



**IBM WebSphere ILOG JRules V7.1.1.1**

**SupportPac LB01 Version 1.0:**

**WebSphere ILOG JRules**

**Integration for WebSphere**

**Business Monitor**

# **Copyright**

## **Copyright notice**

**© Copyright IBM Corp. 1987, 2010**

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

---

## **Trademarks**

IBM, the IBM logo, ibm.com, WebSphere, and ILOG are trademarks or registered trademarks of International Business Machines Corp., 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*.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

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

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

## ***Table of contents***

<b>Installing.....</b>	<b>5</b>
Installation overview.....	7
Installation prerequisites.....	8
Extracting the SupportPac deliverable.....	11
Installing the development environment.....	13
Installing Rule Studio in WebSphere Integration Developer or Rational Application Developer.....	14
Installing the SupportPac integration feature.....	15
Installing the runtime environment.....	17
Setting up a WebSphere Business Monitor profile.....	18
Installing Rule Execution Server and setting up the Decision Warehouse database.....	19
Installing the monitoring service across multiple cells.....	20
Configuring WebSphere Business Monitor with JMS.....	23
<b>Overview.....</b>	<b>24</b>
<b>Sample: Decision monitoring.....</b>	<b>26</b>
<b>Creating rule projects and RuleApps.....</b>	<b>39</b>
Importing or creating rule projects.....	40
Creating RuleApps for ruleset deployment.....	41
<b>Enabling ruleset monitoring.....</b>	<b>43</b>
Adding ruleset properties.....	44

Starting the server.....	46
Deploying a RuleApp to Rule Execution Server.....	47
(Optional) Viewing the decision service WSDL file.....	48
<b>Generating a monitor model.....</b>	<b>51</b>
Monitor model and contexts.....	53
Decision event.....	55
Metrics and KPIs.....	56
Generating a monitor model from a rule project.....	58
Creating business metrics.....	60
Creating KPIs.....	62
Changing the path to the monitored ruleset.....	64
Setting the Decision Warehouse trace metric.....	66
<b>Generating Java EE monitor projects.....</b>	<b>68</b>
<b>Packaging the runtime file in the client application.....</b>	<b>69</b>
SupportPac runtime files.....	70
Packaging a Java SE or POJO client application.....	71
Repackaging the JRules HTDS, EJB3, and MDB archives using Ant.....	73
Setting up the environment.....	74
Repackaging the HTDS archive.....	75
Repackaging the EJB3 archive.....	77
Repackaging the MDB archive.....	79
Packaging the rule session JAR into the client application.....	81
<b>Deploying the monitoring artifacts.....</b>	<b>82</b>
<b>Opening Business Space.....</b>	<b>83</b>
<b>Customizing the trace.....</b>	<b>84</b>
<b>Using a customized trace DAO.....</b>	<b>86</b>
<b>Troubleshooting.....</b>	<b>87</b>
<b>Index.....</b>	<b>89</b>

# ***Installing***

To install the SupportPac LB01 Version 1.0: WebSphere® ILOG® JRules Integration for WebSphere Business Monitor, you must first extract the contents of the SupportPac ZIP file, and install the Eclipse feature in Rule Studio. Then, you install the runtime environment by installing Rule Execution Server on WebSphere Application Server.

## **In this section**

### **Installation overview**

After installing all the required products and features, the development and testing environment should contain Rule Studio in WebSphere Integration Developer or Rational Application Developer, the SupportPac integration feature, and Rule Execution Server installed on the monitor profile.

### **Installation prerequisites**

To install the JRules Integration for WebSphere Business Monitor SupportPac, you must make sure that you have installed the required products and the correct product versions.

### **Extracting the SupportPac deliverable**

Extract the contents of the SupportPac ZIP file in your JRules installation.

### **Installing the development environment**

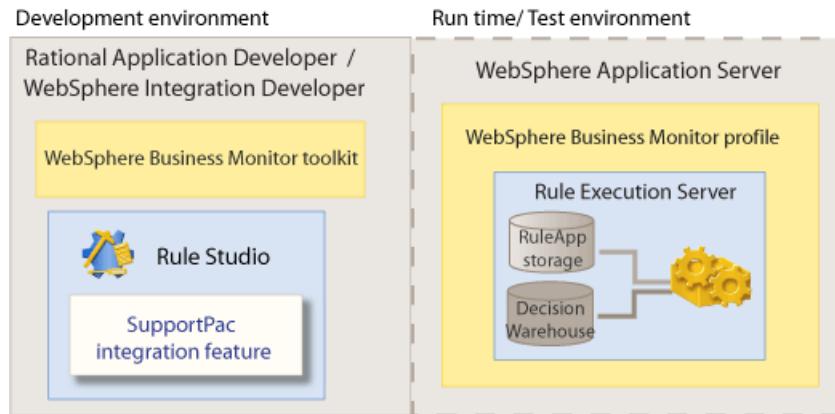
Install Rule Studio in WebSphere Integration Developer or Rational Application Developer, and then install the SupportPac integration feature.

### **Installing the runtime environment**

To deploy and test the monitor artifacts, you must have a monitor profile on WebSphere Application Server. In a single-server topology, you then install Rule Execution Server in the WebSphere Business Monitor profile.

## Installation overview

The following figure illustrates the development and testing environment **after** installing Rule Studio, Rule Execution Server, and the SupportPac feature.



### Development

This environment is based on WebSphere Integration Developer or Rational Application Developer, and includes:

- ◆ WebSphere Business Monitor Development Toolkit that includes both the development environment and the test environment, see *Installation prerequisites*.
- ◆ Rule Studio installed on top on WebSphere Integration Developer or Rational Application Developer, see *Installing Rule Studio in WebSphere Integration Developer or Rational Application Developer*.
- ◆ The SupportPac integration feature, see *Installing the SupportPac integration feature*.

### Run time / Test

Rule Execution Server is installed in the WebSphere Business Monitor profile on WebSphere Application Server, see *Installing Rule Execution Server and setting up the Decision Warehouse database*.

### See also

- Installation prerequisites*  
*Extracting the SupportPac deliverable*  
*Installing the development environment*  
*Installing the runtime environment*

---

## Installation prerequisites

A hardware and software test environment is commonly set up to run tests in an effort to eliminate unit-level and regression bugs during development.

A production environment hosts actual or real data, as opposed to test data, and is available on a publicly accessible network or server.

It is important to set up a test environment that resembles your production environment as closely as possible.

---

## Development and testing environment

To develop monitoring artifacts and test them in a runtime environment, this SupportPac requires that you have installed the following products:

- ◆ **WebSphere ILOG JRules BRMS V7.1.1.1**, or later V7.1.1 fix packs, and the Java EE add-on for WebSphere Application Server.
- ◆ **WebSphere Integration Developer V7.0.0.2**, or **Rational Application Developer V7.5**
- ◆ **WebSphere Business Monitor Development Toolkit V7.0.0.2**.

If you are using Rational Application Developer, you must install the complete WebSphere Business Monitor Development Toolkit. If you are using WebSphere Integration Developer, the development environment is already included, but you must install the monitor test environment.

The WebSphere Business Monitor Development Toolkit includes:

- A monitor **development environment**: that is already included in WebSphere Integration Developer, but that you must install if you are working in Rational Application Developer.

For more information on how to install the development toolkit, see the section “Installing and removing the toolkit > Installing the development toolkit into WebSphere Integration Developer”, and “Installing and removing the toolkit > Installing the toolkit into Rational Application Developer”, in the *WebSphere Business Monitor Development Toolkit information center*.

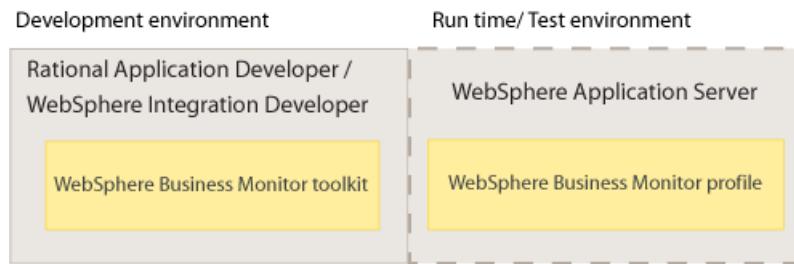
- A monitor **test environment**: that includes a WebSphere Business Monitor profile on WebSphere application Server, and Business Space.

For more information on how to install the test environment, see the section “Installing and removing the toolkit > Installing the Monitor test environment”, and “Installing and removing the toolkit > Development topology” in the *WebSphere Business Monitor Development Toolkit information center*.

**Important:** If you have installed WebSphere Business Monitor development toolkit V7.0 in Rational Application Developer V7.5, you may have difficulties upgrading to the WebSphere Business Monitor Toolkit V7.0.0.2 fix pack. For more information on how to solve this problem, see the technote *WebSphere Business Monitor development toolkit V7.0 cannot be upgraded*.

**Tip:** To check the version of WebSphere Integration Developer, Rational Application Developer, or WebSphere Business Monitor Development Toolkit that you have installed, start the IBM Installation Manager, and on the **File** menu click **View Installed Packages**.

The following figure illustrates the environment that you must have **before** you install the SupportPac.



## Production environment

To monitor the artifacts in a production environment, this SupportPac requires that you have installed the following products:

### ◆ WebSphere Business Monitor V7.0.0.2

For more information on how to install WebSphere Business Monitor V7.0.0.2, see the section “Installing WebSphere Business Monitor” in the *WebSphere Business Monitor information center*

### ◆ WebSphere Application Server V7.0.0.7

For more information on how to install WebSphere Application Server V7.0.0.7, see the *WebSphere Application Server Version information center*

### ◆ WebSphere ILOG JRules BRMS V7.1.1.1 or later V7.1.1 fix packs, and the Java EE add-on for WebSphere Application Server.

For more information on how to install JRules 7.1.1.1, see the “Installing” section in the *WebSphere ILOG JRules BRMS Version information center*.

## See also

*Installation overview*

*Installing the development environment*

*Installing the runtime environment*

---

## Extracting the SupportPac deliverable

The SupportPac ZIP file contains the required files to enable the design of the monitoring model and to view the metrics and Key Performance Indicators (KPIs) in WebSphere Business Monitor.

You must extract the contents of the SupportPac ZIP file at the root of your JRules installation in <JRules\_InstallDir>, typically C:/Program Files/IBM/WebSphere/ILOGJRules711.

The ZIP file contains the following:

- ◆ /SupportPacs/SupportPacLB01/bin/setup-wbm.xml: The Ant task to repackage the HTDS, EJB3 or MDB rule session with the SupportPac runtime files.
- ◆ /SupportPacs/SupportPacLB01/lib/jrules-wbm-runtime.jar: The runtime extension file to package in the client application.
- ◆ /SupportPacs/SupportPacLB01/lib/dvs-dw.properties: The runtime properties file to package in the client application.
- ◆ /SupportPacs/SupportPacLB01/license/: The license files for the SupportPac.
- ◆ /SupportPacs/SupportPacLB01/eclipse/: The JRules integration for WebSphere Business Monitor feature to install into Rule Studio.
- ◆ /SupportPacs/SupportPacLB01/samples/decisionmonitoring/: The sample files to run the sample.
- ◆ /SupportPacs/SupportPacLB01/readme.html: The readme file for the SupportPac.

After extracting the ZIP file, the contents must be in: <JRules\_InstallDir>/SupportPacs/SupportPacLB01/.

### See also

*Installation prerequisites*

*Installation overview*

*Installing the development environment*

*Installing the runtime environment*



---

# Installing the development environment

Install Rule Studio in WebSphere Integration Developer or Rational Application Developer, and then install the SupportPac integration feature.

## In this section

### **Installing Rule Studio in WebSphere Integration Developer or Rational Application Developer**

Before installing the SupportPac integration feature, you must install Rule Studio into your installation of WebSphere Integration Developer or Rational Application Developer.

### **Installing the SupportPac integration feature**

Install the JRules integration for WebSphere Business Monitor feature to enable the development part of your monitoring application.

## Related sections

*Installation prerequisites*

*Extracting the SupportPac deliverable*

*Installation overview*

*Installing the runtime environment*

---

## Installing Rule Studio in WebSphere Integration Developer or Rational Application Developer

After installing WebSphere ILOG JRules, you must install Rule Studio in WebSphere Integration Developer or Rational Application Developer to use WebSphere Business Monitor.

For more information on how to install Rule Studio in WebSphere Integration Developer or Rational Application Developer, see the section “WebSphere ILOG JRules BRMS 7.1 > Installing > Installing Rule Studio in IBM Eclipse-based products” in the *WebSphere ILOG JRules BRMS information center*.

### See also

*Installing the SupportPac integration feature*

---

## Installing the SupportPac integration feature

To generate a monitor model from a rule project, you must install the SupportPac integration feature in WebSphere Integration Developer or Rational Application Developer.

Make sure that your installation complies with the requirements listed in *Installation prerequisites*.

### To install the SupportPac feature:

1. Start WebSphere Integration Developer or Rational Application Developer.
2. Switch to the **Java** perspective. On the **Window** menu, click **Open Perspective > Java**.
3. On the **Help** menu, click **Software Updates**.
4. In the Software Updates and Add-ons dialog, click the **Available Software** tab.
5. Click the **Add Site** button, and then click **Local**.
6. In the Browse for Folder dialog, navigate to the `<JRules_InstallDir>/SupportPacs/SupportPacLB01/eclipse` directory, click **OK**, and then click **OK** again in the Add Site dialog.

The IBM WebSphere ILOG JRules Integration for WebSphere Business Monitor site appears in the Available Software tab.

7. Expand `<JRules_InstallDir>/SupportPacs/SupportPacLB01/eclipse`, select the check box next to **IBM WebSphere ILOG JRules Integration for WebSphere Business Monitor**, and click **Install**.  
Eclipse calculates the dependencies. This might take a few seconds.
8. In the Install dialog, click **Next**.
9. Review the terms of the license, select **I accept the terms of the license agreement**, and then click **Finish**.
10. If prompted to restart Eclipse, click **Yes**.
11. After Eclipse has restarted, click **Help > Software Updates**.
12. Click the **Installed Software** tab, and check that **IBM WebSphere ILOG JRules Integration for WebSphere Business Monitor** is in the list.

### See also

*Installing Rule Studio in WebSphere Integration Developer or Rational Application Developer*



---

# Installing the runtime environment

To deploy and test the monitor artifacts, you must have a monitor profile on WebSphere Application Server. In a single-server topology, you then install Rule Execution Server in the WebSphere Business Monitor profile.

## In this section

### **Setting up a WebSphere Business Monitor profile**

To monitor a ruleset using WebSphere Business Monitor, you must have a WebSphere Business Monitor profile on WebSphere Application Server.

### **Installing Rule Execution Server and setting up the Decision Warehouse database**

You must install Rule Execution Server on WebSphere Application Server, and set up the Decision Warehouse database.

### **Installing the monitoring service across multiple cells**

Depending on your environment topology, you might have to connect an existing Rule Execution Server profile to a WebSphere Business Monitor profile.

### **Configuring WebSphere Business Monitor with JMS**

The dvs-dw.properties file provides the JNDI properties that you must define if you have to customize the default behavior.

## Related sections

*Installation prerequisites*

*Extracting the SupportPac deliverable*

*Installation overview*

*Installing the development environment*

---

## Setting up a WebSphere Business Monitor profile

To prepare the runtime environment, you must set up a WebSphere Business Monitor profile on WebSphere Application Server. The profile includes all of the files that the server processes in the runtime environment.

You can use the profile created when installing the Monitor test environment or create a monitor profile in WebSphere Business Monitor.

For more information about the profiles created as part of the test environment installation, see the sections “Installing the Monitor test environment”, “Adding a new server to the test environment”, and “Resetting the Monitor test environment” in the *WebSphere Business Monitor Development Toolkit information center*.

For more information on how to create monitor profiles in WebSphere Business Monitor, see the sections “Profiles” and “Creating and augmenting profiles”, in the *WebSphere Business Monitor information center*.

If you are not using the Monitor test environment or if you augmented an existing profile, you must enable the recording of events.

### To enable events recording:

1. Click **Applications > Monitor Services > Recorded Events Management > Enable/Disable Events Record**.
2. In the **Select** column, select the event service that you want to record events for.
3. Click **Enable Events Record**.

For more information, see Administering your monitoring environment > Managing events > Managing recorded events > Enable and disable event recording, in the *WebSphere Business Monitor information center*.

### See also

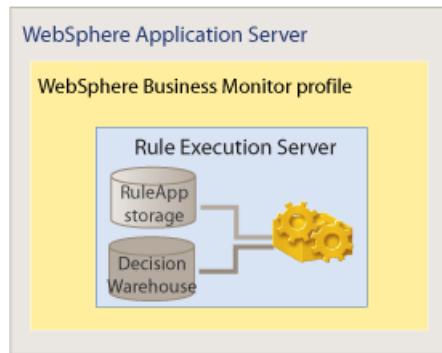
*Installing Rule Execution Server and setting up the Decision Warehouse database*

## Installing Rule Execution Server and setting up the Decision Warehouse database

To enable ruleset execution, and store the ruleset execution traces, you must install Rule Execution Server on WebSphere Application Server, and set up the access to the Decision Warehouse database.

In a single-server topology, Rule Execution Server and the monitoring components are installed on the same physical server. This is ideal for development test environments, and simple deployments.

The following figure illustrates the installation of Rule Execution Server on top of a WebSphere Business Monitor profile on WebSphere Application Server.



For more information on how to install Rule Execution Server, and set up the Decision Warehouse, see the *WebSphere ILOG JRules BRMS information center*:

- ◆ WebSphere ILOG JRules BRMS V7.1: Java EE add-ons > Installing on WebSphere Application Server > Installing Rule Execution Server
- ◆ WebSphere ILOG JRules BRMS V7.1 > Rule Studio > Testing, validating, and auditing rules > Configuring and customizing Decision Warehouse

### See also

*Setting up a WebSphere Business Monitor profile*

---

## Installing the monitoring service across multiple cells

In a production environment, you can install the monitoring service in a cross-cell topology, one cell hosting the Rule Execution Server, the other cell hosting a WebSphere Business Monitor server.

**Important:** Rule sessions running in a cross-cell topology are only supported in a non-transactional mode.

To connect an existing Rule Execution Server profile to a WebSphere Business Monitor profile, you must configure the remote service integration bus.

### To install the monitoring service across multiple cells:

1. Get the WebSphere Business Monitor configRemoteMonitorBus utility.

See the section Reference > Commands and scripts > Configuration commands > Queue-based bus configuration across multiple cells, in the *WebSphere Business Monitor information center*.

2. Configure the script.

The following example shows how to configure the script to connect a Rule Execution Server node to a WebSphere Business Monitor node.

```
SECURE_CONFIGURATION=true

LOCAL_WAS_HOST=localhost
LOCAL_WAS_PORT=8880
LOCAL_WAS_MESSAGING_ENGINE=qnode.server1-MONITOR.qcell.Bus
LOCAL_WAS_USERID=admin
LOCAL_WAS_PASSWORD=admin

REMOTE_WAS_HOST=localhost
REMOTE_WAS_PORT=8881
REMOTE_WAS_USERID=websphere
REMOTE_WAS_PASSWORD=websphere

REMOTE_WAS_ME_NODE=resNode
REMOTE_WAS_ME_SERVER=server2
REMOTE_WAS_ME_STORE=default

REMOTE_WAS_BUS_USERID=admin
REMOTE_WAS_BUS_PASSWORD=admin

LINK_USERID=admin
LINK_USERID_REMOTE_PASSWORD=admin
LINK_USERID_LOCAL_PASSWORD=admin
```

3. Create a JNDI connection to the remote queue.
  - a. Log on to the administration console of the Rule Execution Server profile.
  - b. Click **Resources > JMS > Queue connection factories**.
  - c. Select the **Node=<nodename>, Server=<servername>** scope.
  - d. Click **New**.
  - e. Select the provider, and then click **OK**.
  - f. Specify the details for the new Queue connection factory, for example:
    - ◆ **Name:** MonitorEventEmitterCF
    - ◆ **JNDI name:** jms/MonitorEventEmitter/QueueConnFactory
    - ◆ **Bus Name:** MONITOR.<res\_cell\_name>.bus
  - g. Click **Apply**, and then click **Save**.
  - h. Click **Resources > JMS > Queues**.
  - i. Select the **Node=<nodename>, Server=<servername>** scope.
  - j. Click **New**.
  - k. Select the provider, and then click **OK**.
  - l. Specify the details for the new Queue, for example:
    - ◆ **Name:** MonitorEventEmitterQueueProxy
    - ◆ **JNDI name:** jms/MonitorEventEmitter/Queue
    - ◆ **Bus Name:** MONITOR.qcell.bus
    - ◆ **Queue name:** select **other, please specify**, and enter MonitorEventEmitterQueue.
  - m. Click **Apply**, and then click **Save**.
  - n. Restart the Rule Execution Server profile.

To enable the security between the WebSphere Business Monitor and the Rule Execution Server cells, see the *WebSphere Business Monitor information center*

- ◆ Installing WebSphere Business Monitor > Configuring how to receive events > Receiving events from CEI > Using the JMS queues > Across multiple cells
- ◆ Securing your environment > Configuring server security > Configuring server-to-server SSL in multiple-cell environments

## **See also**

*Configuring WebSphere Business Monitor with JMS*  
*Packaging the runtime file in the client application*

---

## Configuring WebSphere Business Monitor with JMS

Events are emitted in WebSphere Business Monitor by Java Message Service (JMS) messages.

The `dvs-dw.properties` file provides the JNDI properties that you must define if you have to customize the default behavior. At run time, the properties are read by WebSphere Application Server and are used to create the initial context and connect to the WebSphere Business Monitor custom resources with the values specified.

### JNDI initial context creation properties

By default, the JNDI local node initial context is used to find the resources that send the message containing the trace. If you deploy Rule Execution Server and WebSphere Business Monitor in two separate nodes, you must specify the initial context. You can use JNDI properties such as `java.naming.provider.url` and `java.naming.factory.initial`.

For more information, search for the section “Developing applications that use JNDI”, in the *WebSphere Application Server Express (Distributed platforms and Windows) information center*.

### JNDI names for JMS destination resources

The emitter send the message containing the trace to a WebSphere Business Monitor JMS queue. By default, the connection to the JMS queue is set as `wbm.queueconnectionfactory.jndi.name=jms/MonitorEventEmitter/QueueConnFactory` and `wbm.queue.jndi.name=jms/MonitorEventEmitter/Queue`.

To create a connection to a custom WebSphere Business Monitor JMS queue, specify the name of your resource. For example:

- ◆ `wbm.queueconnectionfactory.jndi.name=<myCustomQueueConnFactory>`
- ◆ `wbm.queue.jndi.name=<myCustomQueue>`

### See also

*Installing the monitoring service across multiple cells*  
*Using a customized trace DAO*

---

## Overview

Business Rule Management System (BRMS) offers a controlled way for business users and analysts to make operational business changes. JRules allows business users and analysts to view their business policy implemented in a natural language. They can also make changes to the implementation themselves, and test business rules before deploying them to production. This process improves business quality and can shorten a product's time-to-market.

To respond quickly to market changes, business users and analysts must be able to monitor business performance and analyze Key Performance Indicators (KPIs). They also need to make sure that the changes they have made or are about to make give the expected results.

Using JRules, business performance can be assessed by recording how decisions are made with the current set of rules. A trace allows business users to analyze the transactional data that is used during the execution of that decision. Transactional data contains information on the rules that applied in the transactions.

In addition to these features, this SupportPac provides real-time monitoring by connecting JRules and WebSphere Business Monitor. Business users can use WebSphere Business Monitor to view KPIs and receive alerts when unusual situations are detected. They use business dashboards in Business Space to monitor performance, and view the KPIs in the form of gauges and tables.

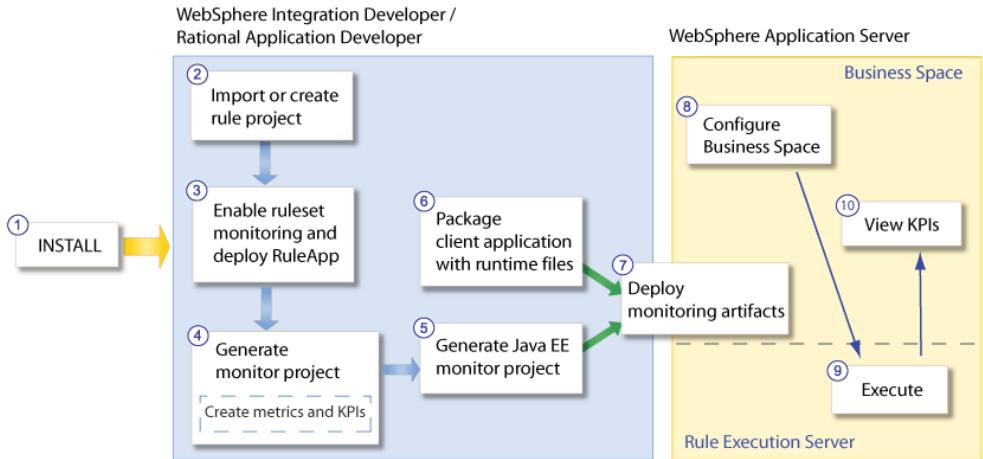
Developers can generate monitor artifacts from rule projects. When creating a monitor model from a rule project, developers can select predefined execution metrics and KPIs such as the execution duration, the percentage of rules fired, the tasks executed, and so on.

Business users and analysts can use this SupportPac to compare two versions of the same ruleset deployed in a production environment. One version of the ruleset is used for the real production application, and the other version is deployed for monitoring purposes. Business users can then monitor the performance of the rulesets in WebSphere Business Monitor. They can identify which of the two ruleset performs best. This is similar to the Champion/Challenger approach, where an environment can be configured to push a small percentage of the transactions through a challenger, while pushing the majority through the champion. Results from the different approaches can be compared and measured over time. If a challenger does better than the champion, it can be made the new champion and the process of identifying and testing new challengers is repeated to continually improve the decision.

---

## Technical overview

The following tasks illustrate the workflow to create and deploy a monitor model generated from a rule project.



1. Install the SupportPac in WebSphere Integration Developer or Rational Application Developer.
2. Import or create a rule project with an XSD XOM, and create a RuleApp project to deploy the ruleset.
3. Enable monitoring of the ruleset, and deploy the RuleApp.
4. Generate the monitor model from the rule project.  
You can select the predefined execution metrics and KPIs, and create business metrics and KPIs.
5. Generate a Java EE monitor application project.
6. Create a rules-enabled client application with execution code to retrieve information from the monitoring-enabled ruleset. Then, package the SupportPac runtime files into the client application.
7. Deploy the client application and the Java EE monitor application to WebSphere Application Server.
8. Create a space in Business Space and add widgets to view the KPIs.
9. Execute the client application.
10. View the KPIs results in Business Space.

## See also

*Sample: Decision monitoring*

---

## Sample: Decision monitoring

This sample shows how to monitor a decision event, how to create business metrics and Key Performance Indicators (KPIs), and how to view KPIs in Business Space.

This sample illustrates the following key features:

- ◆ Creating a monitor model from a rule project.
- ◆ Creating business metrics and KPIs based on ruleset parameters and on the execution trace.
- ◆ Generating a Java EE application project that contains the monitor model.
- ◆ Deploying the Java EE monitor application and the client application.
- ◆ Sending the decision event to WebSphere Business Monitor.
- ◆ Accessing the decision trace directly from Business Space.
- ◆ Viewing KPIs in Business Space.

---

### Prerequisites

To install and run this sample, you must make sure that you have the required environment:

- ◆ Rule Studio and the SupportPac integration feature installed in WebSphere Integration Developer or Rational Application Developer
- ◆ WebSphere Business Monitor Development Toolkit installed in WebSphere Integration Developer or Rational Application Developer:
  - Development toolkit
  - Monitor test environment: WebSphere Business Monitor profile on WebSphere Application Server, and Rule Execution Server installed on the WebSphere Business Monitor profile.

For more information on the required environment, see *Installing*.

This sample uses Firefox as browser to enable session sharing between Business Space and the Rule Execution Server Console.

You must make sure that you have a user name and password to access Business Space and the WebSphere Application Server administrative console. This sample documentation uses admin / admin as user name and password. This sample explains how to give the correct rights to the user to access the Rule Execution Server Console.

---

## Setting up the sample

Before running the sample, you must first import the sample projects, and then edit the decision trace metric.

### Importing this sample

Before running this sample, you must import the sample projects into a new workspace in WebSphere Integration Developer or Rational Application Developer.

#### To import the sample projects:

1. Click **File > Import**, and select **General > Existing Projects into Workspace**, and then click **Next**.
2. Click **Browse**, and select `<JRules_InstallDir>/SupportPacs/SupportPacLB01/samples/decisionmonitoring`
3. Make sure the following projects are selected:
  - ◆ **loan-appli**: the loan web client application project
  - ◆ **loan-monitor**: the monitor project
  - ◆ **loan-rules**: the rule project
  - ◆ **loan-ruleapp**: the RuleApp project
  - ◆ **loan-web**: the loan web module project

**Note:** If you are using Rational Application Developer, the `loan-web` project contains a warning that you can ignore.

4. Select the **Copy project into workspace** check box, and click **Finish**.

### Installing the sample

In this sample, you access Rule Execution Server to view the Decision Warehouse trace from Business Space, after executing your ruleset. To enable the link to the trace, you must make sure that the URL in the Decision Trace metric is correct and modify it if necessary.

#### To edit the Decision Trace metric:

1. Switch to the **Business Monitoring** perspective. On the **Window** menu, click **Open Perspective > Other**, select **Business monitoring** and click **OK**.
2. In the Project Explorer, expand `loan-monitor/Monitor Models`, and double-click `MonitorLoanDecision`.

The MonitorLoanDecision monitor model was generated from the loan-rules rule project, and then customized. For more information on the customization of the monitor model, see *Details*.

3. In the model tree, expand **Eligibility**.
4. Select the **Decision Trace** metric.
5. Under Metric Value Expression, click the Expression and click the (...) button.
6. Make sure that the URL to Rule Execution Server is correct, and modify it if necessary.

The host and the port number should be the same as the server.

The following filter specifies the host as `localhost` and the port as `9080`.

```
fn:concat('<a href="javascript: void(0)"  
onclick="javascript:window.open ("http://localhost:9080/res/  
protected/traceTree.jsf?executionId=', Decision_ID, ''',  
"Trace")">', 'View Trace</a>')
```

7. Click **OK**.
8. Save the changes.

---

## Running this sample

To publish the monitor model to WebSphere Business Monitor, you must first generate Java EE monitor projects from the model.

### To generate monitor JEE projects:

1. Make sure that you are in the **Business Monitoring** perspective.
2. In the Project Explorer, right-click the MonitorLoanDecision monitor model, and click **Generate Monitor JEE Projects**.
3. Click **Finish**.

This might take a few seconds to complete.

4. Switch to the **Java EE** perspective. On the **Window** menu, click **Open Perspective > Other**, select **Java EE** and click **OK**.

In the Enterprise Explorer, you can see that three projects are generated:

- ◆ **MonitorLoanDecisionApplication**: The Java EE application project that contains the model logic and moderator projects. You deploy this project to the server in the next steps.

- ◆ `MonitorLoanDecisionModelLogic`: The model logic project that is included in the Java EE application project.
- ◆ `MonitorLoanDecisionModerator`: The moderator project that is included in the Java EE application project.

For more information on these projects, see *Generating Java EE monitor projects*.

Before deploying the Java EE monitor application and the client application, you must start the server.

#### To start the server:

1. Open the **Servers** view.

**Note:** If you cannot see the Servers view, on the **Window** menu, click **Show View > Servers**.

2. Right-click the **WebSphere Business Monitor Server on WebSphere Application Server**, and click **Start**.

If you do not have the correct server instance, create a new server instance:

- a. Right-click in the Servers view, and click **New > Server**.
- b. Select **WebSphere Business Monitor Server on WebSphere Application Server**.
- c. Make sure the **Server runtime environment** is correct: **WebSphere Business Monitor Server on WebSphere Application Server**.

Click **Next**.

- d. In the **Profile name** field, make sure that the profile selected is the WebSphere Business Monitor profile (`qmwas`).
- e. Enter a User ID and Password, for example:

- ◆ **User ID**: admin
- ◆ **Password**: admin

- f. Click **Finish**.

Now, you must make sure that you have the correct administrator rights to access the Rule Execution Server Console from Business Space.

#### To give the administrator rights to open the Rule Execution Server Console:

1. In the Servers view, right-click the **WebSphere Business Monitor Server**.
2. Click **Administration > Run administrative console**.

3. If prompted to accept the security certificate, click **Yes**.
4. Enter the user ID and password, for example:
  - ◆ **User ID:** admin
  - ◆ **Password:** admin
5. Click **Log in**.
6. Click **Users and Groups > Manage Users**.
7. In the table that lists the users, click the **admin** user ID.
8. Click the **Groups** tab.
9. Click **Add**.
10. Click **Search**.

11. Select `resAdministrators`, and then click **Add**.

The user `admin` was added to the `resAdministrators` group.

12. Click **Close** to return to the user.

The `resAdministrators` group is now in the list of groups that the user belongs to. You will now be able to login to the Rule Execution Server Console with the `admin` user ID and password.

13. Restart the server.

Now, you must deploy the ruleset to Rule Execution Server.

**To deploy the ruleset to Rule Execution Server:**

1. In the Enterprise Explorer, expand `loan-ruleapp`, and double-click `archive.xml` to open it.
2. In the RuleApp editor, under Deployment, click **Deploy a RuleApp to one or more Rule Execution Servers**.
3. Make sure **Increment RuleApp major version** is selected, and then click **Next**.
4. Select **Create a temporary Rule Execution Server configuration**:
  - a. In the **URL** field, change the URL to reflect the server host and port number.
  - b. Enter the administrator login and password:
    - ◆ **Login:** admin
    - ◆ **Password:** admin
5. Click **Finish**.

The Console view indicates the versions of the RuleApp and the ruleset deployed.

You can now sign in to Rule Execution Server to check that the RuleApp was successfully deployed.

**To view the RuleApp and ruleset in Rule Execution Server:**

1. Open the Rule Execution Server Console: <http://localhost:9080/res>

**Important:** Make sure that you use the same port number as the server.

2. Enter the username and password
  - ◆ **Username:** admin
  - ◆ **Password:** admin
3. Click **Sign In**.
4. Click the **Explorer** tab.
5. Expand **RuleApps** in the Navigator panel to display the RuleApps View.
6. Expand **/loanruleapp/1.0**, and then select **/loanrules/1.0** to view the ruleset.
7. Do not sign out of the Rule Execution Server Console.

After deploying the ruleset to Rule Execution Server, you can deploy the Java EE monitor application, as well as your client application.

**To deploy the Java EE monitor application and the client application:**

1. In the Servers view, right-click the server and click **Add and Remove Projects**.
2. Under Available projects, select `loan-appli` and `MonitorLoanDecisionApplication`, click **Add**, and then click **Finish**.

In the Servers view, the status of the server shows that the projects are being published to WebSphere Application Server. Wait until both projects are displayed under the server and started.

To view the KPIs, you open Business Space and import the space created for the sample.

**To import the sample space into Business Space:**

1. Right-click the server, and click **Business Space**.  
If prompted to accept the security certificate, click **Yes**.
2. Enter the administrator User ID and password, for example:

◆ **User ID:** admin

◆ **Password:** admin

**3. Click Login.**

Business Space might take a few seconds to open.

**4. In Business Space, click Manage Spaces.**

**5. Click Import Space.**

**6.** To import the sample space, click **Browse** and select <JRules\_InstallDir>/SupportPacs/SupportPacLB01/samples/decisionmonitoring/Decision+Monitoring+Sample.zip, and then click **OK**.

**7. Click the Decision Monitoring Sample space to open it.**

The space shows an empty diagram and empty pie charts that represent KPIs from the monitor model. The diagrams and charts do not show any value yet because you have not executed the ruleset.

**8. Keep Business Space open.**

To execute the rules, you use the web application designed for this sample.

**To execute the decision monitoring web application:**

1. Open a new browser window and enter the URL for the web application: `http://localhost:9080/loan-web`.

**Important:** Make sure that you use the same host and port number as the server.

2. In the web application, enter the number of scenarios that you want to execute, for example 10.

The application executes the ruleset on data that is randomly generated, for the number of scenarios that you have entered.

**3. Click Execute.**

The application shows both the input and the output values for each execution. You can see the values for the loan and borrower input parameters and the results for each execution. The results show whether the loan was approved, the yearly repayment for the loan, and the number of rules executed. The metrics and KPIs defined in the monitor model are based on these values.

The KPIs in Business Space have now been updated with the values of the execution.

## To view the KPI results in Business Space:

1. Switch back to Business Space.

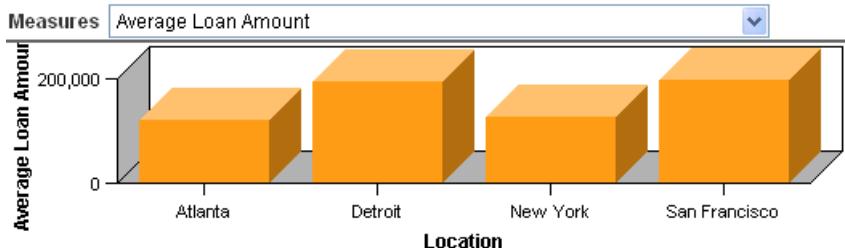
The KPI gauges and tables show the values of the executions. The Instances table lists the executions that you triggered by clicking the **Execute** button in the web application. The KPI values and gauges are updated after each execution.

2. To update the dimension widget, refresh the Business Space page in the browser, or click the arrow ▾ in the Dimension title bar, and then click **Refresh**.
3. In the **Measures** list, select the **Average Loan Amount** KPI.
4. Right-click **All locations**, and click **Drill Down**.

The dimension now shows the country level and displays the average loan amount for each country.

5. Right-click a country, and click **Drill Down** again to get to the city level.

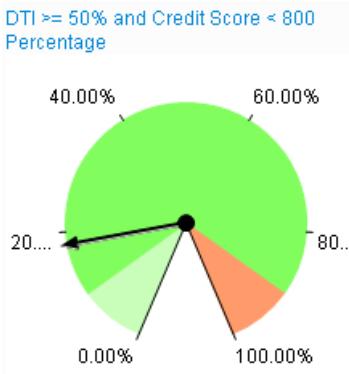
The following figure shows the Average Loan Amount for Atlanta, Detroit, New York, and San Francisco.



6. Take a look at the gauge graphs to view the business KPI values.

The gauge graph shows the KPI value as a black arrow, and the ranges as colored regions. The black arrow indicates the current value of the KPI. The arrow is in the medium range and shows the percentage value corresponds to what is expected.

The following figure shows the percentage of borrowers who have a debt-to-income ratio (DTI) of 50% or more, and a credit score of under 800, is medium.



## 7. Take a look at the KPI values in the KPI table.

The following figure highlights the KPI values for Paris, France.

State	City	CreationTime	Decision Trace	List of Rules Fired	Interest Rate	Loan Amount
US	New York	October 22, 2010 2:50:31 AM	<a href="#">View Trace</a>		0.04	195,000
France	Paris	October 22, 2010 2:50:54 AM	<a href="#">View Trace</a>	eligibility.repayment and score 7, eligibility.minimum income	0.05	335,000
Australia	Camberra	October 22, 2010 2:50:54 AM	<a href="#">View Trace</a>		0.03	390,000

## 8. Click **View Trace** to view the decision trace.

The decision trace opens in the Rule Execution Server Console.

**Note:** If you cannot access the trace the first time, try clicking the View Trace again. If your Rule Execution Server session has expired, you might have to login again.

## Details

This section describes the implementation details for this sample.

### XOM

In this sample, the rule project uses an XSD XOM named `loan.xsd`. You can only create business metrics and KPIs if the rule project used to generate the monitor model contains an XSD XOM.

### Ruleset properties

Specific properties must be defined in the ruleset to enable ruleset monitoring.

### To view the ruleset properties in the RuleApp project:

1. Switch to the **Rule** perspective. On the **Window** menu, click **Open Perspective > Other**, select **Rule** and click **OK**.
2. Expand `loan-ruleapp`, and double-click `archive.xml`.
3. Click the **Ruleset Archives** tab, and select the `loan-rules` under Ruleset Archives.
4. Under Ruleset Properties, have a look at the properties enabled:
  - ◆ `monitoring.events.enabled`
  - ◆ `monitoring.enabled`
  - ◆ `monitoring.dw.enabled`: This property is enabled for the purpose of this scenario. However, it can affect performance in a production environment, so you should only use it for debugging purposes. If the property is disabled, the execution trace is no longer available in Rule Execution Server but you can still monitor the ruleset.

For more information on the event emission and Decision Warehouse properties, see *Customizing the trace*.

## Monitor project

The `loan-monitor` monitor project was generated from the `loan-rules` rule project.

The `loan-monitor` monitor project has been customized for the purpose of this sample:

- ◆ The context name is **Eligibility**.
- ◆ New metrics, KPIs, and dimensions have been added:
  - New business metrics and KPIs related to the input parameters of the rule project. For example, the `City` metric is based on the `input.borrower/city` parameter value.
  - New execution metrics and KPIs related to the execution of the rules. For example, the `DTI >= 50%` and `Credit Score < 800` metric is based on which rule is fired in a decision table.
  - New measures and three levels of dimension (State, City, Interest) based on business KPIs.

## Business metrics and KPIs

Several business metrics and KPIs are defined in the monitor model. The business metrics and KPIs are based on the event parts of the decision event. The event parts consist of the decision trace and of the ruleset parameters.

### To view the business metric details:

1. Switch to the **Business Monitoring** perspective. On the **Window** menu, click **Open Perspective > Other**, select **Business monitoring** and click **OK**.
2. In the Project Explorer, expand `loan-monitor/Monitor Models`, and double-click `MonitorLoanDecision`.
3. In the model tree, expand `Eligibility`.
4. Select the **DTI >= 50% and Credit Score < 800** metric.
5. Under Metric Value Expression, click the Expression and click the (...) button.

The following expression checks if the list of rules fired contains the rule that corresponds to the seventh row of the `eligibility.repayment` and `score` decision table:

```
if (fn:contains(wbm:serialize(DecisionEvent/decisionTrace/
trace:rulesFired), 'repayment and score 7') = fn:true())
then 1 else 0
```

#### To view the business KPI details:

1. In the Business Monitoring perspective, click the **KPI Model** tab.
2. Expand **Eligibility KPI Context**.
3. Select **DTI >= 50% and Credit Score < 800 Percentage**.

In the KPI Definition, under KPI Value, look at the KPI calculation.

This KPI is calculated based on the `DTI >= 50%` and `Credit Score < 800` and on the `Number of Eligibility Decisions` metrics.

## Dimension model

#### To view the dimension details:

1. In the Business Monitoring perspective, click the **Dimensional Model** tab.
2. Expand **MiniloanLoanDecision > Eligibility Cube**.

The dimensional model contains several measures created from existing metrics.

3. Select the **Average Loan Amount** measure.

The **Source metric** field indicates that this measure is based on the `Average Loan Amount` metric.

4. Expand **Location**.

There are three dimension level details: State, City, Interest. They are based on the business metrics defined on the Borrower input parameter.

## Source files

The execution code for the sample web module is located in `loan-web/src/loanweb`:

Class	Description
<code>DataFactory.java</code>	This class contains the functions to create the input loan and the input borrower from static and random data.
<code>LoanBean.java</code>	This class contains the bean to execute the ruleset. See the <code>validateWithJRules</code> function to see how to execute a ruleset on a POJO rule session.
<code>XmlBorrower.java</code>	This class is used to serialize in XML and from XML, and to display a borrower in JSF.
<code>XmlLoan.java</code>	This class is used to serialize in XML and from XML, and to display a loan in JSF.
<code>Result.java</code>	This class is used in the JSF page to display the results in a table.

The JSF page of the sample web module is described in `loan-web/WebContent/loan.jsp`. The JSF page uses the `loanBean` bean to execute the ruleset.

The `dvs-dw.properties` runtime file is in `loan-web/src/`. For more information on the `dvs-dw.properties` file, see *SupportPac runtime files*, *Configuring WebSphere Business Monitor with JMS*, and *Using a customized trace DAO*.

### See also

*Generating Java EE monitor projects*

*Metrics and KPIs*

*Packaging the runtime file in the client application*

*Enabling ruleset monitoring*



# ***Creating rule projects and RuleApps***

To be able to generate a monitor model for your ruleset, you must first import or create a rule project in Rule Studio. Then, you create a RuleApp project to deploy the ruleset to Rule Execution Server.

## **In this section**

### **Importing or creating rule projects**

Import an existing rule project into your workspace, or create a new rule project.

### **Creating RuleApps for ruleset deployment**

To deploy the rules to Rule Execution Server, you must first create a RuleApp project.

---

## Importing or creating rule projects

To generate a monitor model from a rule project, you must import a rule project into your workspace or create a new rule project.

To create business metrics and KPIs, your rule project must contain or reference an XSD Execution Object Model (XOM). If the rule project contains a Java XOM, you can only create execution metrics and KPIs.

**Note:** This SupportPac only supports ruleset parameters of XSD type. The SupportPac does not support ruleset parameters of Java type including primitive types.

- ◆ XSD XOM files that are available within the workspace are not copied into the generated Monitor Project. The XSD XOM files are referenced in the generated Monitor Model.
- ◆ XSD XOM files that are not available within the workspace, for example if they are stored on the file system, are copied into the generated Monitor Project and referenced in the generated Monitor Model.

### To create a new rule project:

1. Switch to the **Rule** perspective. On the **Window** menu, click **Open Perspective > Other**, select **Rule**, and then click **OK**.
2. Click **File > New > Rule Project**.

**Note:** If you already have a rule project, you can import it into your workspace. Click **File > Import**, select **General > Existing Projects into Workspace**, and then click **Next**. Navigate to your rule project, and then click **Finish**.

For more information on how to create rule projects, see the section *WebSphere ILOG JRules BRMS V7.1 > Rule Studio > Creating rule projects*, in the *WebSphere ILOG JRules BRMS Information Center*.

---

## Creating RuleApps for ruleset deployment

In order to deploy a ruleset to Rule Execution Server you need to create a RuleApp project and add a ruleset archive to it. Ruleset archives can be added to a RuleApp project from either a ruleset archive JAR file, or a rule project.

In Rule Studio, a RuleApp is handled as an Eclipse project, and is deployed to a running Rule Execution Server by way of the RuleApp project.

### To create a RuleApp project:

1. On the **File** menu, click **New Project**.
2. Expand **Rule Studio** and select **RuleApp Project**.
3. Click **Next**.
4. On the RuleApp Project page, enter the name of the project, then click **Next**.  
Use only a–z, A–Z, 0–9, and underscore (\_). For the first character, 0–9 is not allowed.
5. Optionally, add ruleset archives to the project using the wizard, as follows:

- a. On the **Rule Projects** tab, click **Add** and select a rule project from the list displayed.

If you want to use an extractor to filter rules contained in the project, click the Extractor radio button and select the required extractor from the list displayed.

The list you see depends on the rule project you select. If no ruleset extractor is specified in the rule project, the list is empty. By default, a rule project has no ruleset extractor defined.

- b. Click **OK**.
- c. Click the **Ruleset Archives** tab.
- d. Click **Add**, select the archives you want to add to the RuleApp project, then click **OK**.
- e. Click **Finish**.

The RuleApp project is created and is added to the list of projects. The RuleApp project editor opens automatically.

For more information on rulesets and RuleApps, see the section *WebSphere ILOG JRules BRMS V7.1 > Rule Execution Server > Deploying rulesets*, in the *WebSphere ILOG JRules BRMS Information Center*.



# ***Enabling ruleset monitoring***

To enable ruleset monitoring, you add properties to the ruleset and you deploy the ruleset to Rule Execution Server. After deploying the ruleset, you can access Rule Execution Server to view the ruleset path.

## **In this section**

### **Adding ruleset properties**

To enable the monitoring of your ruleset, you must set specific properties in your ruleset.

### **Starting the server**

Before deploying the RuleApp, you must start the server.

### **Deploying a RuleApp to Rule Execution Server**

In Rule Studio, deploy the RuleApp to Rule Execution Server.

### **(Optional) Viewing the decision service WSDL file**

In the Rule Execution Server Console, view the RuleApps and rulesets deployed and get the WSDL file.

## **Related sections**

*Sample: Decision monitoring*

*Creating rule projects and RuleApps*

## Adding ruleset properties

After creating your RuleApp project, you add the following properties to the ruleset.

Property	Description
monitoring.enabled	Set this ruleset property to <code>true</code> to enable monitoring of ruleset execution for Decision Warehouse.
monitoring.events.enabled	Set this property to <code>true</code> to enable the ruleset emission of trace events.
monitoring.dw.enabled	This property is <b>optional</b> . This property stores decisions in the Decision Warehouse, you should only use it for debugging purposes as it can affect performance in a production environment. If the property is disabled, the execution trace is not available in Rule Execution Server. Set this property to <code>true</code> to enable Decision Warehouse persistence.

### To add monitoring ruleset properties:

1. Make sure that you are in the **Rule** perspective. On the **Window** menu, click **Open Perspective > Other**, select **Rule**, and then click **OK**.
2. In the Rule Explorer, expand the RuleApp project, then double-click the RuleApp descriptor (`archive.xml` file).
3. Click the **Ruleset Archives** tab.
4. Under Ruleset Archives, select the ruleset archive to deploy.
5. Under Ruleset Properties, click **New**.
6. Select the predefined property **monitoring.enabled** from the drop-down menu, type `true` in the **Value** field, and then click **OK**.
7. To enter another property, click **New**.
8. In the **Name** field, type `monitoring.events.enabled`, and in the **Value** field, type `true`, and then click **OK**.
9. Optional: Add the property `monitoring.dw.enabled` as described in the previous step.

The monitoring properties are added to the ruleset. You can now deploy the RuleApp to Rule Execution Server.

**See also**

*Customizing the trace*

---

## Starting the server

After adding the properties to the ruleset, you must start the server to deploy the RuleApp that contains the ruleset to Rule Execution Server.

When you installed the Monitor test environment, a WebSphere Business Monitor server was created in Rational Application Developer or WebSphere Integration Developer. The test server is displayed in the Servers view.

If you do not have a test server in the Servers view, see the sections “Preparing the test environment” and “Adding a new server to the test environment” in the *WebSphere Business Monitor Development Toolkit information center*.

### To start the server in WebSphere Integration Developer or Rational Application Developer:

1. On the **Window** menu, click **Show view > Other**, select **Server > Servers**, and then click **OK**.
2. In the Server view, select WebSphere Business Monitor Server on WebSphere Application Server, and click the **Start the server** button .

### See also

*Deploying a RuleApp to Rule Execution Server*

*Sample: Decision monitoring*

*Deploying the monitoring artifacts*

---

## Deploying a RuleApp to Rule Execution Server

After you have created a RuleApp project and configured an application server, you can use the Rule Studio deployment wizard to deploy a RuleApp to Rule Execution Server.

### To deploy a RuleApp to Rule Execution Server:

1. In the RuleApp editor, click the **Overview** tab.

If you have closed the RuleApp editor, you can also expand the RuleApp project, then double-click the RuleApp descriptor (archive.xml file).

2. In the RuleApp editor, under **Deployment**, click **Deploy**.
3. Select the deployment type, and then click **Next**.
4. Select the required Rule Execution Server(s), or create a temporary server, then click **Finish**.

**Note:** For more information, see the section WebSphere ILOG JRules BRMS 7.1 > Rule Execution Server > Deploying and exporting RuleApps, in the *WebSphere ILOG JRules BRMS information center*.

### See also

*Starting the server*

---

## (Optional) Viewing the decision service WSDL file

After deploying the RuleApp to Rule Execution Server, sign in to the Rule Execution Server Console to view the RuleApps and rulesets that you deployed. You can retrieve the Web Services Description Language (WSDL) file and view the ruleset path of the deployed ruleset. You can then use the ruleset path to create your monitor project.

### To open the Rule Execution Server Console and retrieve the WSDL file:

1. Make sure that the server is started.
2. Open the Rule Execution Server Console by starting a web browser and typing: `http://localhost:9080/res`.

**Important:** Make sure that the host and port server are the same as your test server.

The console opens and displays a prompt for you to sign in.

3. Enter the username and password of the Rule Execution Server role.  
Both depend on the role you have been assigned.
4. Click **Sign In**.
5. Click the **Explorer** tab.
6. Click **RuleApps** in the Navigator panel to display the RuleApps View.
7. Click the name of the RuleApp that contains the ruleset.
8. In the RuleApp View, click the relevant ruleset.
9. On the **WSDL** line of the Ruleset View, click one of the **Get WSDL** links.
  1. Click **Get WSDL for this ruleset version** to generate the WSDL from the canonical ruleset path.
  2. Click **Get WSDL for the latest ruleset version** to generate the WSDL from a ruleset path that does not include the ruleset version number.

In the WSDL file, you can view the ruleset path and the ruleset parameters. You will then use the ruleset path when generating the monitor model, to define which ruleset to use for the decision event.

### See also

*Starting the server*

*Deploying a RuleApp to Rule Execution Server*



# ***Generating a monitor model***

Generate a monitor project and monitor model from a rule project, and define metrics and KPIs.

## **In this section**

### **Monitor model and contexts**

Monitor models contain monitoring contexts and associated event definitions.

### **Decision event**

When you create a monitor model from a rule project, the SupportPac generates a decision event that receives the results of the ruleset execution.

### **Metrics and KPIs**

Metrics and KPI measures are based on the execution results stored in the decision trace.

### **Generating a monitor model from a rule project**

Generate a monitor project and a monitor model from the rule project in your workspace.

### **Creating business metrics**

To monitor the performance of a ruleset, you can create business metrics and KPIs that are based on the execution trace and on the ruleset parameter values.

### **Creating KPIs**

Create Key performance indicators (KPIs) using a metric and an aggregation function, or create KPIs based on other KPIs or user-defined XPath functions.

### **Changing the path to the monitored ruleset**

To compare two versions of the same ruleset, you can create a new context file from the same application and modify the ruleset path to point to a different ruleset version.

### **Setting the Decision Warehouse trace metric**

You must edit the decision trace metric to point to the correct URL for Rule Execution Server.

### **Related sections**

*Generating Java EE monitor projects*

*Sample: Decision monitoring*

*Creating rule projects and RuleApps*

---

## Monitor model and contexts

A monitoring-enabled application generates events that are sent to WebSphere Business Monitor in XML format. The application needs a monitor model to describe the events to WebSphere Business Monitor.

Based on the monitor model, WebSphere Business Monitor:

- ◆ Receives and extracts required information from the events.
- ◆ Updates the metrics defined in the model.
- ◆ Displays the information on a dashboard in a way that meets business needs.

Monitor models are represented as XML documents that specify how values are extracted from events at run time. A monitor model also contains the information about how the values are collected, combined, and stored to be represented on a dashboard.

The information is extracted from the data carried by inbound events, and is held in metrics, counters, and stopwatches, which represent the business measures that a monitoring context collects.

The monitor model is a container and consists of the following parts:

- ◆ Monitor details model: monitoring contexts and their associated metrics, events, keys, stopwatches, counters, and triggers. The monitor details model holds most of the monitor model information.
- ◆ KPI model: KPI contexts and KPIs
- ◆ Dimensional model: cubes, dimensions, and measures
- ◆ Visual model: diagrams associated with monitoring contexts and KPI contexts
- ◆ Event model: references to event definitions

A monitor model contains monitoring contexts, which define the associated events. A monitoring context also defines the information to be collected at run time. The monitoring context receives events, and updates performance measures such as metrics and KPIs based on information extracted from these events.

All communication into and out of a monitoring context is handled by events. Monitoring context instances subscribe to events, update their state based on information in the events, and emit events to report a business situation.

The SupportPac generates a monitoring context with the associated predefined execution metrics, and the decision event. If you select predefined KPI measures when creating the monitor model, the SupportPac also generates a KPI context.

You can create more monitoring contexts if you need them. For example, you can create a monitoring context that contains different metrics and KPIs, or another ruleset path to monitor a different version of the ruleset.

For more information on monitor models and monitoring contexts, see the *WebSphere Business Monitor information center*, or the *WebSphere Business Monitor Development Toolkit information center*.

## See also

*Decision event*

*Metrics and KPIs*

*Generating a monitor model from a rule project*

---

## Decision event

When you create a monitor model from a rule project, it generates an event called DecisionEvent. The decision event is an inbound event, that is, an event that the monitoring context subscribes to. The decision event receives the results of the ruleset execution from Rule Execution Server. The execution results are presented as a decision trace.

The filter condition of the decision event defines the path to the targeted ruleset.

The decision event consists of event parts that describe the structure of the event. The event parts are defined by XML schemas.

The event parts that make up the decision event are based on:

◆ **The decision trace**

The decision trace event part (`decisionTrace`) is always in the event.

◆ **The ruleset parameters**

The event parts based on ruleset parameters are only available if you generated the monitor model from a rule project with an XSD XOM. If you used a rule project with a Java XOM, you only get the decision trace as event part.

The ruleset parameter event parts are presented as follows:

- If a ruleset parameter named `borrower` is of type `IN`, the corresponding event part is `input.borrower`.
- If a ruleset parameter named `loan` is of type `IN_OUT`, two event parts are created:  
`input.loan` and `output.loan`.

**Important:** This SupportPac only supports ruleset parameters of XSD type. The SupportPac does not support ruleset parameters of Java type including primitive types.

### See also

*Monitor model and contexts*

*Metrics and KPIs*

*Generating a monitor model from a rule project*

---

## Metrics and KPIs

A metric is a business performance measurement. A metric can be used to define the calculation for a key performance indicator (KPI), which measures performance against a business objective. Metrics and KPI measures are based on the execution results stored in the decision trace. The metric value expression is evaluated when the decision event that it references arrives.

Key performance indicators (KPIs) measure the improvement or deterioration in the performance of an activity. KPIs are aggregations of values. They can be based on a metric and an aggregation function such as average, maximum, minimum, sum, and so on. A KPI measure can also be an expression that is a calculation based on other KPIs or user-defined XML Path Language (XPath) functions.

A KPI can have a target to achieve and a set of ranges to measure the business objectives.

### Predefined metrics and KPIs

When you generate a monitor model from a rule project, you define the monitoring templates to associate with the monitoring context.

The SupportPac provides predefined execution metrics and execution KPIs as monitoring templates. These predefined metrics and KPIs reference the decision event, and are based on the **execution** results.

The SupportPac provides execution metrics and KPIs, for example:

- ◆ the total number of rules fired
- ◆ the total number of tasks executed
- ◆ the percentage of rules fired
- ◆ the average number of rules fired
- ◆ the average processing time
- ◆ the link to the decision trace from Decision Warehouse
- ◆ the ruleset version (executed ruleset path)

The predefined KPI templates do not include a target and a range. You can define the target and ranges for the KPI according to your business objectives.

### Custom business metrics and KPIs

When you generate a monitor model from a rule project that contains an XSD XOM, the ruleset parameters are added as event parts of the decision event.

Business metrics are based on the **ruleset parameters**.

To create a business metric such as the “loan approved”, you use the output value of the `loan` ruleset parameter (`output.loan`). You can also create a business metric based on an existing metric, for example, you can create a metric “approved loan amount” that uses the metric “loan approved”.

After creating business metrics, you can create KPIs based on the metrics you have created. For example, you can create a KPI “Total Approved Loan Amount” that uses the metric “approved loan amount”, and that calculates the sum of the loan amounts for the loans that have been approved.

## See also

*Creating business metrics*

*Setting the Decision Warehouse trace metric*

*Generating a monitor model from a rule project*

*Sample: Decision monitoring*

---

## Generating a monitor model from a rule project

After creating or importing a rule project into your workspace, you can generate a monitor project and a monitor model from the rule project.

### To generate a monitor model from a rule project:

1. Make sure that you are in the **Rule** perspective (**Window > Open Perspective > Other**, select **Rule** and click **OK**).
2. In the Rule Explorer, right-click the rule project, and click **Generate Monitor Model**.
3. On the Ruleset Path definition page, select the type of ruleset path for the decision event:
  - ◆ **requested ruleset path**: If you select the requested ruleset path, only the name of the RuleApp and the name of the ruleset is requested. If you don't add any version information, the monitor model uses the latest ruleset version deployed.
  - ◆ **executed ruleset path**: If you select the executed ruleset path, the RuleApp version and the ruleset version are required. The monitor model uses only the ruleset path with the specific RuleApp and ruleset versions.
4. Enter the name of the RuleApp, and if required, the version number of the RuleApp.
5. Enter the name of the ruleset, and if required, the version number of the ruleset.  
The ruleset path is displayed in the Ruleset Path field.
6. Click **Next**.
7. In the **Target monitor project** field, select a monitor project or click the **New Project** button:
  1. Enter a name for the monitor project, and then click **Next**.
  2. Click **Finish**.
8. In the **Target monitor model name** field, enter a name for the monitor model, and then click **Next**.
9. On the Choose What to Monitor page, in the Event Source section, select the monitor context.
10. In the Monitoring Templates tab, select the predefined execution metrics and KPIs that you want to use for monitoring.  
To view the description of a metric or KPI, select the name of the metric or KPI.
11. To select or view the event to monitor, click the **Emitted Events** tab.

If you have selected predefined execution metrics and KPIs in the Monitoring Templates tab, the **DecisionEvent** is automatically selected. The decision event is now enabled for emission in the application that you will be monitoring.

You cannot remove an event if you have chosen a template that requires that event.

12. Click **Next**.
13. On the Monitor Model page, keep the default implementation settings, and click **Next**.
14. Click **Finish**.
15. If prompted to open the Business Monitoring perspective, click **Yes**.

**Note:** You can also click **Window > Open Perspective > Other**, select **Business Monitoring**, and then click **OK**.

The monitor model is displayed in the Project Explorer.

The monitor model contains the monitoring context, and its associated event definitions and metrics:

- ◆ On the **Monitor Details Model** tab, you can view the monitoring context and the predefined metrics that you selected, as well as the decision event.
- ◆ On the **KPI Model** tab, you can view the KPI context that was created if you selected predefined KPIs, as well as the predefined KPIs that you selected.

**Tip:** If you want to add other predefined metrics and KPIs after generating the monitor model and the monitoring context, you can do so from the generated monitoring context.

1. On the Monitor Details Model tab, select the monitoring context in the model tree.
2. Under Application Element, in the **Templates** list, click **Add**.
3. Select the predefined metrics and KPIs and then click **OK**.

## See also

*Monitor model and contexts*

*Metrics and KPIs*

---

## Creating business metrics

To monitor the performance of a ruleset, you can use the ruleset parameter to create business metrics that you can then use to create KPIs.

**Important:** To create business metrics and KPIs, the rule project must contain an XSD XOM.

### To create a business metric:

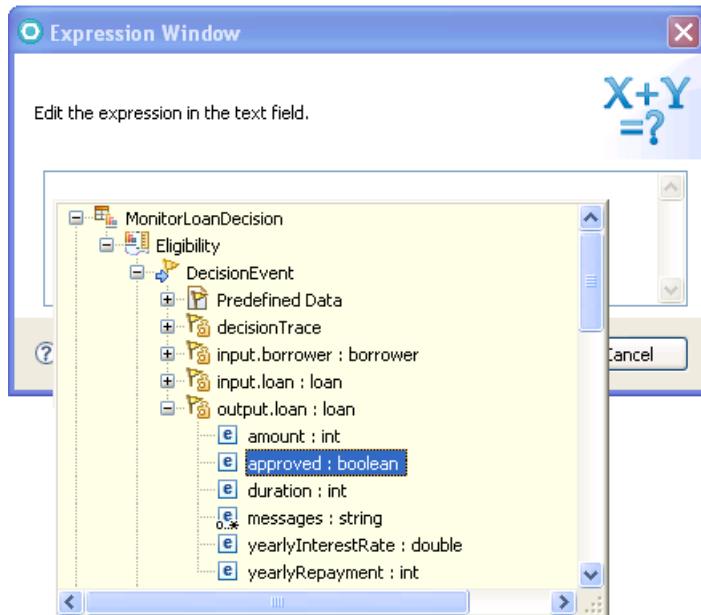
1. Make sure that you are in the Business Monitoring perspective.
2. In the Monitor Details Model tab, select **DecisionEvent** in the tree.
3. Under Event Type details, look at the event parts:
  - ◆ Ruleset execution trace: `decisionTrace`
  - ◆ Ruleset input parameter, for example: `input.borrower`
  - ◆ Ruleset input and output parameters, for example: `input.loan` and `output.loan`
4. In the model tree, right-click the monitoring context and click **New > Metric**.
5. Enter a name and an ID for the metric.
6. In the **Type** field, enter the type of the metric, and then click **OK**.

The new metric is added to the monitoring context.
7. Enter the default value if required.
8. Under Metric Value Expressions, click **Add**.

A new line to enter the value expression is displayed in the table.
9. Click in the **Expression column**, and click the (...) button.
10. In the Expression Window, press **Ctrl+Spacebar** to trigger the auto completion menu.
11. Expand the monitor model, expand the decision event and navigate to the value you want to add.

For example, the value could be `DecisionEvent/output.loan/approved : boolean`.

The following figure shows the approved value in the completion menu.



**12.** Double-click the value, and then click **OK**.

The new expression is displayed in the Expression column, for example  
DecisionEvent/output.loan/approved.

**13.** Save the changes.

You can now use this metric to create other metrics or KPIs.

For more information on metrics and how to create them, see the *WebSphere Business Monitor Development Toolkit information center*, Creating monitor models > Defining monitor details models > Defining metrics.

## See also

*Sample: Decision monitoring*

*Metrics and KPIs*

*Creating KPIs*

---

## Creating KPIs

Key performance indicators (KPIs) measure the improvement or deterioration in the performance of an activity. KPIs are aggregations of values that you create using a metric (execution or business) and an aggregation function. A KPI can also be a calculation based on other KPIs or user-defined XPath functions. To measure business objectives, you can define a target to achieve, a set of ranges, or both.

### To create a KPI based on a metric:

1. Make sure that you are in the Business Monitoring perspective.
2. If you do not have a KPI context already created, or if you want to create a new KPI context:
  - a. Click the **KPI Model** tab.
  - b. In the model tree, right-click the KPI model, and click **New > KPI Context**.
  - c. Enter a name and an ID for the KPI context, and then click **OK**.

The new KPI context is added to the model tree.

3. Right-click the KPI context, and click **New > KPI**.
4. Enter a name and an ID for the KPI, and then click **OK**.
5. Optional: Fill in the KPI details, and the KPI target and ranges.  
For example, the ranges could be high, medium, and low.
6. In the KPI Definition, under KPI value, select **Base this KPI on a metric and an aggregation function**.
7. Under KPI details, click **Browse** to select the monitoring context for the metric that you want to use.
8. In the **Metric** field, click **Browse**, select the metric that you want to use, and then click **OK**.
9. In the **Aggregation function** field, select an aggregation function.
10. Save the changes.

For more information on KPIs and how to create them, see the *WebSphere Business Monitor Development Toolkit information center*, Creating monitor models > Defining KPI models > Defining key performance indicators (KPIs).

### See also

*Creating business metrics*  
*Metrics and KPIs*

*Sample: Decision monitoring*

---

## Changing the path to the monitored ruleset

After you have generated a monitor model based on an application that you want to monitor, some of the elements in the monitor model are read-only, such as the event filter condition that contains the ruleset path. To monitor and compare two versions of the same ruleset, you can add a new context file to the model from the same application, and change the path to the monitored ruleset.

### To create a new context file from the same application:

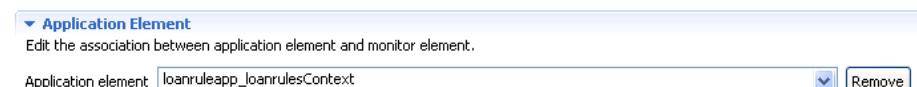
1. Make sure that you are in the Business Monitoring perspective.
2. In Monitor Details Model tab, right-click the monitor model, and click **New > Create from Application**.
3. Under Event Source, select the monitor context and select the monitor templates (metrics and KPIs), click **Next**, and then click **Next** again.
4. Click **Finish**.

The monitor model now contains another context file with the predefined metrics and KPIs that you selected.

The new context file might have the same name as the first context file, but it has a different ID.

5. In the model tree, select the new monitor context.
6. Under Application Element, click the **Remove** button next to the **Application element** field.

The following figure shows a dependency on an application element called `loanruleapp_loanrulesContext`.



7. Click **OK** to confirm that you want to remove the dependency.

This removes the dependency between the application element and the new monitoring context file. In the decision event, the filter condition that contains the ruleset path is no longer read-only, and you can now edit it.
8. In the tree, select **DecisionEvent**.
9. Under **Filter Condition**, edit the ruleset path and change it to point to the ruleset that you want to monitor.

The following figure shows a user changing the ruleset path to /loanruleapp/1.0/loanrules/1.2.

**▼ Filter Condition**

Define a condition based on the event attributes to identify whether to accept an event of this type.

```
fn:exists(DecisionEvent/decisionTrace/trace:requestedRulesetPath) and  
fn:contains(DecisionEvent/decisionTrace/trace:requestedRulesetPath,'/loanruleapp/1.0/loanrules/1.2')
```

- 10.** Save the changes.

---

## Setting the Decision Warehouse trace metric

If you have enabled the `monitoring.dw.enabled.ruleset` property, Decision Warehouse stores the decision traces resulting from ruleset executions.

When defining a monitoring context, you can select the predefined Decision Warehouse Trace Metric. This metric is a link to the decision trace of Decision Warehouse.

To view the decision trace in Business Space, you must edit the decision trace metric to modify the URL to Rule Execution Server.

### To edit the decision trace metric:

1. Make sure that you are in the Business Monitoring perspective.
2. In Monitor Details Model tab, in the model tree, select the decision trace metric.
3. Under Metric Value Expression, click the Expression and click the (...) button.
4. Change the host and the port number as required.

The following href attribute of the HTML `<a>` tag specifies `localhost` as the host and `9080` as the port.

```
<a href="javascript: void(0)" onclick="javascript:window.open ("http://localhost:9080/res/protected/traceTree.jsp?executionId=' + Decision_ID + ', ' + 'Trace")"> 'View Trace</a>')
```

5. Click **OK** to save the changes.

**Important:** The way to access the decision trace from Business Space depends on your browser and on the browser settings.

- ◆ **Firefox** shares the sessions between the different browser instances. Therefore, when you want to view the Decision Warehouse trace from Business Space, you can access the trace directly without having to login to Rule Execution Server again. However, you must first grant the user the correct rights to access Rule Execution Server.

To give the administrator rights to open the Rule Execution Console:

1. In the Servers view, right-click the **WebSphere Business Monitor Server**.
2. Click **Administration > Run administrative console**.
3. If prompted to accept the security certificate, click **Yes**.
4. Enter the user ID and password, for example:

- User ID: admin
  - Password: admin
5. Click **Log in**.
  6. Click **Users and Groups > Manage Users**.
  7. Click **Search**.
  8. In the table that shows the results of the search, click the `admin` user ID.
  9. Click the **Groups** tab.
  10. Search for the groups
  11. Add `resAdministrators` to the groups.
  12. Restart the server.
- ◆ **Internet Explorer** uses a unique session ID for each browser instance so there is no session sharing. Therefore, the user must first login to Rule Execution Server before accessing the trace in Decision Warehouse.

For more information on viewing trace information in Decision Warehouse, see the *WebSphere ILOG JRules BRMS information center*, Rule Execution Server online help > Monitoring and managing Decision Warehouse > Filtering and viewing trace information

## See also

*Sample: Decision monitoring  
Metrics and KPIs  
Enabling ruleset monitoring*

---

## Generating Java EE monitor projects

After creating the monitor model, you must deploy and test the model within the WebSphere Monitor test environment. You can then view the results of your test in Business Space to make sure that the metrics and KPIs behave as expected.

To deploy and test the monitor model you must generate Java EE monitor projects from the monitor model.

### To generate a Java EE monitor project:

1. Make sure you are in the Business Monitor perspective. On the **Window** menu, click **Open Perspective > Other**, select **Business monitoring**, and then click **OK**.
2. In the Project Explorer, right-click the monitor model (`.mm`), and click **Generate Monitor JEE Projects**.
3. Click **Finish**.

This might take a few seconds to complete.

4. Switch to the **Java EE** perspective. On the Window menu, click **Open Perspective > Other**, select **Java EE** and click **OK**.

In the Enterprise Explorer, you can see that three projects were generated:

- ◆ The **Model Logic project**: An EJB project that contains the event processing logic of the monitor model.
- ◆ The **Moderator project**: An EJB project that contains the event consumption logic of the monitor model.
- ◆ The **Java EE Application project**: An enterprise application (EAR) project that contains the two EJB projects. The Java EE Application project is the project that you deploy to the server to test the monitor model.

For more information on the Java EE monitor projects, see the section *Testing monitor models > Generating Monitor JEE projects* in the *WebSphere Business Monitor Development Toolkit information center*.

### See also

- Generating a monitor model from a rule project*
- Deploying the monitoring artifacts*
- Sample: Decision monitoring*

# ***Packaging the runtime file in the client application***

Before deploying your client application, you must include the SupportPac runtime files in the client application. For Java SE and POJO rule sessions, you include the files in the application class path. For HTDS, EJB3, and MDB client applications, you can use an Ant task to perform the packaging of the rule session archives with the SupportPac runtime files.

## **In this section**

### **SupportPac runtime files**

This SupportPac provides two runtime files to monitor performance in WebSphere Business Monitor.

### **Packaging a Java SE or POJO client application**

Package the SupportPac JAR file and the properties file in the class path of your client application.

### **Repackaging the JRules HTDS, EJB3, and MDB archives using Ant**

To automate the packaging of HTDS, EJB3, and MDB archives, add the Ant task to your build and deployment process. The Ant tasks repackage the JRules archives with the SupportPac files.

## **Related sections**

*Deploying the monitoring artifacts*

---

## SupportPac runtime files

The SupportPac JAR file and the `dvs-dw.properties` file are available in the `SupportPacLB01` directory:

- ◆ <JRules\_InstallDir>/SupportPacs/SupportPacLB01/lib/  
`jrules-wbm-runtime.jar`
- ◆ <JRules\_InstallDir>/SupportPacs/SupportPacLB01/lib/  
`dvs-dw.properties`

### **jrules-wbm-runtime.jar**

The runtime JAR file to monitor the decision event in WebSphere Business Monitor.

### **dvs-dw.properties**

The runtime file that describes the JNDI environment properties, and the Queue Connection Factory.

For more information on the `dvs-dw.properties` file, see *Using a customized trace DAO* and *Configuring WebSphere Business Monitor with JMS*.

### **See also**

*Packaging a Java SE or POJO client application*

*Repackaging the JRules HTDS, EJB3, and MDB archives using Ant*

---

## Packaging a Java SE or POJO client application

For Java SE and POJO, you must repackage your client application so that it includes the SupportPac JAR file and the `dvs-dw.properties` file in the application class path.

### To include the SupportPac files:

1. Copy `jrules-wbm-runtime.jar` to the `WEB-INF/lib` folder of your client application.
2. Copy `dvs-dw.properties` to the `WEB-INF/classes` folder of your client application.

### See also

*Rewriting the JRules HTDS, EJB3, and MDB archives using Ant*



---

# Repackaging the JRules HTDS, EJB3, and MDB archives using Ant

To automate the packaging of HTDS, EJB3, and MDB archives, add the Ant task to your build and deployment process. The Ant tasks repackage the JRules archives with the SupportPac files.

## In this section

### **Setting up the environment**

To use the Ant task to repackage the HTDS, EJB3, and MDB archives, you must make sure that you have the correct environment.

### **Repackaging the HTDS archive**

Use the `wbm-htds` Ant task to repackage the HTDS EAR to include the SupportPac runtime files.

### **Repackaging the EJB3 archive**

Use the `wbm-ejb3` Ant task to repackage the EJB3 rule session to include the SupportPac runtime files.

### **Repackaging the MDB archive**

Use the `wbm-mdb` Ant task to repackage the MDB JAR to include the SupportPac runtime files.

### **Packaging the rule session JAR into the client application**

After repackaging the rule session JAR files with the SupportPac files, you must package the rule session JAR files into the client application.

## Related sections

*Packaging a Java SE or POJO client application*

---

## Setting up the environment

To use the Ant task to repackage the JRules execution archives, you must make sure that you have installed Ant version 1.6.5 or a later version. You can find a correct version of Ant in <JRules\_InstallDir>/shared/tools/ant.

### To set up the correct version of Ant:

1. Set the ANT\_HOME environment variable to <JRules\_InstallDir>/shared/tools/ant.
2. Set the JAVA\_HOME environment variable to the path of your JDK installation (1.5 or higher).
3. Add the <JRules\_InstallDir>/shared/tools/ant/bin directory to the PATH environment variable.

### See also

*Repackaging the EJB3 archive*

*Repackaging the HTDS archive*

*Repackaging the MDB archive*

---

## Repackaging the HTDS archive

The wbm-htds Ant task is called from <JRules\_InstallDir>/SupportPacs/SupportPacLB01/bin/setup-wbm.xml.

The Ant task packages the SupportPac runtime files to the HTDS EAR:

- ◆ The dvs-dw.properties file is packaged in jrules-res-htds-wbm-WAS7.ear/jrules-res-htds-WAS7.war/WEB-INF/classes.
- ◆ The jrules-wbm-runtime.jar file is packaged in jrules-res-htds-wbm-WAS7.ear/jrules-res-htds-WAS7.war/WEB-INF/lib.

The following table describes the properties in the wbm-htds Ant task.

Element property	Description	Default value
htds.ear.in	The path to the input HTDS EAR file.	/executionserver/applicationservers/WebSphere7/jrules-res-htds-WAS7.ear
htds.ear.out	The path to the output HTDS EAR file that contains the SupportPac runtime files.	/executionserver/applicationservers/WebSphere7/jrules-res-htds-wbm-WAS7.ear
wbm.prop.file	The path to the SupportPac properties file to be included in the output HTDS EAR file.	/SupportPacs/SupportPacLB01/lib/dvs-dw.properties
wbm.jar.file	The path to the SupportPac JAR file to be included in the output HTDS EAR file.	/SupportPacs/SupportPacLB01/lib/jrules-wbm-runtime.jar

To generate a new HTDS EAR file that contains the SupportPac runtime files:

1. Open a command prompt and go to the <JRules\_InstallDir>/SupportPacs/SupportPacLB01/bin/ directory.
2. Enter the following command ant -f setup-wbm.xml and press ENTER.

This command generates a jrules-res-htds-wbm-WAS7.ear file that contains the SupportPac runtime files.

You can use the properties to modify the directories and file names. For example, to change the directory and name of the output HTDS EAR file:

- ◆ Enter the following command `ant -f setup-wbm.xml -Dhtds.ear.out=../wbm-htds.ear` and press ENTER to generate a `wbm-htds.ear` file.

For more information on the JRules execution archives, see the Rule Execution Server documentation in the *WebSphere ILOG JRules BRMS information center*.

### **See also**

*Setting up the environment*

*Repackaging the EJB3 archive*

*Repackaging the MDB archive*

---

## Repackaging the EJB3 archive

The wbm-ejb3 Ant task is called from the <JRules\_InstallDir>/SupportPacs/SupportPacLB01/bin/setup-wbm.xml.

The following table describes the properties in the wbm-ejb3 Ant task.

Element property	Description	Default value
ejb3.jar.in	The path to the input EJB3 rule session JAR file.	/executionserver/applicationservers/WebSphere7/jrules-res-session-ejb3-WAS7.jar
ejb3.jar.out	The path to the output EJB3 rule session JAR file that contains the SupportPac runtime files.	/executionserver/applicationservers/WebSphere7/jrules-res-session-ejb3-wbm-WAS7.jar
wbm.prop.file	The path to the SupportPac properties file to be included in the output EJB3 rule session JAR file.	/SupportPacs/SupportPacLB01/lib/dvs-dw.properties
wbm.jar.file	The path to the SupportPac JAR file to be included in the output EJB3 rule session JAR file.	/SupportPacs/SupportPacLB01/lib/jrules-wbm-runtime.jar

To generate a new EJB3 JAR file that contains the SupportPac runtime files:

1. Open a command prompt and go to the <JRules\_InstallDir>/SupportPacs/SupportPacLB01/bin/ directory.
2. Enter the following command ant wbm-ejb3 -f setup-wbm.xml and press ENTER.

This command generates a jrules-res-session-ejb3-wbm-WAS7.jar file that contains the SupportPac runtime files.

You can use the properties to modify the directories and the file names. For example, to change the directory and the name of the output EJB3 JAR file:

- ◆ Enter the following command `ant -f setup-wbm.xml -Dejb3.jar.out=../wbm-ejb3.jar` and press ENTER to generate an output file called `wbm-ejb3.jar`.

For more information on the JRules execution archives, see the Rule Execution Server documentation in the *WebSphere ILOG JRules BRMS information center*.

### **See also**

- Setting up the environment*
- Repackaging the HTDS archive*
- Repackaging the MDB archive*

---

## Repackaging the MDB archive

The wbm-mdb Ant task is called from the <JRules\_InstallDir>/SupportPacs/SupportPacLB01/bin/setup-wbm.xml.

The following table describes the properties in the wbm-mdb Ant task.

Element property	Description	Default value
mdb.jar.in	The path to the input MDB JAR file.	/executionserver/applicationservers/WebSphere7/jrules-res-mdb-WAS7.jar
mdb.jar.out	The path to the output MDB JAR file that contains the SupportPac runtime files.	/executionserver/applicationservers/WebSphere7/jrules-res-mdb-wbm-WAS7.jar
wbm.prop.file	The path to the SupportPac properties file to be included in the output MDB JAR file.	/SupportPacs/SupportPacLB01/lib/dvs-dw.properties
wbm.jar.file	The path to the SupportPac JAR file to be included in the output MDB JAR file.	/SupportPacs/SupportPacLB01/lib/jrules-wbm-runtime.jar

To generate a new MDB JAR file that contains the SupportPac runtime files:

1. Open a command prompt and go to the <JRules\_InstallDir>/SupportPacs/SupportPacLB01/bin/ directory.
2. Enter the following command ant wbm-mdb -f setup-wbm.xml and press ENTER.

This command generates a jrules-res-mdb-wbm-WAS7.jar file that contains the SupportPac runtime files.

You can use the properties to modify the directories and the file names. For example, to change the directory and the name of the output MDB JAR file:

- ◆ Enter the following command ant -f setup-wbm.xml -Dmdb.jar.out=./wbm-mdb.jar and press ENTER to generate an output file called wbm-mdb.jar.

For more information on the JRules execution archives, see the Rule Execution Server documentation in the *WebSphere ILOG JRules BRMS information center*.

## **See also**

- Setting up the environment*
- Repackaging the HTDS archive*
- Repackaging the EJB3 archive*

---

## Packaging the rule session JAR into the client application

You must then assemble the client application modules into a Java EE application EAR file including the `jrules-res-ejb3-wbm-WAS7.jar` or `jrules-res-mdb-wbm-WAS7.jar` files.

In your `application.xml` file, add the following code. For example:

```
<application>
    .....
    <module>
        <ejb>my_ejb_application.jar</ejb>
    </module>
    <module>
        <ejb>jrules-res-ejb3-wbm-WAS7.jar</ejb>
    </module>
    ....
</application>
```

Then package the Java XOMs in your client. The simplest way is to put the classes into a separate JAR file, then package the archive with the rule session archive in the same EAR file.

**Note:** If you use WebSphere Application Server in secure mode, the client application that uses and packages the Rule Execution Server rule sessions requires a `META-INF/was.policy` file with the following content:

```
grant codeBase "file:jrules-res-ejb3-wbm-WAS7.jar" {
    permission java.lang.Runtime
    Permission "getClassLoader";permission java.io.FilePermission "-",
    "read";
    /* MBean plugin */
    permission com.ibm.websphere.security.WebSphereRuntimePermission
    "AdminPermission";
    permission javax.management.MBeanPermission "*", "*";
    permission java.util.PropertyPermission "*", "read,write";
};
```

---

## Deploying the monitoring artifacts

Before deploying the monitor model to the production environment, you must test the monitor model in the Monitor test environment.

After creating your client application and the Java EE monitor application project that contains your monitor model, you deploy them to WebSphere Application Server.

### To deploy the Java EE monitor application and the client application projects:

1. In the Servers view, select **WebSphere Business Monitor Server on WebSphere Application Server**.

**Note:** If you cannot see the Servers view, click **Window > Show View > Servers**.

2. Right-click the server and click **Start**.

The server might take a few minutes to start. Wait until the server is started and synchronized.

3. Right-click the server and click **Add and Remove Projects**.
4. Under Available projects, select the Java EE monitor application and your client application project, and then click **Add**.
5. Click **Finish** to publish the projects to the test server.

The projects are deployed to the WebSphere Business Monitor profile on WebSphere Application Server, and are displayed under the WebSphere Business Monitor test server.

**Tip:** If you are using HTDS, you must also deploy the `jrules-res-htds-wbm-WAS7.ear` file that contains the SupportPac runtime files.

For more information on deploying the Java EE monitor project application, see the section *Testing monitor models > Adding projects to the Monitor test environment server* in the *WebSphere Business Monitor Development Toolkit information center*.

### See also

*Starting the server*

*Deploying a RuleApp to Rule Execution Server*

*Generating Java EE monitor projects*

---

## Opening Business Space

After deploying your client application and the Java EE monitor application to the server, login to Business Space.

### To open Business Space from WebSphere Integration Developer or Rational Application Developer:

1. Make sure that you are in the Business Monitoring perspective.
2. In the Servers view, select the test server, for example **WebSphere Business Monitor on WebSphere Application Server**.
3. If the server is not started, right-click the server and click **Start**.
4. Right-click the test server and click **Business Space**.  
Business Space opens in a new browser window.
5. Enter your User ID and password, for example:
  - ◆ **User ID:** admin
  - ◆ **Password:** admin
6. Click **Login**.

This might take a few seconds to complete.

To view the KPIs in Business Space, you must first create a space and add widgets to specify how you want to display the information from the KPIs. For example, you can display the KPIs as graphs, gauges, and tables.

After creating the space and widgets in Business Space, you can execute your client application and view the results of your KPIs in Business Space.

For more information on configuring Business Space, creating widgets, viewing KPIs, and setting alerts, see the section “Viewing monitored data” in the *WebSphere Business Monitor information center*

### See also

*Sample: Decision monitoring*

*Starting the server*

*Generating Java EE monitor projects*

---

## Customizing the trace

To customize the trace, you can enable or disable the properties for monitoring, Decision Warehouse persistence, and event emission. You can set these properties either in Rule Studio before deploying the ruleset to Rule Execution Server, or in the Rule Execution Server Console if you have already deployed the ruleset.

---

### Enabling trace events with Decision Warehouse disabled

To emit trace events without storing the execution trace in Decision Warehouse, set the ruleset properties as follows:

- ◆ monitoring.enabled=true
- ◆ monitoring.dw.enabled=false
- ◆ monitoring.events.enabled=true

---

### Enabling trace events with Decision Warehouse enabled

To store the execution trace in Decision Warehouse and also emit trace events, set the ruleset properties as follows:

- ◆ monitoring.enabled=true
- ◆ monitoring.dw.enabled=true
- ◆ monitoring.events.enabled=true

---

### Disabling trace events with Decision Warehouse enabled

To store the execution trace in Decision Warehouse and disable event emission, set the monitoring.enabled ruleset property as follows:

- ◆ monitoring.enabled=true

If you have already added the monitoring.events.enabled and monitoring.dw.enabled to your ruleset, set them set as follows:

- ◆ monitoring.events.enabled=false
- ◆ monitoring.dw.enabled=true

---

## Optimizing the trace content

To limit the size of the execution trace and filter out some trace content, you can use filters.

To reduce the information that is stored in an execution trace in Decision Warehouse or sent to the event, you apply the property `monitoring.filters` to the ruleset and set filters on the property to refine the data stored.

1. Add the `monitoring.filters` ruleset property and set the value to a string that includes the list of filters that you want to apply.
2. Set the filters to `true` according to the data that you want to see in the trace.

**Note:** Do not use the property `monitoring.inout.filters` for a ruleset based on an XSD XOM.

The following filters have been modified for this SupportPac:

Filter	Default	Description
INFO_EXECUTION_EVENTS	false	Lists the rules and tasks executed
INFO_RULES	false	Total number of rules in the ruleset
INFO_TASKS	false	Total number of tasks in the ruleset

The following filters do not apply for event monitoring:

- ◆ INFO\_BOUND\_OBJECT\_BY\_RULE
- ◆ INFO\_TOTAL\_RULES\_NOT\_FIRED
- ◆ INFO\_TOTAL\_TASKS\_NOT\_EXECUTED
- ◆ INFO\_WORKING\_MEMORY
- ◆ INFO\_EXECUTION\_TRACE\_TREE
- ◆ INFO\_USER\_DATA

For more information on filters, see the section Rule Studio > Testing, validating, and auditing rules > Configuring and customizing Decision Warehouse > Optimizing Decision Warehouse, in the *WebSphere ILOG JRules BRMS information center*

---

## Using a customized trace DAO

A data source stores the execution traces of the rulesets that are enabled for Decision Warehouse. If you want to store the data in an existing enterprise database, use the extension API to create a custom Data Access Object (DAO) (see the `ilog.rules.res.persistence.trace` package). This object is then referenced by the calling client.

If you create a custom DAO and you want to be able to query the custom data source from the Decision Warehouse tab of the Rule Execution Server Console, you must also customize how trace data can be queried and loaded. In this case, you must package your custom DAO (which implements the `I1rTraceDAO` interface) in the Rule Execution Server Console archive.

The customization uses the `I1rTraceDAOFactory` class, which is called by the Rule Execution Server Session API. The `dw.factoryclassname` configuration property is specified in the file `dvs-dw.properties` file.

By setting the Trace DAO factory classname to `ilog.rules.res.persistence.impl.jdbc.I1rDatasourceTraceDAOFactory`, the class creates the custom DAO. The `createDAO` method is called to return an `I1rTraceDAO` object. If the `ruleset.monitoring.enabled` is set to true, `I1rTraceDAO.saveTrace(I1rDWTrace, I1rSessionRequest, I1rSessionResponse)` is called each time the ruleset is executed.

For more information on using a customized trace DAO, see the section Samples > Decision Validation Services samples > Decision Warehouse customization sample, in the *WebSphere ILOG JRules BRMS information center*

### See also

*SupportPac runtime files*

---

## Troubleshooting

If you encounter persistence or event emission problems, add the code below to your client application to retrieve a log:

- ◆ Execute and get the response for this request:

```
IlrSessionResponse response = session.execute(sessionRequest);
```

- ◆ Get the list of warnings:

```
List<IlrWarning> warnings = response.getWarnings();
```

- ◆ Display warning messages and corresponding stack traces to the System.err stream:

```
for (IlrWarning warning : warnings) {  
    System.err.println(warning.getMessage());  
    Throwable errorCause = warning.getErrorCause();  
    if (errorCause != null) {  
        errorCause.printStackTrace();  
    }  
}
```

This code returns a list of warnings related to the execution. The warnings that contain a stack trace correspond to an exception. Use the stack trace report to find where the errors occur.

### See also

*Using a customized trace DAO*

*Customizing the trace*



# Index

## A

archives  
  EJB3 **73**  
  HTDS **73**  
  MDB **73**

## B

business  
  KPIs **56**  
  metrics **26, 56, 60**  
Business Space **26, 83**

## C

cells  
  installing, across multiple cells **20**  
client applications  
  Java SE **71**  
  POJO **71**

configuring  
  WebSphere Business Monitor **23**

creating  
  Key Performance Indicators (KPIs) **62**  
  metrics **60**  
  rule projects **40**  
  RuleApps **41**

customizing  
  trace **84**

## D

decision  
  events **26**  
  trace **26**

decision event **26, 55**  
decision trace  
  metric **66**  
Decision Warehouse **66, 84**  
  installing **19**  
deploying  
  client application **82**  
  Java EE monitor application **82**  
  RuleApps **47**  
development environment  
  installing **13**  
dimensions **26**  
dvs-dw.properties **23, 70**

## E

EJB3  
  archive **77**  
event parts **55**  
events  
  decision **26**  
  emission **87**  
Execution Object Model (XOM) **40**

## F

filters  
  trace **84**

## G

generating  
  monitor models **58**

## H

HTDS

<p><b>I</b></p> <ul style="list-style-type: none"> <li>archive <b>75</b></li> <li>installation <b>5</b> <ul style="list-style-type: none"> <li>prerequisites <b>5, 8</b></li> </ul> </li> <li>installing           <ul style="list-style-type: none"> <li>across multiple cells <b>20</b></li> <li>Decision Warehouse <b>19</b></li> <li>development environment <b>5, 13</b></li> <li>Rule Execution Server <b>19</b></li> <li>Rule Studio, in Rational Application Developer <b>14</b></li> <li>Rule Studio, in WebSphere Integration Developer <b>14</b></li> <li>runtime environment <b>5, 17</b></li> <li>SupportPac <b>11, 15</b></li> </ul> </li> </ul>	<p>rulesets <b>43</b></p> <p><b>O</b></p> <ul style="list-style-type: none"> <li>optimizing           <ul style="list-style-type: none"> <li>trace content <b>84</b></li> </ul> </li> </ul> <p><b>P</b></p> <ul style="list-style-type: none"> <li>predefined           <ul style="list-style-type: none"> <li>metrics <b>56</b></li> </ul> </li> <li>prerequisites           <ul style="list-style-type: none"> <li>installation <b>8</b></li> </ul> </li> <li>profiles           <ul style="list-style-type: none"> <li>WebSphere Business Monitor <b>18</b></li> </ul> </li> <li>projects           <ul style="list-style-type: none"> <li>Java EE monitor <b>26, 68</b></li> <li>rule <b>40</b></li> <li>RuleApps <b>44</b></li> </ul> </li> </ul> <p><b>R</b></p> <ul style="list-style-type: none"> <li>Rational Application Developer <b>5</b></li> <li>repackaging           <ul style="list-style-type: none"> <li>EJB3 archive <b>77</b></li> <li>HTDS archive <b>75</b></li> <li>MDB archive <b>79</b></li> </ul> </li> <li>Rule Execution Server <b>19</b> <ul style="list-style-type: none"> <li>installing <b>19</b></li> </ul> </li> <li>Rule Execution Server Console <b>48</b></li> <li>rule project <b>58</b></li> <li>rule projects           <ul style="list-style-type: none"> <li>creating <b>40</b></li> </ul> </li> <li>rule sessions <b>69</b></li> <li>Rule Studio           <ul style="list-style-type: none"> <li>installing <b>14</b></li> </ul> </li> <li>RuleApps <b>44</b> <ul style="list-style-type: none"> <li>creating <b>41</b></li> <li>deploying <b>47</b></li> </ul> </li> <li>ruleset           <ul style="list-style-type: none"> <li>parameters <b>55</b></li> <li>path <b>48, 64</b></li> </ul> </li> <li>ruleset properties <b>44</b></li> <li>rulesets           <ul style="list-style-type: none"> <li>monitoring <b>43</b></li> <li>properties <b>44</b></li> </ul> </li> <li>runtime environment           <ul style="list-style-type: none"> <li>installing <b>17</b></li> </ul> </li> <li>runtime files <b>69</b></li> </ul>
--	---

## S

server  
    starting **46**

SupportPac  
    installing **11, 15**  
    runtime files **70**

## T

trace  
    customizing **84**  
    filters **84**  
    optimizing content **84**  
troubleshooting **87**

## W

WebSphere Application Server **5, 18**

WebSphere Business Monitor **5**  
    configuring **23**  
    profile **18**

WebSphere Business Monitor Development Toolkit  
**5**

WebSphere ILOG JRules BRMS **5**

## X

XOM  
    *See Execution Object Model (XOM)*