ruleset.

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. `<ODM_InstallDir>` 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

Location	When	Files
<ODM_InstallDir>/shared/profiles/profileTemplates/rules	At product installation	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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

Location	When	Files
<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.

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. 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. You cannot augment the cell deployment manager profile.

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 deployment manager profile 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 10 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 data source for the additional nodes. For help on adding a node to an existing cluster, see “Running the addNodeToDSCluster script” on page 12.

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.virtualhost.name

The target virtual host where the applications will be installed. The default is `default_host`.

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:


```
<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:

```
<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

Parameter name	Required	Description
-dmgrAdminUsername	Mandatory	The WebSphere Application Server administrator's user identifier
-dmgrAdminPassword	Mandatory	The WebSphere Application Server administrator's password.
-clusterPropertiesFile	Mandatory	The path to the properties file.
-uninstall	Optional	Removes all the applications and resources from the cluster. The cluster itself is not removed because it might contain other applications. See "Uninstalling Rule Execution Server from a cluster" on page 14.
-createNode	Optional	If you specify this parameter, the script creates a default node profile with the name set in the -targetNodeName parameter.
-targetNodeName	Optional	By default, the name of the target installation node is set to RulesNode01.
-dmgrHostName	Optional	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.
-dmgrPort	Optional	The SOAP TCP port of the deployment manager. By default, 8879.

Results

The script performs the following actions:

- Installs the JDBC provider, execution unit (XU), 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.
- 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, make sure that the node is already federated, and set the `ODM_HOME` environment variable. The script that you run in step 2 on page 13 uses that variable.

To federate a new node into the cell of the deployment manager, perform the following steps from the new (likely remote) node:

1. Create a managed node, by running the following command from the `<WAS_InstallDir>\bin` directory:
`manageprofiles -create -templatePath ..\profileTemplates\managed -profileName <my new node profile name>`
2. Add the node to the deployment manager cell, by running the following command from the `<WAS_InstallDir>\profiles\my new node profile name\bin` directory:
`addNode.bat <dmgr hostname> <dmgr Soap port> -username <dmgr admin user> -password <dmgr admin password>`

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.virtualhost.name

The target virtual host where the applications will be installed. The default is `default_host`.

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

- 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 -targetNodeServerName serverA
```

- 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 -targetNodeServerName serverA
```

Table 4. Cluster configuration parameters

Parameter name	Required	Description
-dmgrAdminUsername	Mandatory	The WebSphere Application Server administrator's user identifier
-dmgrAdminPassword	Mandatory	The WebSphere Application Server administrator's password.
-clusterPropertiesFile	Mandatory	The path to the properties file.
-targetNodeName	Optional	By default, the name of the target installation node is set to RulesNode01.
-dmgrPort	Optional	The SOAP TCP port of the deployment manager. By default, 8879.
-targetNodeServerName	Optional	The name of the server on the target installation node. If you do not specify this parameter, a new server is created.

Results

The script configures an existing WebSphere Application Server node, which is already federated into the deployment manager of the cell, with the Decision Server artifacts. The script adds a new member to the cluster that is specified in the configureDSCluster.properties file that you pass as a parameter.

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/<profile_name>/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:

```
<WAS_InstallDir>\profiles\Dmgr01\bin\configureDSCluster.bat
-dmgrAdminUsername websphere -dmgrAdminPassword websphere
-clusterPropertiesFile
<WAS_InstallDir>\profiles\Dmgr01\bin\rules\configureDSCluster.properties
-targetNodeName DecisionServerNode01 -dmgrHostName localhost -dmgrPort 8879
-uninstall
```

- On UNIX:

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

Table 5. Cluster configuration parameters

Parameter name	Mandatory/Optional	Description
-dmgrAdminUsername	Mandatory	The WebSphere Application Server administrator's user identifier that you defined when you installed Decision Server
-dmgrAdminPassword	Mandatory	The WebSphere Application Server administrator's password that you defined when you installed Decision Server.
-clusterPropertiesFile	Mandatory	The full path to the properties file.
-uninstall	Mandatory	Removes all the applications and resources from the cluster. The cluster itself is not removed because it might contain other applications.
-createNode	Optional	If you specify this parameter, the script creates a default node profile with the name set in the -targetNodeName parameter.
-targetNodeName	Optional	By default, the name of the target installation node is set to RulesNode01.
-dmgrHostName	Optional	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.
-dmgrPort	Optional	The SOAP TCP port of the deployment manager. By default, 8879.

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.

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/Dmgr01/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. You cannot augment the cell deployment manager profile.

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 deployment manager profile 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 18 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: Run the configuration script to create the cluster and configure the specified target node as the first node. To add a new cluster member and configure an additional node, see “Running the `addNodeToDCCluster` script” on page 19. Alternatively, you can manually create the first cluster member and configure the execution unit (XU) and the datasource for the additional nodes.

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.virtualhost.name

The target virtual host where the applications will be installed. The default is `default_host`.

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 ...`

- On Windows:

```
<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:

```
<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

Parameter name	Required	Description
-dmgrAdminUsername	Mandatory	The WebSphere Application Server administrator's user identifier that you defined when you installed Decision Center
-dmgrAdminPassword	Mandatory	The WebSphere Application Server administrator's password that you defined when you installed Decision Center.
-clusterPropertiesFile	Mandatory	The full path to the properties file.
-uninstall	Optional	Removes all the applications and resources from the cluster. The cluster itself is not removed because it might contain other applications. See "Uninstalling Decision Center from a cluster" on page 22.
-createNode	Optional	If you specify this parameter, the script creates a default node profile with the name set in the -targetNodeName parameter.
-targetNodeName	Optional	By default, the name of the target installation node is set to RulesNode01.
-dmgrHostName	Optional	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.
-dmgrPort	Optional	The SOAP TCP port of the deployment manager. By default, 8879.

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: The Decision Center EAR file is very large. Therefore, one or more steps in the **configureDCCluster** script may fail due to hardware limitations or other environment constraints. If so, address the issue before rerunning the script.

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, make sure that the node is already federated, and set the `ODM_HOME` environment variable. The script that you run in step 2 on page 20 uses that variable.

To federate a new node into the cell of the deployment manager, perform the following steps from the new (likely remote) node:

1. Create a managed node, by running the following command from the `<WAS_InstallDir>\bin` directory:

```
manageprofiles -create -templatePath ..\profileTemplates\managed -profileName <my new node profile name>
```
2. Add the node to the deployment manager cell, by running the following command from the `<WAS_InstallDir>\profiles\my new node profile name\bin` directory:

```
addNode.bat <dmgr hostname> <dmgr Soap port> -username <dmgr admin user> -password <dmgr admin password>
```

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.virtualhost.name

The target virtual host where the applications will be installed. The default is default_host.

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 ...

- 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
-targetNodeServerName serverA
```

- On UNIX:

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

Table 7. Cluster configuration parameters

Parameter name	Required	Description
-dmgrAdminUsername	Mandatory	The WebSphere Application Server administrator's user identifier that you defined when you installed Decision Center

Table 7. Cluster configuration parameters (continued)

Parameter name	Required	Description
-dmgrAdminPassword	Mandatory	The WebSphere Application Server administrator's password that you defined when you installed Decision Center.
-clusterPropertiesFile	Mandatory	The full path to the properties file.
-targetNodeName	Optional	By default, the name of the target installation node is set to RulesNode01.
-dmgrPort	Optional	The SOAP TCP port of the deployment manager. By default, 8879.
-targetNodeServerName	Optional	The name of the server on the target installation node. If you do not specify this parameter, a new server is created.

Results

The script configures an existing WebSphere Application Server node, which is already federated into the deployment manager of the cell, with the Decision Center artifacts. The script adds a new member to the cluster that is specified in the `configureDCCluster.properties` file that you pass as a parameter.

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

Note: The Decision Center EAR file is very large. Therefore, one or more steps in the `addNodeToDCCluster` script may fail due to hardware limitations or other environment constraints. If so, address the issue before rerunning the script.

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

Parameter name	Mandatory/Optional	Description
-dmgrAdminUsername	Mandatory	The WebSphere Application Server administrator's user identifier that you defined when you installed Decision Center
-dmgrAdminPassword	Mandatory	The WebSphere Application Server administrator's password that you defined when you installed Decision Center.
-clusterPropertiesFile	Mandatory	The full path to the properties file.

Table 8. Cluster configuration parameters (continued)

Parameter name	Mandatory/Optional	Description
-uninstall	Mandatory	Removes all the applications and resources from the cluster. The cluster itself is not removed because it might contain other applications.
-createNode	Optional	If you specify this parameter, the script creates a default node profile with the name set in the -targetNodeName parameter.
-targetNodeName	Optional	By default, the name of the target installation node is set to RulesNode01.
-dmgrHostName	Optional	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.
-dmgrPort	Optional	The SOAP TCP port of the deployment manager. By default, 8879.

Results

This command has the following results:

- Uninstalls the clustered application (teamsaver).
- 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 88.

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 93
- To use, or to switch to, WebSphere MQ as the JMS provider, see “Configuring WebSphere MQ to be the JMS provider” on page 95
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*.
 - 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 93 or “Configuring WebSphere MQ to be the JMS provider” on page 95.

-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*.

-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\ODM870\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/ODM870/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\ODM870\WAS\profileTemplates\wbe\management"
-profileName WODMDmgr01
```

The second example unaugments an existing Decision Server Events management profile called `WODMDmgr01` on Linux and UNIX.

```
manageprofiles.sh -unaugment -templatePath
"/opt/ibm/ODM870/ODM/WAS/profileTemplates/wbe/management"
-profileName WODMDmgr01
```

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\ODM870\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/ODM870/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\ODM870\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/ODM870/ODM/WAS/profileTemplates/wbe/managed"
-profileName WODMCustom01
```

Creating profiles for an ideal topology cluster

In this task, you create the WebSphere Application Server profiles augmented with Decision Server Events that are required for an ideal 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 *"Augmenting an existing WebSphere Application Server management profile"* on page 23. You must select WebSphere Application Server default messaging as the messaging provider.
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 31.
 - 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 on page 31, 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 31, 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 28.
9. On *Computer2*, repeat steps 5 on page 31 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. See “Configuring a cluster by using the ideal topology.”

Configuring a cluster by using the ideal topology

In this task, you base the configuration of a cluster on the ideal topology, to complete creating the cluster environment.

Before you begin

Before configuring the cluster, you must first create the profiles. See “Creating profiles for an ideal topology cluster” on page 31.

The following steps assume that security is enabled for the service integration bus, as described in “Securing the service integration bus” on page 121. 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 2, 8 on page 35, 9 on page 35,10 on page 36.

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. 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 > DecisionServerCluster > 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\ODM870
```
 - On Linux and UNIX:


```
/opt/ibm/ODM870/ODM
```
 - For example, if you installed using Installation Manager and accepted the default install location:
 - On Windows:


```
C:\Program Files\IBM\ODM870
```
 - On Linux and UNIX:


```
/opt/ibm/ODM870
```
 - 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 108.

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:\filestore\directory`. 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.
- Click **Next**.
- g. Click **Change heap sizes**. Accept the default values.
 - h. 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\ODM870\runtime\wberuntimeear.ear`; and on Linux and UNIX enter `/opt/ibm/ODM870/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**.
 - g. In the Map modules to servers window, in the **Clusters and Servers** field, click **WebSphere:cell=cell-name ,cluster=DecisionServerCluster**, where *cell-name* is the name of the cell. Select both check boxes, click **Apply**, then **Next**.
 - h. Click **Finish** and **Save**.
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 `EventWidgetEar` 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 33.
 - b. Select the application server and click **Start**.

Results

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

What to do next

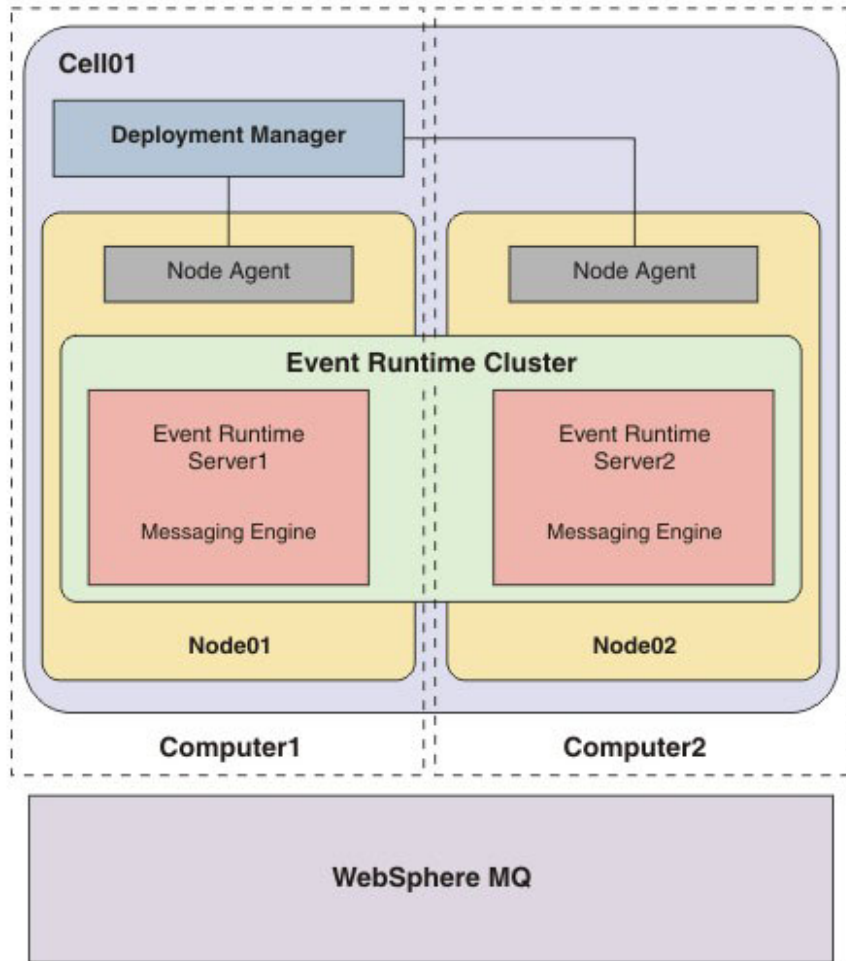
Configure the technology connectors. See Administering connector application deployment in a managed server environment for the File System, HTTP, JDBC, JMS, REST, and SOAP connectors; or see “Configuring technology connectors on a separate computer” on page 106 for the stand-alone technology connectors.

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 39.
- 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 39.
- Configure your cluster for Decision Server Events. See “Configuring a cluster that uses WebSphere MQ as the messaging provider” on page 41.

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 41.

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 “Augmenting an existing WebSphere Application Server management profile” on page 23. 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 39.
 - 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 40, 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 28.
9. On *Computer2*, repeat steps 5 on page 40 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 39.

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\ODM870`

On Linux and UNIX:

`/opt/ibm/ODM870/ODM`

- For example, if you installed using Installation Manager and accepted the default install location:

On Windows:

`C:\Program Files\IBM\ODM870`

On Linux and UNIX:

`/opt/ibm/ODM870`

- 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\ODM870\runtime\wberuntimeear.ear`; and on Linux and UNIX enter `/opt/ibm/ODM870/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 41.
 - 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 106 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. When you define the required 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 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.

Installation Steps	Persistence		
	File	Data source	JDBC
"Step 1: Selecting and applying the persistence type" on page 48		Default persistence mode	
"Step 2: Restricting database user permissions" on page 48			
"Step 3: Setting up your database" on page 49	Not applicable		

Installation Steps		Persistence		
		File	Data source	JDBC
"Step 4: Setting up a data source and connection pool" on page 51	"Creating a JDBC provider" on page 51	Not applicable		
	"Creating a data source and connection pool" on page 52	Not applicable		
	"Creating J2C authentication data" on page 53	Not applicable		
	"Setting custom properties" on page 54	Not applicable		
	"Testing the connection to the database" on page 55	Not applicable		
"Step 5: Activating security on WebSphere Application Server" on page 56	"Creating users and groups" on page 57			
	"Mapping user groups to the Monitor role" on page 59			
	"Security policies for the Rule Execution Server console" on page 60			
"Step 6: Deploying the Rule Execution Server MBean descriptors" on page 60				
"Step 7: Deploying the XU RAR" on page 61				
"Step 8: Deploying the Rule Execution Server management EAR file" on page 63				
"Step 9: Creating a Rule Execution Server database schema" on page 65	Not applicable			
"Step 10: Deploying the hosted transparent decision service EAR file" on page 71	Optional		Optional	Optional
"Step 11: Setting the DecodeUrlAsUTF8 custom property" on page 72	Required if you deployed the EAR file for hosted transparent decision services (previous step).		Required if you deployed the EAR file for hosted transparent decision services (previous step).	Required if you deployed the EAR file for hosted transparent decision services (previous step).
"Verifying the deployment and configuration" on page 81	Optional		Optional	Optional

Step 1: Selecting and applying the persistence type

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

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

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

Solaris users

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

Decision Warehouse

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

MySQL persistence

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

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

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

Related tasks:

“Repackaging the Rule Execution Server archive using Ant” on page 72

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

Step 2: Restricting database user permissions

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

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

Connection to the Rule Execution Server database, as established in the data source credentials, and any subsequent requests to the database are handled through a

database user. This database user (name and password), for example resdbUser, is defined by the database administrator and has no relation to the standard Rule Execution Server groups.

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

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

Step 3: 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 63.

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 52.

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 45.

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 49, 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

Database type	Derby	DB2
Provider type	Derby JDBC Provider	DB2 Universal JDBC Driver Provider
Implementation type	XA data source	Connection pool data source
Name	Enter a name, for example Rule Execution Server JDBC Provider	Enter, for example, DB2 Universal JDBC Driver Provider for RES

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.

```
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 45.

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 51 and click **New**.
3. In **Step 1**, enter the data source and JNDI names, and then click **Next**.
 - 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 51, 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.

Database	Properties
DB2 Universal JDBC Driver	<ul style="list-style-type: none"> • <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. • <code>driverType</code>: 2 or 4 <p>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:</p> <ul style="list-style-type: none"> • <code>serverName</code>: TCP/IP address or host name • <code>portNumber</code>: TCP/IP port number
DB2 legacy CLI-based Type 2	<code>databaseName</code> : for example, Sample.
Oracle JDBC Driver	URL: for example, <code>jdbc:oracle:oci:@sample</code>
Derby	<ul style="list-style-type: none"> • <code>databaseName</code>: path to the location of the database files. This directory must not exist already. For example, <code>c:\derbydata</code>. • Clear the option Use this data source in container managed persistence (CMP). <p>For more information, refer to the Derby documentation.</p>

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 45.

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 52.
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.
7. Under Security settings, for **Component-managed authentication alias**, select the <nodeName>/ResDerbyUser alias.
<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 <node name>/ResDerbyUser.
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 45.

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

Database	Properties
DB2 Universal JDBC Driver	<ul style="list-style-type: none"> • databaseName: The actual database name if driverType is set to 4, or a locally cataloged database name if driverType is set to 2 • driverType: The possible values are 2 or 4. The following properties are required only if driverType is 4: <ul style="list-style-type: none"> – serverName: The TCP/IP address or host name – portNumber: The TCP/IP port number
DB2 Universal JDBC XA Driver	<ul style="list-style-type: none"> • databaseName: The locally cataloged database name • driverType: 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 driverType is 4: <ul style="list-style-type: none"> – serverName: The TCP/IP address or host name – portNumber: The TCP/IP port number
DB2 legacy CLI-based Type 2	databaseName: for example, Sample
Oracle JDBC Driver	URL: For example, jdbc:oracle:oci:@sample
Derby	databaseName: The path to the location of the database files. For more information, refer to the Derby documentation.

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 45.

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

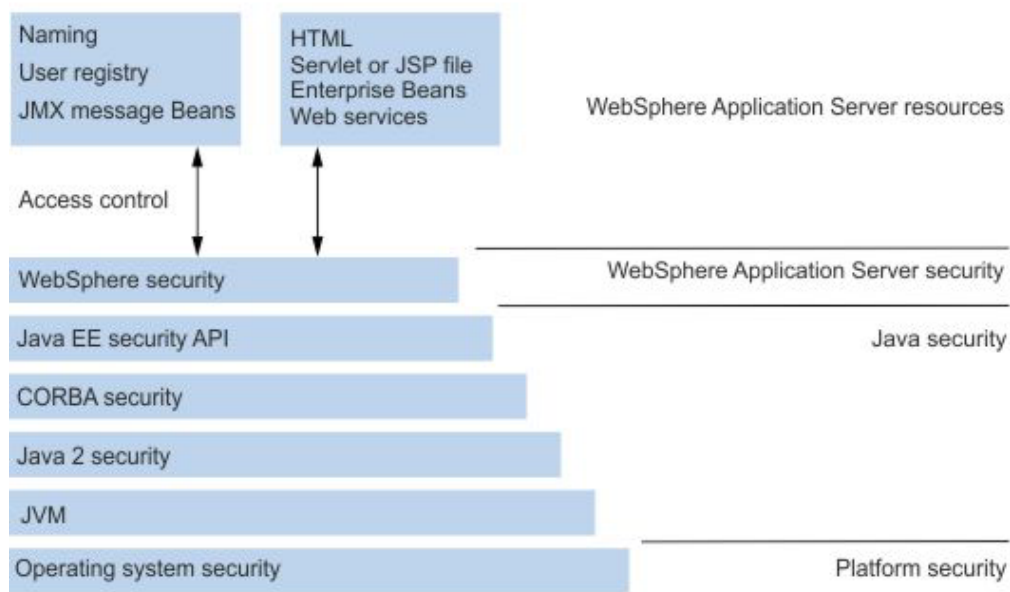
WebSphere Application Server provides security infrastructure and mechanisms to protect sensitive Java EE resources and administrative resources, and to address enterprise end-to-end security requirements on authentication, resource access control, data integrity, confidentiality, privacy, and secure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



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 59

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 45.

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

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 45.

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 45.

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.

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

Procedure

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/applicationserver/WebSphere8/jrules-res-xu-WAS8.rar`
 - `<ODM_InstallDir>/executionserver/applicationserver/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 it RES 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 5 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 Application Server Version 8.5 Information Center.

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 45.

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/applicationserver/WebSphere8/jrules-res-management-WAS8.ear`
 - `<ODM_InstallDir>/executionserver/applicationserver/WebSphere85/jrules-res-management-WAS85.ear` for WebSphere Application Server 8.5 or 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.

Role	Mapped groups
<code>resAdministrators</code>	<code>resAdministrators</code>
<code>resDeployers</code>	<code>resDeployers</code>
<code>resMonitors</code>	<code>resMonitors</code>

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.

Persistence		RuleApps	
		file	datasource or jdbc
Java XOMs	file	No Installation Settings wizard	RuleApps persistence details only
	datasource or jdbc	Java XOMs persistence details only No Decision Warehouse	Complete Installation Settings wizard

Opening the Rule Execution Server console

To open the Rule Execution Server console and the Installation Settings wizard, you must sign in with 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 68

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:

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

Option	Description
Keep drop SQL statements	Select this option to flush data from an existing Rule Execution Server database.
Show SQL statements	Select this option to display the SQL statements.

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.

Database	Database tool
IBM DB2	DB2 command line processor
Derby	ij command line processor
MySQL	mysql command line processor
Oracle	sqlplus command line processor - Run all the scripts in the SQL Plus client.
Postgre SQL	Postgre SQL command-line tool
SQL Server	Query Tool
Sybase	isql command line processor

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:

```
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 tablespace:

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

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

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

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

Procedure

1. Connect to the Rule Execution Server database:

```
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:

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

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

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

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

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

6. Disconnect from the database:

```
db2 DISCONNECT {RESDBName}
```

7. Exit the `{DB2AdminName}` UserID:

```
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 45.

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 45.

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 enterprise 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 **DecodeUriAsUTF8** 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 DecodeUriAsUTF8 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 **DecodeUriAsUTF8** 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 45.

Procedure

To set the **DecodeUriAsUTF8** 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 **DecodeUriAsUTF8** 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.

Repackaging the Rule Execution Server archive using Ant

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

Before you begin

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

About this task

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

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

Procedure

To repackage an archive file using Ant:

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

```
ant -Dxu.in=../applicationserver/WebSphere<version_number>/jrules-res-xu-WAS<version_number>.rar
-Dxu.out=myxu.rar
-Dconsole.ear.in=../applicationserver/WebSphere<version_number>/jrules-res-management-WAS<version_number>.ear
-Dconsole.ear.out=mymanagement.ear
-Dpersistence.type=file -f ressetup.xml setup
```

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**.

Option	Description
Name	JRules Topic Connection Factory
JNDI name	jms/BRESTopicConnectionFactory

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**.

Option	Description
Name	JRules Input Topic
JNDI name	jms/BRESTopicIn
Topic name	JRulesTopicIn

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/IlrRuleExecutionEJB

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:
<InstallDir>/executionserver/applicationservers/
WebSphere<version_number>/jrules-res-mdb-WAS<version_number>.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

According to your development phases, you can set up different Rule Execution Server environments (for example development, QA, and production) 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.

Component	Description	Default installation
Event Capture and Replay widgets	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.	Yes
Event Connectors	Event connectors provide data connections between the event runtime and external systems. You can configure the event connectors in Event Designer.	Yes
Event Designer	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.	Yes
Event Runtime	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.	Yes

Table 11. Decision Server Events components. (continued)

Component	Description	Default installation
Event Tester widget	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.	Yes
Integration Components	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.	No
Samples and Tutorials	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.	Yes

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.
 - 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.
- **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 23.

- **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 28.

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 86
- “Customizing the event runtime” on page 86

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 the product with the launchpad 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 88.

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 111.

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 *Creating a new Decision Server Events application server profile* or *Augmenting an existing WebSphere Application Server application server profile with Decision Server Events*.
- **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:

- Ideal topology:

An ideal 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 cluster by using the ideal topology” on page 33.

- 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 37.

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 88.

- **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 103. 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 an ideal 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 106. 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 101.
 - 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 102.

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 88.

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 name	SQL statement
Asset definitions (maestro_asset, properties)	<pre>CREATE TABLE maestro_asset ALTER TABLE maestro_asset CREATE TABLE properties CREATE TABLE labels ALTER TABLE labels</pre>
Time delays	<pre>CREATE TABLE time_based_asset CREATE INDEX FIRETIME_IDX CREATE INDEX TBDEL_IDX</pre>
Event Flows	<pre>CREATE TABLE steps CREATE CLUSTERED INDEX STREAM_IDX CREATE TABLE csio_entry CREATE INDEX csio_entry_idx on csio_entry CREATE INDEX csio_entry_times</pre>
User Console	<pre>CREATE TABLE actions ALTER TABLE actions CREATE TABLE actions_prefs ALTER TABLE actions_prefs CREATE TABLE auth_users CREATE TABLE auth_groups</pre>

Table 12. SQL statements to create each table. (continued)

Table name	SQL statement
History	<pre> CREATE TABLE history_event CREATE INDEX history_e_idx CREATE TABLE history_action CREATE INDEX history_a_idx CREATE TABLE history_ent_obj CREATE INDEX history_eo_idx CREATE TABLE history_ent_obj_values CREATE INDEX history_eo_v_idx CREATE TABLE history_filter CREATE TABLE history_rule CREATE INDEX history_rule_idx CREATE TABLE history_watch_time CREATE VIEW as_director_events CREATE VIEW as_director_actions CREATE VIEW as_director_ent_objs CREATE VIEW as_director_ent_objs_values CREATE VIEW as_director_filters CREATE VIEW as_director_rules CREATE VIEW as_director_watch_times </pre>

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 90.
 - 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 name	Property
Asset definitions	as.director.common.db.asset.instance
Time delays	as.director.common.db.timebased.instance
Event Flows	as.director.common.db.eventflow.instance
User Console	as.director.connectors.pem.instance
History	as.director.common.db.history.instance

For more information, see “Setting properties” on page 97.

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 97.
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 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 87.

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

Operating system	Location in which you must run the command	Commands
Windows	<InstallDir>\config\was\	configure_MQ_JMS_Messaging.bat -undo
Linux and UNIX	<InstallDir>/config/was/	configure_MQ_JMS_Messaging.sh -undo

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

Operating system	Location in which you must run the command	Commands
Windows	<InstallDir>\config\was\	configure_bus.bat configure_messaging.bat
Linux and UNIX	<InstallDir>/config/was/	configure_bus.sh configure_messaging.sh

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 121.

- 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

- Create the queue manager, by using the **crtmqm** command.
- Start the queue manager, by using the **strmqm** command.
- If you are using WebSphere MQ V7.1, publish/subscribe must be enabled. Use the **ALTER QMGR** command for this queue manager.
- 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.

- 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.

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 87.

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.

Operating system	Location in which you must run the command	Command
Windows	<InstallDir>\config\was\	configure_messaging.bat -undo
Linux, UNIX and z/OS	<InstallDir>/config/was/	configure_messaging.sh -undo

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.

Connection factories	JNDI name
Queue connection factory	jms/WbeQueueConnectionFactory
Topic connection factory	jms/WbeTopicConnectionFactory
Connection factory	jms/WbeConnectionFactory

Table 18. Topics.

Topics	JNDI name
Action topic	jms/actionTopic
Command topic	jms/commandTopic
CbeListener	topic jms/WBE/CbeListener
Durable action topic	jms/durableActionTopic
Durable event topic	jms/durableEventDestination

Table 18. Topics. (continued)

Topics	JNDI name
Event topic	jms/eventDestination
History topic	jms/historyModuleDestination

Table 19. Activation specifications.

Activation specification name	JNDI name
wbe_events	jca/wbe_events
wbe_events_durable	jca/wbe_events_durable
wbe_history	jca/wbe_history
wbe_reset_watch	jca/wbe_reset_watch
wbe_ute_receive	jca/wbe_ute_receive

- 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.

Method	Platforms
“Configuring properties by using the WebSphere Application Server administrative console”	Multiplatforms and z/OS
“Configuring properties by using the wsadmin scripting tool”	Multiplatforms and z/OS
“Configuring properties by using the Properties tool” on page 99	Multiplatforms only

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 :

```
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 21. Custom properties to be set.

Property	Value	Purpose	Required
----------	-------	---------	----------

Table 21. Custom properties to be set. (continued)

as.director.server.eventflow.persistentStepStore	false	Turns off persistent storage and causes step data to be stored in memory	yes
as.director.server.eventflow.cacheStepStoreTimeout.ms	5000 milliseconds (default)	Determines amount of time, in milliseconds, for which an instance of step data is kept in memory	no – default is 5000 ms (5 seconds)

See “Setting properties” on page 97 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 97 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:

```
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 22. Durable event queue size property

Property name	Default	Units
as.director.server.DurableEventQueueSize	20	MB

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 “Setting properties” on page 97.

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

Property	Value
For event rules: as.director.server.timebased.event.maxtardiness.minutes	The maximum amount of time, in minutes, beyond the scheduled time when a time-delayed event rule is still run. 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.
For actions: as.director.server.timebased.action.maxtardiness.minutes	The maximum amount of time, in minutes, beyond the scheduled time when a time-delayed action is still run. 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.

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 97. 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/ODM870/ODM` and on Windows `C:\Program Files\IBM\ODM870`).
 - 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 87 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>\derby\userconsole;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:

Table 24. Property values to set.

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. In you are using another other supported database:

Table 25. Property values to set.

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

- JMS messaging

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 Logging information.
- Turn off the recording of history, if not required. See “Configuring the event runtime to record history” on page 102.
- 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

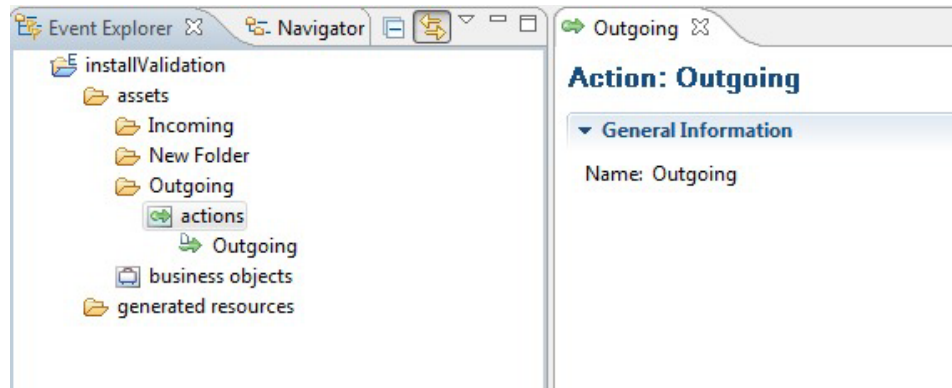
Install and configure Decision Server Events, including configuring a JMS messaging provider.

Ensure that you have installed Event Designer. The event runtime can be installed on a separate computer.

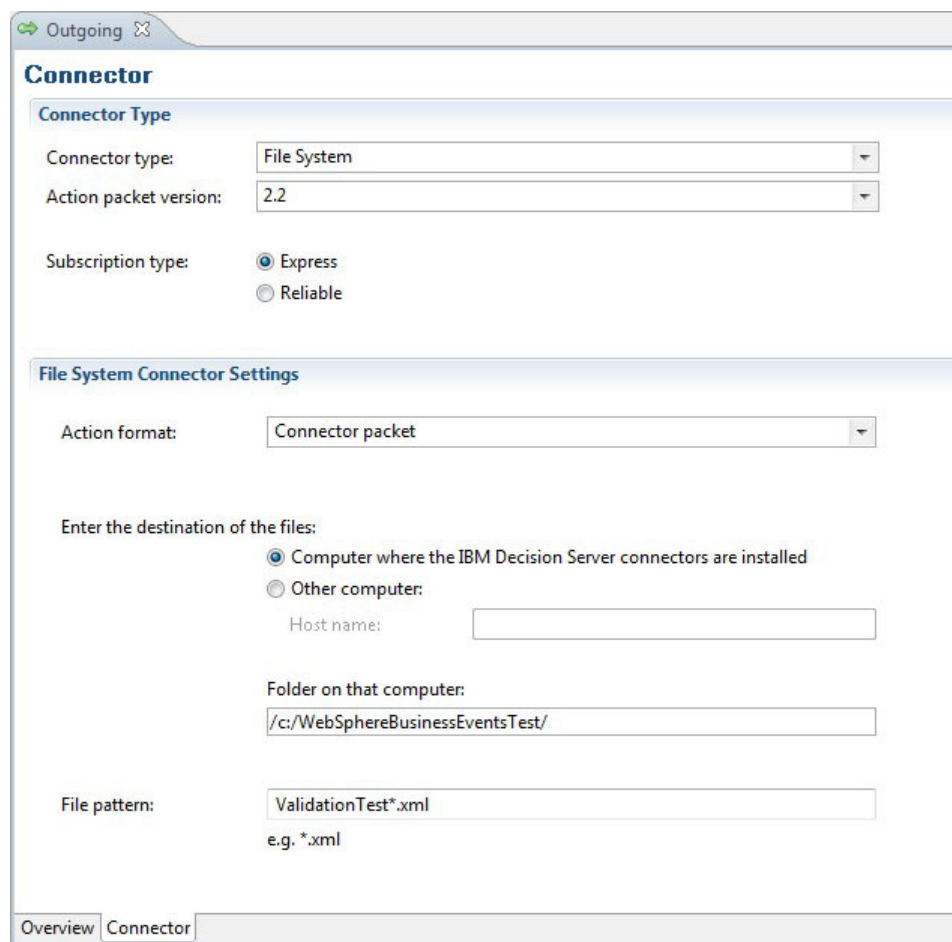
If you turned on WebSphere Application Server security during installation, you must configure user authentication to the event runtime before verifying the installation. See “User authentication” on page 115

Procedure

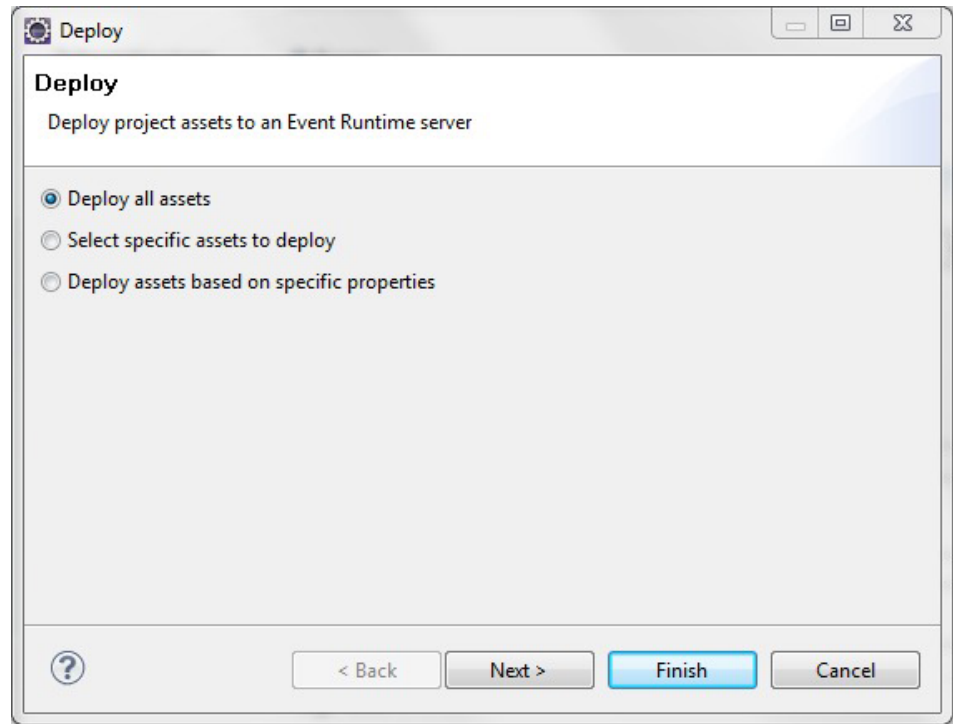
1. On the computer on which you have installed the event runtime, ensure that WebSphere Application Server is running.
2. Check that the event runtime application has been installed and started on WebSphere Application Server:
 - a. Start the WebSphere Application Server administrative console. See Starting 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
 - EventWidgetsEarEnsure that the following application is listed and shown as started if you have prepared the environment for testing:
 - wbetesterear
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:



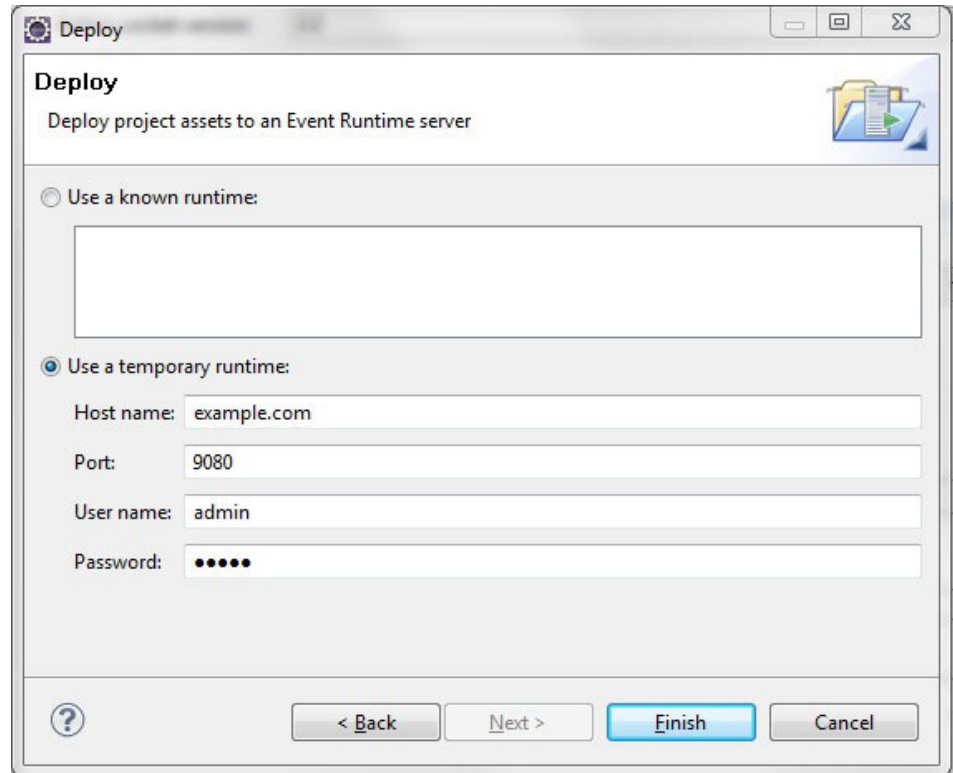
8. In the Action editor, click the **Connector** tab, shown in the following screen capture:



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\ODM870\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
Connectors>
```

For more information about starting connectors, including details of the optional parameters, 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\ODM870\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/ODM870/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.

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.

Property	Value to set
as.director.common.authentication.method	AppServer 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.
as.director.server.authentication.username	A user ID that is the primary administrative user for WebSphere Application Server.
as.director.server.authentication.password	The password associated with the user ID.

For details of how to set these properties, see “Setting properties” on page 97.

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 116.

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 27. Groups to be created.

Access level	Name of group to be created	Example
Administrator	<p>WBEAdmins.</p> <p>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.</p> <p>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).</p>	WBEAdmins
User	Choose any name, but the name must match the value of <code>as.director.common.authentication.userGroup</code> set in Properties.	WBEUsers

2. For details of how to set these properties, see “Setting properties” on page 97.
3. Change the following properties:

Table 28. Properties to be changed.

Property	Action
<code>as.director.common.authentication.method</code>	Select LDAP.
<code>as.director.common.authentication.server</code>	<p>Replace <HOST-NAME> with the name of the server where LDAP resides.</p> <p>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.</p>
<code>as.director.common.authentication.namecontext</code>	Remove <code>dc=<MYDOMAIN></code> and replace <code><MYCOMPANY></code> with company name.
<code>as.director.server.authentication.member.filterEEE</code>	Enable.
<code>as.director.server.authentication.member.attribute</code>	Enable.
<code>as.director.server.authentication.userid.format</code>	Only for internal use, do not change.
<code>as.director.server.authentication.security.method</code>	<p>Set to one of:</p> <ul style="list-style-type: none"> • none (default) • simple • strong
<code>as.director.common.authentication.security.protocol</code>	<p>Set to one of:</p> <ul style="list-style-type: none"> • unspecified (default) • ssl
<code>as.director.common.authentication.userGroup</code>	Set the value to match the name of the user group defined in the Active Directory (the default is WBEUsers).

Table 28. Properties to be changed. (continued)

Property	Action
as.director.server.authentication.username	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 118).
as.director.server.authentication.password	Replace admin with the password associated with the user ID.

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 116. 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.

Access level	Name of group to be created	For example
Administrator	<p>WBEAdmins.</p> <p>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.</p> <p>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).</p>	WBEAdmins
User	Choose any name, but the name must match the value of <code>as.director.common.authentication.userGroup</code> set in Properties.	WBEUsers

2. For details of how to set these properties, see “Setting properties” on page 97.
3. Change these properties:

Table 30. Properties to change.

Property	Action
as.director.common.authentication.method	Select Active Directory

Table 30. Properties to change. (continued)

Property	Action
as.director.common.authentication.server	Replace <HOST-NAME> 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.
as.director.common.authentication.namecontext	Replace <MYDOMAIN> with domain name and <MYCOMPANY> with company name.
as.director.common.authentication.userGroup	Set the value to match the name of the user group defined in the Active Directory (the default is WBEUsers).
as.director.server.authentication.username	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 119).
as.director.server.authentication.password	Replace admin with the password associated with the user ID.

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 116. 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 97.
 - b. Change these properties:

Table 31. Properties to be changed.

Property	Value to set
as.director.common.authentication.method	Database
as.director.server.authentication.username	user ID
as.director.server.authentication.password	password of your choice

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 97.

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 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 95 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 93.

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 97.

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 for distributed platforms or for Linux on System z, 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 16.

Operational Decision Manager 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 the Decision Center consoles on WebSphere Application Server for distributed platforms or for Linux on System z, you install the necessary software, secure the appropriate access rights, and follow specific steps. Some are mandatory, while others are optional.

Before you begin

This configuration guide assumes that you use a profile with administrative security enabled.

Before you configure Decision Center, you must install the following software:

- WebSphere Application Server for Linux on System z 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

About this task

Decision Center includes two consoles: Enterprise and Business. Both consoles work on WebSphere Application Server 8.0 and 8.5.

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 consoles). The first time that you open Decision Center, it does not contain a rule project. You must publish a project from Rule Designer (see Publishing a project).

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.

Step	Required
Step 1: Restricting database user permissions	

Step		Required
"Step 2: Configuring database connectivity" on page 128	"Creating the DB2 database" on page 128	Optional
	Creating a JDBC provider	
	"Creating a data source and a connection pool" on page 130	
	"Creating J2C authentication data" on page 131	Optional
	"Changing the custom properties of your data source" on page 132	Optional
	"Testing the connection to the database" on page 133	
"Step 3: Configuring security" on page 133	"User registry" on page 134	
	"Creating users and groups" on page 135	Optional
	"Security policies" on page 136	Optional
"Step 4: Deploying the Decision Center EAR file" on page 137	"Declaring custom groups" on page 137	Optional
	"Deploying the EAR file" on page 138	
	"Changing module properties" on page 140	
"Step 5: Verifying the deployment of the Decision Center consoles" on page 141		
"Step 6: Completing the configuration of the Decision Center consoles" on page 141	Completing the configuration from the Decision Center Enterprise console	
	Completing the configuration by using Ant tasks	Select one of the two methods. Required for distributed platforms only.

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 `rtbdbUser`, is defined by the database administrator and has no relation to the standard Decision Center groups.

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

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

Step 2: 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 the DB2 database

An empty DB2 database is required for Decision Center.

Before you begin

To create databases, you need DB2 Administrator rights.

About this task

If the Decision Center database does not exist, you must create it.

Procedure

To create the DB2 database:

1. Use 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.DB2.

About this task

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.

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.

1. *InstallDir/teamsserver/jcl/HBRDCCDB.jcl*
2. *InstallDir/teamsserver/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.

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 125.

Procedure

1. In the side panel of the console, open **Resources > JDBC**, and click **JDBC Providers**.

2. In **Scope**, select Node=xxx, Server=yyy.
xxx is the name of your node and yyy 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

Database type	Derby	DB2
Provider type	Derby JDBC Provider	DB2 Universal JDBC Driver Provider
Implementation type	Connection pool data source	Connection pool data source
Name	Enter a name, for example Rule Execution Server JDBC Provider	Enter, for example, DB2 Universal JDBC Driver Provider for DC

5. Click **Next**.
6. In **Step 2**, confirm or modify the database class path and click **Apply**, and then click **Next**.
Step 3 is a summary.
7. Click **Finish** and **Save** to save the changes directly to the master configuration.
8. Optional: If you are configuring Decision Center for WebSphere Application Server for Linux on System z, the DB2 installation might have installed the file db2jcc_license_cisuz.jar in a location that is not in the WebSphere Application Server path.
 - a. Use the following command to find the db2jcc_license_cisuz.jar file: `find / -name db2jcc_license_cisuz.jar`

Tip: It is best to do this as the root user to avoid permissions messages.
 - b. From the WebSphere Integrated Solutions Console, open **Environment > WebSphere variables** and click **DB2UNIVERSAL_JDBC_DRIVER_PATH**.
 - c. Set the value to the path discovered by the **find** command.
9. 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 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 125.

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 129, 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**.
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 129, and click **Next**.
5. In **Step 3**, enter the database properties for the data source and click **Next**.
The following table shows the minimum set of mandatory properties to define the supported databases. If the table does not include your driver, check the WebSphere Application Server documentation for more information.

Database	Properties
DB2 Universal JDBC Driver	<ul style="list-style-type: none"> • <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 • <code>driverType</code>: 2 or 4 <p>For <code>driverType</code> 4:</p> <ul style="list-style-type: none"> • <code>serverName</code>: TCP/IP address or host name • <code>portNumber</code>: TCP/IP port number
DB2 legacy CLI-based Type 2	<code>databaseName</code> : for example, <code>Sample</code> .
Oracle JDBC Driver	URL: for example, <code>jdbc:oracle:oci:@sample</code>
Derby	<ul style="list-style-type: none"> • <code>databaseName</code>: path to the location of the database files. This directory must not exist already. For example, <code>c:\derbydata</code>. • Clear the option Use this data source in container managed persistence (CMP). <p>For more information, refer to the Derby documentation.</p>

Tip: If the transaction isolation level is not set to `READ-COMMITTED`, you might encounter database access problems. Configure data source isolation to provide a better concurrency experience. For more information, see the following technote: <http://www-01.ibm.com/support/docview.wss?uid=swg21224492>.

6. In **Step 4**, proceed as follows:
 - WebSphere Application Server for distributed platforms: Select the default values in the list and click **Next**.
 - WebSphere Application Server for Linux on System z: Select the DB2 Administrator's authentication alias from the **Component-managed authentication alias** option, and then 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 125.

About this task

J2C is a secure mechanism for integrating enterprise information systems to an application server and enterprise applications. After you have created your data source and connection pool, you can create the J2C authentication data. If you completed this task as part of the Rule Execution Server configuration, you do not need to do it a second time. The procedure is included here solely for completeness.

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, for Derby uses the following values:
 - **Alias:** `RtsDerbyUser`
 - **User ID:** `rtsAdmin`
 - **Password:** `rtsAdmin`
 - For DB2, provide the DB2 Administrator's credentials in the **User ID** and **Password** fields.
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 125.

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.

Testing the connection 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 WebSphere 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.
```

Potential connection errors might be:

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

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.

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
- Stand-alone Lightweight Directory Access Protocol (LDAP) registry
- Stand-alone custom registry

A federated user repository serves as the active user registry.

- If you no groups or users are defined or if you want to define new groups or users, go through the steps of “Creating users and groups” on page 135. 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 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, you must 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.

Group	Use	Default user and password
<code>rtsAdministrator</code>	Mandatory, gives the user administrator access.	<code>rtsAdmin</code> , <code>rtsAdmin</code>
<code>rtsConfigManager</code>	Mandatory, gives the user configuration manager access.	<code>rtsConfig</code> , <code>rtsConfig</code>
<code>rtsUser</code>	Mandatory, gives a user standard access.	<code>rtsUser1</code> , <code>rtsUser1</code>
<code>rtsInstaller</code>	Mandatory, gives the user access to the Installation Manager.	<code>rtsAdmin</code> , <code>rtsAdmin</code>
<code>Validator</code>	Optional custom group, used in the Decision Center permissions tutorial.	<code>Val</code> , <code>Val</code>
<code>Eligibility</code>	Optional custom group, used in the Decision Center permissions tutorial.	<code>Eli</code> , <code>Eli</code>

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 125.

About this task

At this stage of the configuration, creating users and groups is optional.

- If suitable groups and users are already defined, skip this procedure and connect users to their appropriate roles when you deploy your applications.
- If no groups and users are defined or if you want to define new groups and users, configure a federated repository as follows.

Procedure

To configure a federated repository:

1. From the side panel of the administration 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**, 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, 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 existing groups.

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**, 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**.
12. Enter a given name and surname for rtsUser1, then enter the rtsUser1 password, and click **Create**.
13. Click **Close**.
14. Click **Create Like** and repeat steps 12 to 13 to create more users and assign them roles as shown in the following table. The Val and Eli users are optional.

Table 33. Decision Center users and groups

User ID and password	Roles
rtsUser1 - rtsUser1	rtsUser
rtsAdmin - rtsAdmin	rtsAdministrator, rtsInstaller
rtsConfig - rtsConfig	rtsConfigManager
Val - Val	rtsUser, Eligibility, Validator
Eli - Eli	rtsUser, Eligibility

Back on the Manager Users page, the table shows your users.

15. Restart your application server.

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.

After the global security of WebSphere Application Server is activated, the MBean server cannot be accessed from the deployed application.

After the global security of WebSphere Application Server is activated, the MBean server cannot be accessed from the deployed application. Therefore, 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 theDecision 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 repack the EAR file, as explained in “Repackaging the Decision Center archive” on page 156.

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.

Location in the administrative console	Property and value	Description
Application servers > server1 > Session management > Custom properties	<code>InvalidateOnUnauthorizedSessionRequestException = true</code>	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: http://www.ibm.com/support/docview.wss?uid=swg21609826 .
Global Security > Custom properties	<code>com.ibm.websphere.security.setContextRootForFormLogin = true</code>	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: http://www.ibm.com/support/docview.wss?uid=swg1PM58885 .

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`.

However, you must add any custom groups that you declared previously, as described in “Declaring custom groups” on page 137. For example, 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 before deployment.

- WebSphere Application Server for distributed platforms: `<ODM_InstallDir>/teamserver/applicationservers/WebSphere<version_number>/jrules-teamserver-WAS<version_number>.ear`
- WebSphere Application Server for System z on Linux: `<+JRULESHOME+>jrules-teamserver-WAS<version_number>.ear`. The `<+JRULESHOME+>` placeholder is your `z/OS zFS JRules` installation directory.

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](#).

Tip:

Procedure

To add your custom groups to the Decision Center deployment descriptor:

1. Before you modify the EAR file, create a back-up copy of the EAR file.
2. Decompress 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 156.

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 125.

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 module properties are lost. See “Changing module properties” on page 140. 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, and then click **Next**.
 - WebSphere Application Server for distributed environments:
 - <ODM_InstallDir>/teamserver/applicationservers/WebSphere8/jrules-teamserver-WAS8.ear
 - <ODM_InstallDir>/teamserver/applicationservers/WebSphere85/jrules-teamserver-WAS85.ear
 - WebSphere Application Server for Linux on System z:
<+JRULESHOME++>jrules-teamserver-WAS8.ear. <+JRULESHOME++> is your z/OS zFS Operational Decision Manager installation directory.
3. Select **Detailed - Show all installation options and parameters**.
4. Expand **Choose to generate default bindings and mappings**, select the **Generate Default Bindings** check box, and then click **Next**.
5. If a security warning is displayed, click **Continue**.
6. On the Install New Application page, in **Step 1**, click **Next** to accept the default settings.
7. In **Step 2**, select the target server and the WAR files and click **Next**. Keep the default setting if you have only one server.
8. For **Step 3 to Step 8**, click **Next** to accept the default settings.
9. 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 9a to 9d 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

Role	Mapped groups
rtsUser	rtsUser
rtsAdministrator	rtsAdministrator
rtsConfigManager	rtsConfigManager
rtsInstaller	rtsInstaller
Validator	Validator
Eligibility	Eligibility

10. 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.

11. Click **Finish**.
12. Click **Save** to save your workspace changes to the master configuration.

What to do next

Change some module properties.

Changing module properties

After you deploy the EAR file, you must change some module properties: the class loading sequence and the session management.

Before you begin

To change the module 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 125.

About this task

The Decision Center application does not support the default parent first configuration, and **Override session management** must be checked.

Procedure

To change the module properties:

1. In the console, open **Applications > Application Types > WebSphere enterprise applications**.
2. Click the name that you specified for the Decision Center application. For example, `teamserver-WAS85`.
3. Under **Modules**, click **Manage Modules**.

4. Click teamsserver.
5. In the **Class loader order** menu, select **Classes loaded with local class loader first (parent last)**, and then click **Apply**.
6. In **Additional Properties** click **Session Management**, then select **Override session management**, and then click **OK**.
7. Click **Save** to save the changes directly to the master configuration.
8. Click decisioncenter and repeat the same actions.
9. In the side panel, open **Applications > Application Types > WebSphere enterprise applications**.
10. 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 consoles

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

About this task

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

Procedure

1. Start the server.
2. Start a new browser instance and enter the URL to access the Decision Center consoles.
 - The URL to access the Enterprise console is `http://localhost:<PORT_NUMBER>/teamsserver`. Set `<PORT_NUMBER>` to the port number for the web application. The default is 8080, but you can change the connector port in the `server.xml` file. If your browser is not running on the same host as the application server, replace `localhost` with the address of the host.
 - The URL to access the Business console is `http://localhost:<PORT_NUMBER>/decisioncenter`.

The sign 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 consoles*.

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

You use the Installation Settings wizard for the following actions.

Table 35. Actions in the Installation Settings wizard

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

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

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

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

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

Step 1: Configure the database

You use the Installation Settings wizard to configure the database.

About this task

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

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

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

Procedure

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

Step 2: Set up message files

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

About this task

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

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

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

Procedure

To declare a message file in the Installation Settings wizard:

1. Click **New**.

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

Results

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

Example

For example:

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

Step 3: Set up groups

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

Before you begin

Before you set up groups in the Enterprise console, make sure to add all the groups that you want to see in the available list when you enforce project security or set 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 or WAR 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 **teamserv** prefix, which is not in the `preferences.properties` file. You must include the prefix when you set configuration parameters in the Installation Settings wizard.

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

Procedure

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

What to do next

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

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

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

[-DdbSchemaName=<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=output.sql
```

Results

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

Executing the database schema script:

You execute the database schema script.

Procedure

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

-Dserver.url=<server url>

-DdatasourceName=<data source name>

[-Dfile=<SQL file>]

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

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

Uploading the database schema extension:

You upload the database schema extension.

Procedure

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

-Dserver.url=<server url>

-DdatasourceName=<data source name>

-DextensionModel=<model file>

The model description (`.brmx` extension).

-DextensionData=<data file>

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

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

Results

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

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

Uploading a list of roles or groups to the database:

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

Before you begin

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

About this task

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

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

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

Procedure

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

-Dserver.url=<server url>

-DdataSourceName=<data source name>

-Droles=<role list>

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

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

Removing a database schema:

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

About this task

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

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

Procedure

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

-Dserver.url=<server url>

-DdataSourceName=<data source name>

-DextensionModel=<model file>

The description of the database schema to remove.

[-DdbSchemaName=<database schema name>]

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

[-DoutputFile=<SQL file>]

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

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

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

-Dserver.url=<server url>

-DdataSourceName=<data source name>

[-Dfile=<SQL file>]

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

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

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

Example

Here is the complete code sample:

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

Defining and uploading message files

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

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

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

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

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

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

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

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

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. You must either invoke the Ant command from the *InstallDir/teamserver/bin* directory, as in the example, or specify the path to a target *build.xml* file.

Use the following format for the names:

```
<Users>
  <User><LoginId>user1</LoginId></User>
  <User><LoginId>user2</LoginId></User>
</Users>
```

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=rtAdmin -DrtsAdmin.password=rtAdmin
-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`.

Parameter	Use
<code>teamserver.<extractorValidator>.class</code>	Specify a ruleset extractor validator class to use for the <code>extractorValidator</code> name. The class must implement the <code>IlrExtractorValidator</code> interface. After this class is defined, specify this name as the extractor validator to use when defining a ruleset extractor.

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

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

set-config-param

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

Parameters:

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

For example:

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

remove-config-param

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

Parameters:

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

print-config-param

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

Parameters:

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

- **-DdatasourceName=<data source name>**
- **[-Duser=<username>]**
- **-Dkey=<parameter key>**

Repackaging the Decision Center archive

You can 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 servers that require WAR files.

-DsourceEar=<source ear>

-DsourceWar=<source war> for servers that require WAR files.

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

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

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

Setting parameters for the Solr search engine

The search function in the Decision Center Business console runs on Apache Solr.

By default, the search uses an embedded instance of the engine. Alternatively, you can run the search on a remote instance of the engine on either another computer or the same computer but in a dedicated web application. For more information, see “Using a remote Solr search engine” on page 156.

You can set three parameters for the search engine:

Table 36. Search parameters

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

You set the configuration parameters for the search engine in the `preferences.properties` file for Decision Center. The `preferences.properties` file can be placed in any package, and the names of the search properties depend on the location of the file, for example with `SearchProvider`:

- If you place the `preferences.properties` file in `../rules/decisioncenter/web/search/`, the name of the property must be `SearchProvider`.
- If you place the `preferences.properties` file in `../rules/decisioncenter/web/`, the name of the property must be `search.SearchProvider`.

See also `IlrPropertyManager`

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

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

If you do not specify a directory for `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 `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 testing and simulation

You can deploy and configure testing and simulation features to complement your Decision Center configuration on WebSphere Application Server.

Before you start

You start by installing certain applications, and then you follow a sequence of steps to deploy the testing and simulation features.

Make sure the following applications are installed and configured on the same application server:

- Rule Execution Server
- Decision Center

To use the testing and simulation features in the Decision Center consoles, you must deploy the testing and simulation archive, which contains the following WAR files:

- Scenario Service Provider (SSP): Used to run tests and simulations in the Enterprise console and tests in the Business console.
- Decision Runner: Used to run simulations in the Business console.

What steps to follow

The following table summarizes the steps to configure the testing and simulation features.

Note: You can also do the configuration by following the steps in Configuring Rule Execution Server on WebSphere Application Server for z/OS using wsadmin scripts.

Step	Mandatory/Optional
“Step 1: Creating Decision Warehouse database resources”	Mandatory. If you used the Rule Execution Server console to create database resources, you do not have to do this step.
“Step 2: Creating Decision Runner database resources” on page 161	Mandatory. To run simulations in the Business console, you must create dedicated tables for the Decision Runner in your database.
“Step 3: Deploying the testing and simulation archive” on page 161 This archive is the default testing and simulation archive that is packaged with the installer.	Mandatory
“Step 4: Checking the availability of the testing and simulation services” on page 163	Optional
“Step 5: Using an Ant task to package the SSP archive” on page 164	Optional
“Step 6: Redeploying a testing and simulation archive” on page 165	Optional

Step 1: Creating Decision Warehouse database resources

You set up the Decision Warehouse to support testing and simulation services.

To use the Decision Warehouse, you must create dedicated tables in your database. You can use SQL scripts to create these tables. The SQL scripts are 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 the Decision Warehouse. If you are configuring the testing and simulation features 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. Also, you can use the Decision Warehouse only if the Rule Execution Server persistence is set to `datasource` or `jdbc`.

Only users belonging to `resAdministrators` or `resMonitors` groups can see the Decision Warehouse tab in the Rule Execution Server console after the testing and simulation features are configured.

Use any tool that can handle SQL to import and run the SQL scripts. 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.

The following table shows the tools for the supported databases:

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

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

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

Install a database client for the database that you use (refer to the documentation for the database).

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.

Using the DB2 database

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;
```

- Depending on your database settings, you might have to modify the script further.

Step 2: Creating Decision Runner database resources

To run simulations in the Business console, you must create dedicated tables in your database for the Decision Runner.

About this task

You can create the tables by using the Rule Execution Server console or an SQL script.

Procedure

Select a method to create the tables:

- To create the tables with the Rule Execution Server console:
 1. Open the Rule Execution Server console (see Rule Execution Server console online help).
 2. Run **Diagnostics**.
 3. In the Decision Runner section, follow the link to the installation wizard and use the wizard to create the tables.

Note: The Decision Runner section shows the link to the wizard only if the tables do not already exist.

- To manually create the tables:
 1. In `<ODM_InstallDir>/executionserver/databases`, select the SQL script that matches your database and run it with the appropriate database tool. The script that creates the tables for the Decision Runner is named `decisionrunner_<database_name>.sql`.

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

Step 3: Deploying the testing and simulation archive

After creating the database resources, you can deploy the testing and simulation archive.

Before you begin

You deploy the testing and simulation archive from the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 125.

About this task

Use this procedure to deploy the default testing and simulation archive that is packaged with the installer or for any subsequent deployment of a repackaged

archive. You can also deploy the default file and use it to test your rules if you have an XML XOM. An XML XOM is included in the ruleset archive of each RuleApp, so you do not have to repackage the default file to include the XOM.

You must deploy the archive to the same server as the execution unit (XU). Moreover, for the testing and simulation services to work, the SSP and Decision Runner WARs in the archive must be installed on the same server and port as the Rule Execution Server console.

If a Rule Execution Server console instance is not deployed on the same server and port, you must implement the `IlrSSPResolver` interface for SSP. In the resolver, you can use the server name to return a specific server URL.

For the Decision Runner, if a Rule Execution Server console instance is not deployed on the same server and port, you must set the `RES_URL` init parameter on the Decision Runner web application to the Rule Execution Server console that uses the Decision Runner.

If you are installing the testing and simulation features on z/OS, you can perform the task by using customizable installation scripts. For more information, refer to *Configuring Rule Execution Server on WebSphere Application Server for z/OS* using `wsadmin` scripts.

Procedure

To deploy the testing and simulation 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/applicationsservers/WebSphere8/jrules-ssp-WAS8.ear`
 - For WebSphere Application Server 8.5 or 8.5.5: `<ODM_InstallDir>/executionserver/applicationsservers/WebSphere85/jrules-ssp-WAS85.ear`

The EAR file contains the WAR files for the Scenario Service Provider (SSP) and the Decision Runner.

4. Select the check box **Detailed - Show all installation options and parameters**.
5. Expand **Choose to generate default bindings and mappings**, select the check box **Generate Default Bindings**, and click **Next**.
6. Click **Continue** to accept the security warning.
7. In **Step 1** through **Step 6** click **Next** to accept the default settings.
8. In **Step 7, Map resource references to resources**, select the work managers that you want to use for SSP and the Decision Runner. SSP and the Decision Runner use work managers to run test suites and simulations in managed threads. You have to select work managers that are defined on the application server.

You can select the default work manager (`wm/default`) or use a dedicated work manager that you will have to create on the application server. 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. Click **Next**.

9. In **Step 8** through **Step 9**, click **Next** to accept the default settings.
10. In **Step 10**, **Map security roles to users or groups**, if you have activated security, select **resAdministrators** and click **Map groups**.
11. Click **Search**. Modify the maximum number of displayed results if you do not see **resAdministrators** in the Available list, and then click **Search** again.
12. Click the **resAdministrators** group under Available and then click the arrows to move it to the **Selected** column.
13. Click **OK** to return to the **Map security roles to users or groups** page.
14. Repeat 10 through 13 for the **resDeployers** group, selecting **resDeployers**.
15. Click **Next** repeatedly until you get the **Finish** button, and then click **Finish**.
16. After the installation has completed click **Save** directly to the master configuration.
17. In the side panel, open **Applications > Application Types > WebSphere enterprise applications**, and click **jrules-ssp-WAS8** or **jrules-ssp-WAS85**.
18. Click **Manage Modules**.
19. Click **Scenario Service Provider**.
20. Under General Properties, for **Class loader order** select **Classes loaded with local class loader first (parent last)** and click **OK**.
21. Click **Decision Runner**. Under **General Properties**, for **Class loader order**, select **Classes loaded with local class loader first (parent last)** and click **OK**.
22. Click **OK** again and then click **Save** to save directly to the master configuration.
23. In the side panel, open **Applications > Application Types > WebSphere enterprise applications**.
24. 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 4: Checking the availability of the testing and simulation services

You use URLs to check the availability of the Decision Runner and Scenario Service Provider (SSP) applications.

The URLs display different information on each application. Follow these steps to use the URLs:

1. To check the Decision Runner application:
 - a. Enter the URL `http://<host>:<port>/DecisionRunner` in a web browser.
 - b. Log in to the application in one of the Rule Execution Server roles.
The application displays a home page that contains version and patch-level information.
2. To check the SSP application:
 - a. Enter the URL `http://<host>:<port>/testing` in a web browser.
 - b. Log in to the application in one of the Rule Execution Server roles.

The SSP application displays a home page that 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 Data Access Object (DAO) factory class that is used to persist the trace into the Decision Warehouse.

Job store class

The name of the class that is used to persist the SSP 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 test suites with SSP:

- Job ID: Listed in the table when a user clicks **Run** in Decision Center.
- Created column: Records the date and time when each job is initialized.
- Status column: Shows the number of tested scenarios as compared to the total number of scenarios.
- Start time: Records the time when a resource is allocated for the job.
- Parts column: Records the number of parts in the job:
 - A job that is not run in parallel has one part.
 - A job that is run in parallel has one or more parts.
- End time: Records the time when the execution of the job is complete, that is, all the scenarios in the job have been executed, the tests have been executed (for test suites), and the KPIs have been computed (for simulations).

Note:

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 5: Using an Ant task to package the SSP archive

You package the archive for testing and simulation.

About this task

The **ssp-setup** Ant task updates the SSP archive to your specific configuration and XOM (see `ssp-setup`).

Note:

This method works on Windows and other supported distributed platforms only.

Procedure

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

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

```
<taskdef resource="res-tasks.properties"/>
```
2. Use the **ssp-setup** Ant task to update the SSP artifact.
3. Run the Ant task in one of the following ways:
 - From the command line, run 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 archive.

Step 6: Redeploying a testing and simulation archive

You might have to redeploy an archive that was deployed earlier.

Before you begin

You redeploy a testing and simulation archive in the WebSphere Integrated Solutions Console. Log in to the console as explained in “Before you start: Opening the administration console” on page 125.

About this task

You might have to redeploy the default testing and simulation archive that came with the installer or a subsequent repackaged archive. You can also check the availability of the feature by redeploying the default EAR file (`ssp.ear`). You can also redeploy the default file and use it to test your rules if you have an XML XOM. An XML XOM is included in the ruleset archive of each RuleApp, so you do not have to repackage the EAR file to include the XOM.

Procedure

To redeploy a testing and simulation archive that was deployed earlier:

1. In the side panel of the console, open **Applications** and click **WebSphere Enterprise Applications**.
2. Select the currently deployed application and click **Uninstall**.
3. Click **OK** to confirm uninstallation of the application.
4. Click **Save** to save the changes directly to the master configuration.

5. Follow “Step 3: Deploying the testing and simulation archive” on page 161 to deploy an archive.

Distributing rule testing 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 rule 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 archive is deployed on each server along with a XU connection factory and the appropriate data source definition.

About this task

You create a custom resolver to define which Rule Execution Server instances are used to run rule tests.

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 that are defined in Decision Center:

- `testingServer1: http://host1:9080/res`
- `testingServer2: http://host1:9080/res`

And your resolver can evaluate the testing URL as follows:

```
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

When you run a rule test, the execution is directed by the server that you choose.

What to do next

For a better implementation, you can set a dependency on the project that is being tested so that you can distribute the test execution according to that project.

```
ManagerBean bean = ManagerBean.getInstance() ;
IlrSession session = bean.getSession();
IlrRuleProject project = session.getWorkingBaseline().getProject();
String pname = project.getName();
```

Tuning the Decision Runner web application

You improve the performance of simulation runs in the Business console.

Before you begin

The Business console runs simulations on the Decision Runner web application. A simulation can run on one or more threads, and run a ruleset multiple times, requiring an execution unit (XU) (see Execution unit (XU)).

You can change Decision Runner and XU parameters to make simulations more efficient:

- Decision Runner: Change the maximum number of concurrent threads.
- XU: Change the connection pool size and the connection pool wait policy.

About this task

To complete this task, you must first estimate the maximum number of single-threaded simulations (**X**) and multithreaded simulations (**Y**) that might be started in parallel, and the maximum number of threads that might be used for multithreaded simulations (**Z**). Use this formula to determine the maximum number of concurrent threads: $X + (Y \times Z)$.

Procedure

1. Change the maximum number of concurrent threads in the Decision Runner on your server:

Application server	Method
<ul style="list-style-type: none">• WebSphere Application Server 8, 8.5, and 8.5.5• WebLogic 11g (10.3.6) and 12c	Use the work manager that is associated with the Decision Runner web application to set the maximum number of concurrent threads that are used by simulations. Note: To represent accurately the maximum number of concurrent threads that are used by the Decision Runner, the bounded work manager must work with only the Decision Runner. Otherwise, the number of maximum threads must be set much higher than the required amount to accommodate requests from other components.
<ul style="list-style-type: none">• Tomcat 7• JBoss 5.1.2 and 6.1	Edit the <code>THREADPOOL_MAXIMUM_SIZE</code> parameter in the web application deployment descriptor (<code>web.xml</code>) of the Decision Runner application. The default value is 10.
<ul style="list-style-type: none">• Liberty profile 8.5.5.3	You cannot set the maximum number of concurrent threads for simulations directly on the executor service that is associated with the Decision Runner web application. Executor services on Liberty use the Liberty common thread pool. If necessary, you can tune the maximum number of threads directly on the common thread pool. Note that the common thread pool is shared, and its threads are not used by the Decision Runner only.

2. Set the connection pool size for the XU. Change the size to be in line with the capacity of the server and greater than the maximum number of concurrent threads.

If you cannot set the connection pool size to be greater than the maximum number of concurrent threads, you must lower your estimate. For best results, dedicate a XU to simulations. If other applications must use the XU, try to use a number of connections greater than the maximum number of concurrent threads.

Note:

For more information about customizing the connection pool of a XU, see Rule session tuning. For WebSphere Application Server, WebLogic Server and JBoss, follow the steps for Java EE. For Tomcat and Liberty, follow the steps for Java SE.

3. Set the connection pool wait policy for the XU so that the connection pool refuses new connections immediately when the pool is full. If simulations fail, check your application server logs for the following Decision Runner error:

- GBRXU200E: The default connection manager pool is full.

If you get this message, increase the connection pool size to try to solve the problem.

Alternatively, depending on the application server, you can change the connection pool wait policy so that the connection pool is able to wait for a connection to be released to fulfill a connection request when the pool is full. However, if the wait time is set to an indefinite amount of time, it might cause some threads to hang in the system.

Example

In setting up the Decision Runner, you determine that you must be able to run at the same time 10 single-threaded simulations and 4 multithreaded simulations that have a maximum of 8 threads each. The maximum number of concurrent threads comes to $10 + (4 \times 8) = 42$.

You have a server that can handle a connection pool size of 60. The XU is shared, but the other applications take no more than 10 connections at the same time, so we have 50 connections available for simulations.

Using these parameters, you can have up to 8 more single-threaded simulations when compared to the estimated simulation load ($50 - 42 = 8$), or you can have more than 8 threads for some multithreaded simulations, at full performance and without getting failures. If you exceed these parameters, the connection pool wait policy might make new simulations and some existing running simulations fail.

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:

```
http://localhost:7001/teamserver?datasource=jdbc/serverextendedbrm.
```

The locale of the sign-in page is English by default. You can specify a locale parameter in the URL that switches the sign-in page to the required locale. For example:

```
http://localhost:<port>/teamserver?locale=es (assuming that your message files are localized).
```

If you sign in with another locale in the URL and want to change the locale afterward, click **Options** in the top banner of the Enterprise console or **Profile** in the Business console. This saves the locale and restores it the next time you sign in.

If you open Decision Center but no database exists, you automatically access the Installation Settings wizard with only the **Install** tab available.

After completing the installation, Decision Center is ready to use but does not contain a rule project. You have to publish a rule project from Rule Designer.

A diagnostics tool, available in the Configure tab of the Enterprise Console, shows a report on the status of your Decision Center configuration.

To learn more about Decision Center, see [Decision Center](#).

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England SO21 2JN

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England SO21 2JN

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or

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

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

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

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Index

A

- access rights
 - for MBean descriptors 60
- Ant tasks
 - configuring Decision Center 147
 - configuring users 153
 - creating the database schema 149
 - defining and uploading message files 152
 - execute-schema 150
 - executing the database schema 150
 - gen-create-schema 149
 - gen-drop-schema 151
 - print-config-param 154
 - remove-config-param 154
 - removing a database schema 151
 - repackage-war 156
 - repackaging the Decision Center
 - archive 156
 - set-config-param 154
 - setting configuration parameters 154
 - setting the persistence locale 154
 - setting up the environment 147
 - ssp-setup 165
 - to repackage an EAR file 73
 - upload-extensions 150
 - upload-messages 152
 - upload-roles 151
 - upload-users 153
 - uploading groups to the database 151
 - uploading the database schema extension 150
- asynchronous execution
 - pool size 163
 - support in Decision Server 73

C

- cells
 - Rule Execution Server configuration in a cluster environment 80
 - WebSphere Application Server clusters 5
- class loading sequence, changing
 - Decision Center 140
- clusters
 - configuring on WebSphere Application Server 1, 8, 9, 12, 16, 17, 19
 - on WebSphere Application Server
 - ideal topology 5
 - removing Decision Center from 22
 - removing Decision Server Rules from 14
- configuration parameters
 - adding or removing 154
 - for WebSphere Application Server clusters 8, 9, 12, 14, 16, 17, 19, 21
 - setting up 145

- configuring Decision Center
 - declaring custom groups 138
 - user registry 134
- configuring Decision Centers
 - on WebSphere Application Server
 - overview 125
- configuring Decision Validation Services 165
 - checking availability of Decision Validation Services 163
 - creating database resources 159
 - redeploying Decision Validation Services 165
- configuring Rule Execution Server 49
 - diagnostics 81
 - environments 80
 - on WebSphere Application Server 45
 - creating connection pools 52
 - creating data sources 52
 - deploying the hosted transparent decision service EAR file 71
 - federated repository 57
 - J2C authentication data 54
 - JDBC providers 51
 - mapping the admin group to the Monitor role 59
 - security 56
 - WebSphere Application Server security policies 60
- connection factories
 - for WebSphere MQ input and output queues 74
 - for WebSphere MQ input and output topics 76
- connection pool
 - creating 130
- connection pools
 - Rule Execution Server
 - WebSphere Application Server 52
- custom groups, declaring
 - Decision Center 138
- custom properties
 - Decision Center security 137
 - setting for data sources on WebSphere Application Server 54

D

- data source
 - creating 130
- data sources
 - on WebSphere Application Server
 - creating 52
 - setting custom properties on WebSphere Application Server 54
- database
 - testing connection 133
- database resources
 - for Decision Validation Services 159
- database schema
 - creating 149

- database schema (*continued*)
 - executing 150
 - for Rule Execution Server
 - choosing, using Installation Settings wizard 67
 - reviewing, using Installation Settings wizard 67
 - removing 151
 - uploading extension 150
- databases
 - creating schema
 - by running SQL scripts 68
 - from the Installation Settings wizard 65
 - permissions
 - Decision Center 127
 - testing the connection
 - WebSphere Application Server 56, 70
 - user permissions 48
- DB2
 - creating a database for Rule Execution Server configuration 49
 - creating JDBC providers
 - for WebSphere Application Server 51
- DB2 database
 - creating a database schema
 - by running SQL scripts 69
- Decision Center
 - configuring for WebSphere Application Server for Linux on System z 125
 - configuring for WebSphere Application Server on distributed platforms 125
 - configuring the database 143
 - configuring with Ant tasks 147
 - Installation Settings wizard 142
 - opening 170
 - repackaging the archive 156
 - security
 - custom properties 137
 - setting up configuration parameters 145
 - setting up groups 144
 - setting up message files 143
 - setting up persistence locale 145
 - verifying deployment 141
 - WebSphere Application Server clusters
 - removing 22
- Decision Center console
 - configuring
 - completing the configuration 141
- Decision Server Rules
 - WebSphere Application Server clusters
 - removing 14
- Decision Validation Services
 - checking availability 163
 - deploying the archive for the first time 161

- Decision Validation Services (*continued*)
 - distributing tests to multiple servers 166
 - packaging using Ant 165
 - redeploying 165
- Decision Warehouse
 - persistence setting for execution trace 48
- declaring
 - custom groups in Decision Center 138
- defining, user registry
 - Decision Center 135
- deploying
 - Decision Validation Services
 - archive 161
 - hosted transparent decision service
 - EAR file
 - on WebSphere Application Server 71
 - management EAR
 - on WebSphere Application Server 63
 - XU RAR
 - on WebSphere Application Server 61
 - deploying, Decision Center EAR
 - Decision Center 138
 - Derby database
 - creating a database schema
 - by running SQL scripts 68
 - Derby databases 49
 - creating JDBC providers
 - for WebSphere Application Server 51
 - diagnostics
 - in the Rule Execution Server console
 - running 81
 - dropping a database schema
 - See* removing

- E**
- EAR archives
 - repackaging using Ant 73
- ear, deploying
 - Decision Center 138
- environment variables
 - for Ant 147
- environments
 - Rule Execution Server
 - configuration 80
- execution unit (XU)
 - deploying the XU RAR 61

- F**
- failover
 - cluster configuration 1
 - ideal topology 5
- federated repositories
 - Rule Execution Server
 - on WebSphere Application Server 57
- federated repository, using
 - Decision Center 135

- G**
- groups
 - Rule Execution Server
 - on WebSphere Application Server 57
 - setting up 144
 - uploading to the database 151

- H**
- hosted transparent decision services
 - deploying EAR
 - on WebSphere Application Server 71

- I**
- ideal topology
 - on WebSphere Application Server 5
- input/output queues
 - creating the connection factory in WebSphere MQ 74
- input/output queues, WebSphere MQ
 - activation specification 77
- input/output topics, WebSphere MQ
 - activation specification 78
- creating the connection factory 76
- Installation Settings wizard 142
 - choosing the database schema 67
 - Configure Database 143
 - report 68
 - reviewing the database schema 67
- Rule Execution Server on WebSphere Application Server 45
 - Set Configuration Parameters 145
 - Set Persistence Locale 145
 - Set up Groups 144
 - Set up Message File 143
- to create a Rule Execution Server
 - database schema 65
- installing
 - creating a connection pool 130
 - creating a data source 129, 130
 - creating a JDBC provider 129
 - database, testing connection 133
- installing Decision Center
 - defining user registry 135
 - deploying the Decision Center EAR 138
 - restricting user permissions 127
 - security policies 136
 - using federated repository 135
- installing Rule Execution Server
 - on WebSphere Application Server
 - testing the database
 - connection 56, 70

- J**
- J2C authentication data
 - Rule Execution Server
 - on WebSphere Application Server 54

- JBoss application server
 - verifying deployment of Decision Center 141
- JDBC provider, creating 129
- JDBC providers
 - WebSphere Application Server
 - DB2 51
- JMX (Java Management Extensions)
 - limitations with Rule Execution Server
 - console instances 5

- L**
- Last Participant Support (LPS)
 - enabling in WebSphere MQ 80
- Linux on System z
 - configuring Rule Execution Server on WebSphere Application Server 45, 125

- M**
- management EAR, deploying
 - on WebSphere Application Server 63
- mapping
 - the admin group to the Monitor role, in WebSphere Application Server 59
- MBeans
 - and WebSphere MQ 73
 - descriptors for WebSphere Application Server 7.0 60
 - security policy for the Rule Execution Server console 60
- message files
 - defining and uploading 152
 - setting up 143
- message-driven rule beans
 - and WebSphere MQ
 - installation 79
 - queue activation specification 77
 - topic activation specification 78

- MQ
 - Integrating WebSphere MQ in WebSphere Application Server 73

- O**
- opening
 - Decision Center 170
 - Rule Execution Server console 66

- P**
- packaging
 - Decision Validation Services 165
 - Rule Execution Server archives using Ant 73
- parameters
 - See* configuration parameters
- permissions
 - database access 48
 - Decision Center installation 127
 - for MBean descriptors 60

- persistence
 - and Installation Settings wizard 65
 - configuring in Rule Execution Server 48
- persistence locale
 - setting 154
 - setting up 145
- Profile Management Tool 8, 9
- Profile Management Tools 12, 16, 17, 19
- profile templates
 - configure WebSphere Application Server 23
 - to configure Decision Center 16
 - to configure WebSphere Application Server 8
- properties
 - custom, setting 54
- publishing rule projects
 - from Decision Center 169

Q

- queues, WebSphere MQ
 - for request messages 75
 - for response messages 75

R

- redeploying
 - Decision Validation Services 165
- removing
 - database schema 151
- roles
 - in WebSphere Application Server 59
- Rule Execution Server
 - configuring
 - using Ant tasks 73
 - configuring for WebSphere Application Server for Linux on System z 45
 - configuring for WebSphere Application Server on distributed platforms 45
 - configuring on Rule Execution Server
 - setting custom properties 54
 - configuring on WebSphere Application Server
 - deploying the management EAR 63
 - empty database schema and credentials 49
 - diagnostics 81
 - distributing DVS tests to multiple instances 166
 - on WebSphere Application Server
 - deploying the XU RAR 61
 - setting up the persistence mode 48
 - Rule Execution Server configuration
 - creating a database schema
 - by running SQL scripts 68
 - from the Installation Settings wizard 65
 - Rule Execution Server console
 - none or more than one
 - JMX limitations 5
 - opening 66

- Rule Execution Server console *(continued)*
 - security policies
 - WebSphere Application Server 60
- rule projects
 - publishing
 - from Decision Center 169
- rule sessions
 - and WebSphere MQ integration 73

S

- Scenario Service Provider (SSP)
 - application collocated with the Rule Execution Server console 161
 - Rule Execution Server console
 - collocation with the SSP
 - application 161
 - ssp-setup Ant task 165
 - security
 - custom properties 137
 - Decision Center 137
 - enabled when configuring Decision Center 125
 - J2C authentication data 54
 - policies for the Rule Execution Server console
 - WebSphere Application Server 60
 - Rule Execution Server
 - on WebSphere Application Server 56
 - security policies
 - Decision Center 136
 - session management, changing
 - Decision Center 140
 - SQL scripts
 - to create a database schema 68
 - to remove a database schema 151
 - ssp-setup, Ant task 165

T

- testing and simulation
 - configuring
 - on WebSphere Application Server 158
 - simulation 158
 - testing 158
- topics, WebSphere MQ
 - for request messages 76
 - for response messages 77
- transaction management
 - in WebSphere Application Server, Last Participant Support 80

U

- user permissions 48
- user registry, defining
 - Decision Center 134, 135
- users
 - Rule Execution Server
 - on WebSphere Application Server 57

W

- WebLogic Server application server
 - verifying deployment of Decision Center 141
- WebSphere Application Server
 - cluster configuration 8, 9, 12, 16, 17, 19
 - removing applications 14, 21
 - clusters, configuring 1, 8, 9, 12, 14, 16, 17, 19, 21
 - configuring Decision Center 125
 - configuring Rule Execution Server 45
 - database, empty 49
 - security, Rule Execution Server 56
 - setting up the Ant tasks
 - environment 147
- WebSphere MQ
 - input queue, creating 75
 - input topic, creating 76
 - integrating in WebSphere Application Server 73
 - Last Participant Support, enabling 80
 - message-driven rule bean,
 - installing 79
 - output queue, creating 75
 - output topic, creating 77
 - queue activation specification,
 - creating 77
 - queue connection factory, creating 74
 - topic activation specification,
 - creating 78
 - topic connection factory, creating 76
- workload balancing
 - cluster configuration 1
 - ideal topology 5

X

- XU RAR, deploying
 - on WebSphere Application Server 61
- XUConnectionFactory
 - default JNDI name 61, 80



Printed in USA