

IBM WebSphere Operational Decision Management
Version 8 Release 0

*WebSphere Operational Decision
Management Integration for IBM
Business Monitor
SupportPac LB01 V1.2*



Note

Before using this information and the product it supports, read the information in "Notices" on page 63.

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

© **Copyright IBM Corporation 2008, 2012.**

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

Contents

Chapter 1. Installing	1	Creating business metrics.	36
Installation overview	1	Creating KPIs.	37
Installation prerequisites	2	Changing the path to the monitored ruleset	38
Extracting the SupportPac deliverable	4	Setting the Decision Warehouse trace metric	39
Installing the development environment	5	Chapter 7. Generating Java EE monitor projects	43
Installing Rule Designer in Integration Designer or Rational Application Developer	5	Chapter 8. Packaging the runtime files in the client application	45
Installing the SupportPac integration feature.	6	SupportPac runtime files	45
Installing the runtime environment	7	Packaging a Java SE or POJO client application	45
Setting up a Business Monitor profile	7	Repackaging the HTDS, EJB3, and MDB archives using Ant	46
Installing Rule Execution Server and setting up the Decision Warehouse database	8	Setting up the environment	46
Installing the monitoring service across multiple cells	9	Repackaging the HTDS archive.	47
Configuring Business Monitor with JMS.	10	Repackaging the EJB3 archive	48
Chapter 2. Overview	13	Repackaging the MDB archive	49
Chapter 3. Decision monitoring sample 15		Packaging the rule session JAR into the client application	50
Decision monitoring sample summary	15	Chapter 9. Deploying the monitoring artifacts	51
Setting up the Decision monitoring sample.	16	Chapter 10. Opening Business Space 53	
Decision monitoring sample details	17	Chapter 11. Customizing the trace	55
Chapter 4. Creating rule projects and RuleApps	25	Chapter 12. Using a customized trace DAO.	57
Importing or creating rule projects.	25	Chapter 13. Troubleshooting	59
Creating RuleApps for ruleset deployment	25	Index	61
Chapter 5. Enabling ruleset monitoring 27		Notices	63
Adding ruleset properties	27	Trademarks	65
Starting the server	28		
Deploying a RuleApp to Rule Execution Server	29		
(Optional) Viewing the decision service WSDL file	29		
Chapter 6. Generating a monitor model 31			
Monitor models and contexts	31		
Decision event	32		
Metrics and KPIs	33		
Generating a monitor model from a rule project	34		

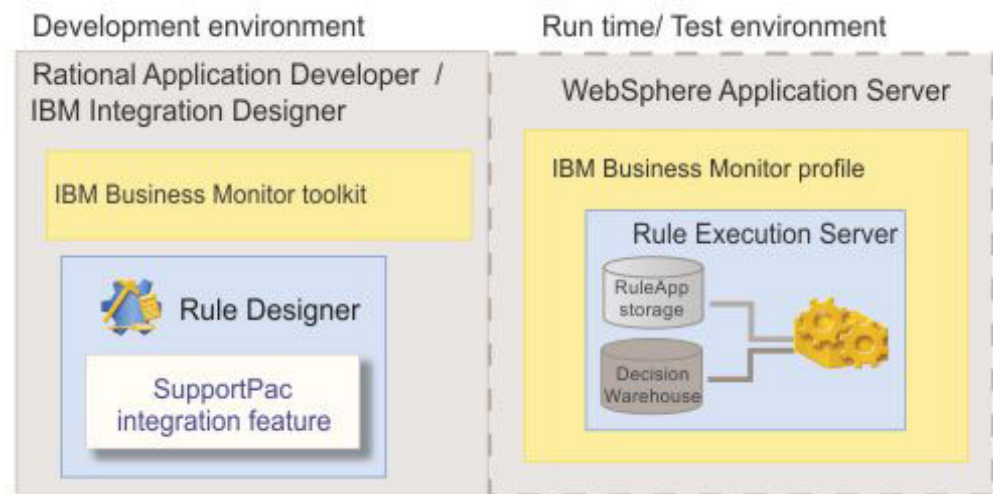
Chapter 1. Installing

To install WebSphere® Operational Decision Management Integration for Business Monitor SupportPac LB01 V1.2, you must first extract the contents of the SupportPac ZIP file, and install the Eclipse feature in Rule Designer. Then, you install the runtime environment by installing Rule Execution Server on WebSphere Application Server.

Installation overview

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

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



Development

This environment is based on Integration Designer or Rational Application Developer, and includes the following products and features:

- Business Monitor development toolkit that includes both the development environment and the test environment, see “Installation prerequisites” on page 2.
- Rule Designer installed on top on Integration Designer or Rational Application Developer, see “Installing Rule Designer in Integration Designer or Rational Application Developer” on page 5.
- The SupportPac integration feature, see “Installing the SupportPac integration feature” on page 6.

Run time / Test

Rule Execution Server is installed in the Business Monitor profile on WebSphere Application Server, see “Installing Rule Execution Server and setting up the Decision Warehouse database” on page 8.

Related information:

“Installation prerequisites”

To install the WebSphere Operational Decision Management for IBM Business Monitor SupportPac, you must make sure that you have installed the required products and the correct product versions.

“Extracting the SupportPac deliverable” on page 4

Extract the contents of the SupportPac ZIP file in your WebSphere Operational Decision Management installation.

“Installing the development environment” on page 5

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

“Installing the runtime environment” on page 7

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 Business Monitor profile.

Installation prerequisites

To install the WebSphere Operational Decision Management for IBM Business Monitor SupportPac, you must make sure that you have installed the required products and the correct product versions.

For a list of the prerequisites and their versions, see www.ibm.com/support/docview.wss?uid=swg24027554.

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 Operational Decision Management and the Java EE add-on for WebSphere Application Server.
- Integration Designer or Rational Application Developer.
- Business Monitor development toolkit.

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

The Business Monitor Development Toolkit includes the following environments:

- A monitor **development environment**: that is already included in Integration Designer, but that you must install if you are working in Rational Application Developer.
- A monitor **test environment**: that includes a Business Monitor profile on WebSphere application Server, and Business Space.

For more information about the installation of the development toolkit or the test environment, see the Business Monitor information center.

Tip:

To check the version of Integration Designer, Rational Application Developer, or 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:

- IBM Business Monitor

For information about the installation of Business Monitor, see the Business Monitor information center

- WebSphere Application Server

For information about the installation of WebSphere Application Server, see the WebSphere Application Server information center

- WebSphere Operational Decision Management and the Java EE add-on for WebSphere Application Server.

For information about the installation of WebSphere Operational Decision Management, see the WebSphere Operational Decision Management information center.

Related information:

“Installation overview” on page 1

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

“Installing the development environment” on page 5

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

“Installing the runtime environment” on page 7

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 Business Monitor profile.

Extracting the SupportPac deliverable

Extract the contents of the SupportPac ZIP file in your WebSphere Operational Decision Management installation.

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 Business Monitor.

You must extract the contents of the SupportPac ZIP file at the root of your WebSphere Operational Decision Management installation in <WODM_InstallDir>, generally C:/Program Files/IBM/WODM80/WODM.

The ZIP file contains the following folders and files:

- /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 WebSphere Operational Decision Management Integration for IBM Business Monitor feature to install into Rule Designer.
- /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: <WODM_InstallDir>/SupportPacs/SupportPacLB01/.

Related information:

“Installation prerequisites” on page 2

To install the WebSphere Operational Decision Management for IBM Business Monitor SupportPac, you must make sure that you have installed the required products and the correct product versions.

“Installation overview” on page 1

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

“Installing the development environment”

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

“Installing the runtime environment” on page 7

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 Business Monitor profile.

Installing the development environment

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

Related information:

“Installation prerequisites” on page 2

To install the WebSphere Operational Decision Management for IBM Business Monitor SupportPac, you must make sure that you have installed the required products and the correct product versions.

“Extracting the SupportPac deliverable” on page 4

Extract the contents of the SupportPac ZIP file in your WebSphere Operational Decision Management installation.

“Installation overview” on page 1

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

“Installing the runtime environment” on page 7

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 Business Monitor profile.

Installing Rule Designer in Integration Designer or Rational Application Developer

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

For more information about the installation of Rule Designer in Integration Designer or Rational Application Developer, see the WebSphere Operational Decision Management information center.

Related tasks:

“Installing the SupportPac integration feature”

Install the WebSphere Operational Decision Management Integration for IBM Business Monitor feature to enable the development part of your monitoring application.

Installing the SupportPac integration feature

Install the WebSphere Operational Decision Management Integration for IBM Business Monitor feature to enable the development part of your monitoring application.

Before you begin

Make sure that your installation complies with the requirements listed in “Installation prerequisites” on page 2.

About this task

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

Procedure

To install the SupportPac feature:

1. Start Integration Designer or Rational Application Developer.
2. Switch to the **Java** perspective. On the **Window** menu, click **Open Perspective > Java**.
3. On the **Help** menu, click **Install New Software**.
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 <WODM_InstallDir>/SupportPacs/SupportPacLB01/eclipse directory, click **OK**, and then click **OK** again in the Add Site dialog.
The WebSphere Operational Decision Management for IBM Business Monitor site appears in the Available Software tab.
7. Expand <WODM_InstallDir>/SupportPacs/SupportPacLB01/eclipse, select the check box next to **WebSphere Operational Decision Management Integration for IBM 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 **WebSphere Operational Decision Management for Business Monitor** is in the list.

Related information:

“Installing Rule Designer in Integration Designer or Rational Application Developer” on page 5

Before installing the SupportPac integration feature, you must install Rule Designer into your installation of Integration Designer 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 Business Monitor profile.

Related information:

“Installation prerequisites” on page 2

To install the WebSphere Operational Decision Management for IBM Business Monitor SupportPac, you must make sure that you have installed the required products and the correct product versions.

“Extracting the SupportPac deliverable” on page 4

Extract the contents of the SupportPac ZIP file in your WebSphere Operational Decision Management installation.

“Installation overview” on page 1

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

“Installing the development environment” on page 5

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

Setting up a Business Monitor profile

To prepare the runtime environment to be able to monitor a ruleset using Business Monitor, you must have a Business Monitor profile on WebSphere Application Server.

About this task

The Business Monitor 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 Business Monitor.

For more information about the profiles created as part of the test environment installation, and the creation of monitor profiles in Business Monitor, see the 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.

Procedure

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**.

Related information:

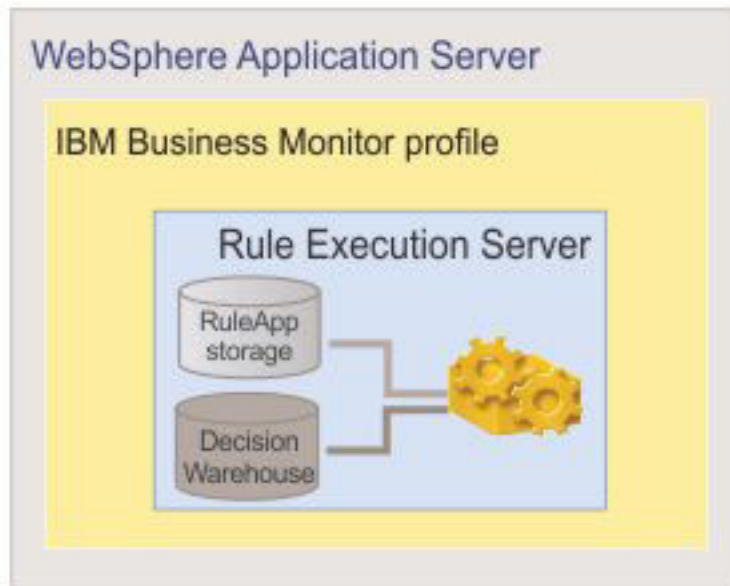
“Installing Rule Execution Server and setting up the Decision Warehouse database”
To enable ruleset execution, and store the ruleset execution traces, you must install Rule Execution Server on WebSphere Application Server, and set up the access to the Decision Warehouse database.

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 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 Operational Decision Management information center.

Related tasks:

“Setting up a Business Monitor profile” on page 7

To prepare the runtime environment to be able to monitor a ruleset using Business Monitor, you must have a Business Monitor profile on WebSphere Application Server.

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 Business Monitor profile.

About this task

In a production environment, you can install the monitoring service in a cross-cell topology. One cell hosts Rule Execution Server, and the other cell hosts a 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 Business Monitor profile, you must configure the remote service integration bus.

Procedure

To install the monitoring service across multiple cells:

1. Get the Business Monitor configRemoteMonitorBus utility.
2. Configure the script.

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

```
SECURE_CONFIGURATION=true
```

```
LOCAL_WAS_HOST=localhost  
LOCAL_WAS_PORT=8880  
LOCAL_WAS_MESSAGING_ENGINE=qnode.server1-MONITOR.qcell1.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.

Results

For information on how to enable the security between Business Monitor and the Rule Execution Server cells, see the Business Monitor information center.

Related information:

“Configuring 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.

Chapter 8, “Packaging the runtime files in the client application,” on page 45
 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.

Configuring 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.

Java Message Service (JMS) messages emit events in Business Monitor.

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 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 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 sends the message containing the trace to a 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 Business Monitor JMS queue, specify the name of your resource. For example:

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

Related tasks:

“Installing the monitoring service across multiple cells” on page 9

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

Related information:

Chapter 12, “Using a customized trace DAO,” on page 57

If you create a custom Data Access Object (DAO), you must also customize how trace data can be queried and loaded.

Chapter 2. Overview

This SupportPac provides real-time monitoring by connecting WebSphere Operational Decision Management and Business Monitor. Business users and analysts can monitor ruleset performance and analyze Key Performance Indicators (KPIs).

WebSphere Operational Decision Management offers a controlled way for business users and analysts to make operational business changes. Business users and analysts can use WebSphere Operational Decision Management 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 the time-to-market of a product.

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

In WebSphere Operational Decision Management, business performance can be assessed by recording how decisions are made with the current set of rules. Business users can view the trace 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.

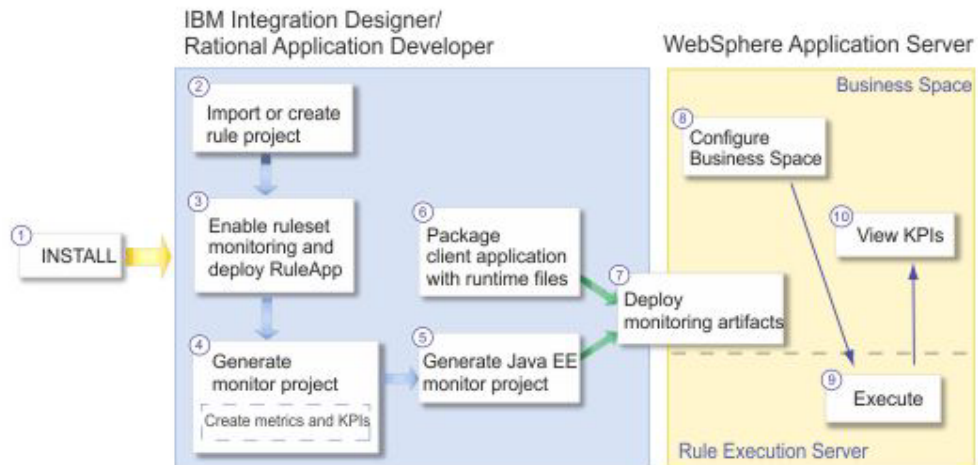
In addition to these features, this SupportPac provides real-time monitoring by connecting WebSphere Operational Decision Management and Business Monitor. Business users can use Business Monitor to view KPIs and receive alerts when unusual situations are detected. They use business dashboards in Business 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, and the tasks executed.

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 Business Monitor. They can identify which of the two rulesets 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 Integration Designer 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.

Related information:

Chapter 3, "Decision monitoring sample," on page 15

This sample shows how to monitor a decision event and how to create and view business metrics and KPIs.

Chapter 3. Decision monitoring sample

This sample shows how to monitor a decision event and how to create and view business metrics and KPIs.

Related tasks:

Chapter 7, “Generating Java EE monitor projects,” on page 43

Create Java EE monitor projects to test the monitor model in the test environment before sending it to production.

Related information:

“Metrics and KPIs” on page 33

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

Chapter 8, “Packaging the runtime files in the client application,” on page 45

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.

Chapter 5, “Enabling ruleset monitoring,” on page 27

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.

Decision monitoring sample summary

This sample shows how to monitor a decision event and how to create and view business metrics and KPIs.

Time required

30 minutes

Prerequisites

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

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

For more information about the required environment, see Chapter 1, “Installing,” on page 1.

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

You must 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.

Description

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 Business Monitor.
- Accessing the decision trace directly from Business Space.
- Viewing KPIs in Business Space.

Next  "Setting up the Decision monitoring sample"

Setting up the Decision monitoring sample

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

Procedure

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 `<WODM_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 Integration Designer or 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**.

Results

If you see any syntax errors in the loan-web project, you might need to include the jrules-res-session-java.jar and jrules-wbm-runtime.jar files in the lib directory of the project.

Decision monitoring sample details

You run the sample in Eclipse and Rule Execution Server.

Running this 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.

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. If necessary, modify the URL to Rule Execution Server.

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

The following filter specifies the host as localhost and the port as <PORT>.

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

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

To publish the monitor model to 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 the following projects are generated:

- MonitorLoanDecisionApplication: The Java EE application project that contains the model logic. 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.

For more information about these projects, see Chapter 7, “Generating Java EE monitor projects,” on page 43.

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 **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 **Business Monitor Server on WebSphere Application Server**.
- c. Make sure the **Server runtime environment** is correct: **Business Monitor Server on WebSphere Application Server**.

Click **Next**.

- d. In the **Profile name** field, make sure that the profile selected is the 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 set security for the admin user:

1. Log in to the WebSphere Application Server Integrated Solutions Console.
2. Click **Security** and then click **Monitor Data Security**.
3. Select the root resource group.
4. In the Roles section, expand **KPI Administrators**.
5. Under Users, click **Edit**.
6. Click **Search**.
7. Add the admin user to the Selected list and click **OK**.

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

1. In the Servers view, right-click the **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.
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.
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:<PORT>/res`
 - Important:** You must use the same port number as the server.
2. Enter the username and password.
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:

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.

Similarly, deploy the client application.

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.
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 `<WODM_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:<PORT>/loan-web`.

Important: You must 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:

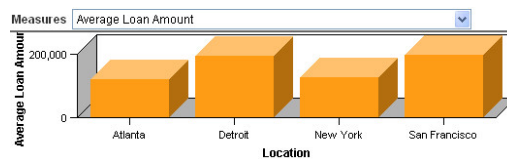
1. Switch back to Business Space.

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.

- 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**.
- For the Average Loan Amount KPI, right-click **All locations** and click **Drill Down**.

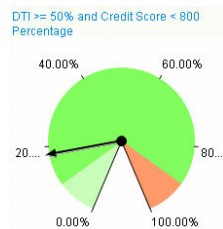
The dimension now shows the country level and displays the average loan amount for each country.

- 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.



- 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.



- 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	View Trace		0.04	195,000
France	Paris	October 22, 2010 2:50:54 AM	View Trace	eligibility.repayment and score 7, eligibility.minimum income	0.05	335,000
Australy	Camberra	October 22, 2010 2:50:54 AM	View Trace		0.03	390,000

- 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.

How this sample works

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

The ruleset must contain specific properties 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 about the event emission and Decision Warehouse properties, see Chapter 11, “Customizing the trace,” on page 55.

Monitor project

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

This 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**.

You can see 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
DataFactory.java	This class contains the functions to create the input loan and the input borrower from static and random data.
LoanBean.java	This class contains the bean to execute the ruleset. See the validateWithJRules function to see how to execute a ruleset on a POJO rule session.
XmlBorrower.java	This class is used to serialize in XML and from XML, and to display a borrower in JSF.
XmlLoan.java	This class is used to serialize in XML and from XML, and to display a loan in JSF.

Class	Description
Result.java	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 about the `dvs-dw.properties` file, see “SupportPac runtime files” on page 45, “Configuring Business Monitor with JMS” on page 10, and Chapter 12, “Using a customized trace DAO,” on page 57.

Chapter 4. Creating rule projects and RuleApps

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

Importing or creating rule projects

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

About this task

To generate a monitor model from a rule project, you must import an existing 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.

Procedure

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**.

Results

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**.

Creating RuleApps for ruleset deployment

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

About this task

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.

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

Procedure

To create a RuleApp project:

1. On the **File** menu, click **New Project**.
2. Expand **Rule Designer** and select **RuleApp Project**.
3. Click **Next**.
4. On the RuleApp Project page, enter the name of the project, and then click **Next**.

Use only a-z, A-Z, 0-9, and underscore (_). For the first character, 0-9 is not allowed.

5. Optional: 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, select **Use an extractor to filter rules contained in the project**, and select the required extractor from the list displayed.

The list you see depends on the rule project that 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, and then click **OK**.
 - e. Click **Finish**.

Results

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

Chapter 5. 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.

Related information:

Chapter 3, “Decision monitoring sample,” on page 15

This sample shows how to monitor a decision event and how to create and view business metrics and KPIs.

Chapter 4, “Creating rule projects and RuleApps,” on page 25

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

Adding ruleset properties

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

About this task

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

Property	Description
monitoring.enabled	Set this ruleset property to true to enable monitoring of ruleset execution for Decision Warehouse.
monitoring.events.enabled	Set this property to true to enable the ruleset emission of trace events.
monitoring.dw.enabled	This property is optional . 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 true to enable Decision Warehouse persistence.

Procedure

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, and 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 list, 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.

Results

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

Related information:

Chapter 11, “Customizing the trace,” on page 55

You can customize the trace by setting specific ruleset properties, or by adding filters on the decision trace.

Starting the server

Before deploying the RuleApp, you must start the server.

About this task

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

When you installed the Monitor test environment, a Business Monitor server was created in Rational Application Developer or Integration Designer. 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 Business Monitor Development Toolkit information center.

Procedure

To start the server in Integration Designer 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 Business Monitor Server on WebSphere Application

Server, and click the **Start the server** button  .

Related tasks:

“Deploying a RuleApp to Rule Execution Server”

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

Chapter 9, “Deploying the monitoring artifacts,” on page 51

To make sure that your monitor model works as expected, you deploy your client application and the Java EE monitor project to the Business Monitor environment.

Related information:

Chapter 3, “Decision monitoring sample,” on page 15

This sample shows how to monitor a decision event and how to create and view business metrics and KPIs.

Deploying a RuleApp to Rule Execution Server

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

About this task

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

Procedure

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, and 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, and then click **Finish**.

Related tasks:

“Starting the server” on page 28

Before deploying the RuleApp, you must start the 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.

About this task

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.

Procedure

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:<PORT>/res`.

Important:

The host and port must be the same as your server.

3. Enter the username and password of the Rule Execution Server role.
Both depend on the role that 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. In the Ruleset View, click **Show HTDS WDL Options**.
10. Click **Retrieve HTDS WSDL File**.
11. Select the **Decision trace information** option and then click **Download**.

Results

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

Related tasks:

“Starting the server” on page 28

Before deploying the RuleApp, you must start the server.

“Deploying a RuleApp to Rule Execution Server” on page 29

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

Chapter 6. Generating a monitor model

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

Related tasks:

Chapter 7, “Generating Java EE monitor projects,” on page 43

Create Java EE monitor projects to test the monitor model in the test environment before sending it to production.

Related information:

Chapter 3, “Decision monitoring sample,” on page 15

This sample shows how to monitor a decision event and how to create and view business metrics and KPIs.

Chapter 4, “Creating rule projects and RuleApps,” on page 25

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

Monitor models and contexts

Monitor models contain monitoring contexts and associated event definitions.

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

Based on the monitor model, Business Monitor performs the following actions:

- 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.

Related tasks:

“Generating a monitor model from a rule project” on page 34

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

Related information:

“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” on page 33

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

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.

The decision event is an inbound event called `DecisionEvent`. An inbound event 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.

Related tasks:

“Generating a monitor model from a rule project” on page 34

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

Related information:

“Monitor models and contexts” on page 31

Monitor models contain monitoring contexts and associated event definitions.

“Metrics and KPIs”

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

Metrics and KPIs

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

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.

Related tasks:

“Creating business metrics” on page 36

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.

“Setting the Decision Warehouse trace metric” on page 39

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

“Generating a monitor model from a rule project”

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

Related information:

Chapter 3, “Decision monitoring sample,” on page 15

This sample shows how to monitor a decision event and how to create and view business metrics and KPIs.

Generating a monitor model from a rule project

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

About this task

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

Procedure

To generate a monitor model from a rule project:

1. Make sure that you are in the **Rule** perspective. Click **Window > Open Perspective > Other**, select **Rule**, and then 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 do not 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 Generate Monitor Model window, specify a target monitor project and model name, and click **Finish**.
8. On the Choose What to Monitor page, in the Event Source section, select the monitor context.
9. 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.
10. 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 want to monitor.
You cannot remove an event if you have chosen a template that requires that event.
11. Click **Next**.
12. On the Monitor Model page, keep the default implementation settings, and click **Next**.
13. Click **Finish**.
14. 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.

Results

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**.

Related information:

“Monitor models and contexts” on page 31

Monitor models contain monitoring contexts and associated event definitions.

“Metrics and KPIs” on page 33

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

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.

About this task

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.

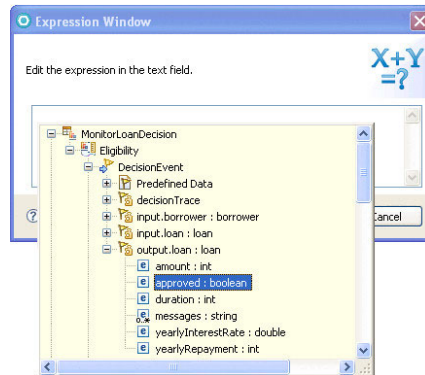
Procedure

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 that 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.

Results

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

Related tasks:

“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.

Related information:

“Metrics and KPIs” on page 33

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

Chapter 3, “Decision monitoring sample,” on page 15

This sample shows how to monitor a decision event and how to create and view business metrics and KPIs.

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.

About this task

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.

Procedure

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: Specify 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.

Related tasks:

“Creating business metrics” on page 36

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.

Related information:

“Metrics and KPIs” on page 33

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

Chapter 3, “Decision monitoring sample,” on page 15

This sample shows how to monitor a decision event and how to create and view business metrics and KPIs.

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.

About this task

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.

Procedure

To create a new context file from the same application:

1. Make sure that you are in the Business Monitoring perspective.

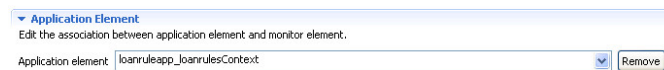
2. In Monitor Details Model tab, right-click the monitor model, and click **New > Create from Application**.
3. Under Event Source, select the monitor context and select the monitor templates (metrics and KPIs), click **Next**, and then click **Next** again.
4. Click **Finish**.

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.

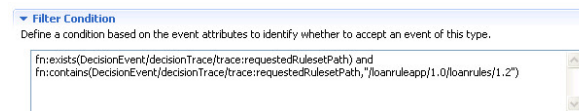
The following figure shows a dependency on an application element called `loanruleapp_loanrulesContext`.



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



10. Save the changes.

Setting the Decision Warehouse trace metric

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

About this task

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.

Procedure

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 <PORT> as the port.

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

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

Results

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 **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.
 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.

Related information:

“Metrics and KPIs” on page 33

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

Chapter 3, “Decision monitoring sample,” on page 15

This sample shows how to monitor a decision event and how to create and view business metrics and KPIs.

Chapter 5, “Enabling ruleset monitoring,” on page 27

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.

Chapter 7. Generating Java EE monitor projects

Create Java EE monitor projects to test the monitor model in the test environment before sending it to production.

About this task

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 tests in Business Space to check 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.

Procedure

To generate a Java EE monitor project:

1. Make sure that you are in the Business Monitoring 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**.

Results

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

- The **Model Logic project**: An EJB project that contains the event processing 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.

Related tasks:

“Generating a monitor model from a rule project” on page 34

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

Chapter 9, “Deploying the monitoring artifacts,” on page 51

To make sure that your monitor model works as expected, you deploy your client application and the Java EE monitor project to the Business Monitor environment.

Related information:

Chapter 3, “Decision monitoring sample,” on page 15

This sample shows how to monitor a decision event and how to create and view business metrics and KPIs.

Chapter 8. Packaging the runtime files 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.

Related tasks:

Chapter 9, “Deploying the monitoring artifacts,” on page 51

To make sure that your monitor model works as expected, you deploy your client application and the Java EE monitor project to the Business Monitor environment.

SupportPac runtime files

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

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

- `<WODM_InstallDir>/SupportPacs/SupportPacLB01/lib/jrules-wbm-runtime.jar`
- `<WODM_InstallDir>/SupportPacs/SupportPacLB01/lib/dvs-dw.properties`

`jrules-wbm-runtime.jar`

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

`dvs-dw.properties`

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

For more information about the `dvs-dw.properties` file, see Chapter 12, “Using a customized trace DAO,” on page 57 and “Configuring Business Monitor with JMS” on page 10.

Related tasks:

“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.

Related information:

“Repackaging the HTDS, EJB3, and MDB archives using Ant” on page 46

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 archives with the SupportPac files.

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.

About this task

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.

Procedure

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.

Related information:

“Repackaging the 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 archives with the SupportPac files.

Repackaging the 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 archives with the SupportPac files.

Related tasks:

“Packaging a Java SE or POJO client application” on page 45

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

Setting up the environment

Setting up Ant to repackage the HTDS, EJB3, and MDB archives.

About this task

To use the Ant task to repackage the WebSphere Operational Decision Management execution archives, you must have installed Ant version 1.6.5 or a later version. You can find a correct version of Ant in `<WODM_InstallDir>/shared/tools/ant`.

Procedure

To set up the correct version of Ant:

1. Set the `ANT_HOME` environment variable to `<WODM_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 `<WODM_InstallDir>/shared/tools/ant/bin` directory to the `PATH` environment variable.

Related information:

“Repackaging the EJB3 archive” on page 48

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

“Repackaging the HTDS archive”

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

“Repackaging the MDB archive” on page 49

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

Repackaging the HTDS archive

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

The wbm-htds Ant task is called from <WODM_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/applicationserver/WebSphere7/jrules-res-htds-WAS7.ear
htds.ear.out	The path to the output HTDS EAR file that contains the SupportPac runtime files.	/executionserver/applicationserver/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 <WODM_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.

Related tasks:

“Setting up the environment” on page 46

Setting up Ant to repackage the HTDS, EJB3, and MDB archives.

Related information:

“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” on page 49

Use the `wbm-mdb` Ant task to repackage the MDB JAR 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.

The `wbm-ejb3` Ant task is called from the `<WODM_InstallDir>/SupportPacs/SupportPacLB01/bin/setup-wbm.xml`.

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

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

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

1. Open a command prompt and go to the `<WODM_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`.

Related tasks:

“Setting up the environment” on page 46
 Setting up Ant to repackage the HTDS, EJB3, and MDB archives.

Related information:

“Repackaging the HTDS archive” on page 47
 Use the wbm-htds Ant task to repackage the HTDS EAR 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.

Repackaging the MDB archive

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

The wbm-mdb Ant task is called from the <WODM_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/ applicationserver/ WebSphere7/jrules-res-mdb- WAS7.jar
mdb.jar.out	The path to the output MDB JAR file that contains the SupportPac runtime files.	/executionserver/ applicationserver/ 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 <WODM_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`.

Related tasks:

“Setting up the environment” on page 46
 Setting up Ant to repackage the HTDS, EJB3, and MDB archives.

Related information:

“Repackaging the HTDS archive” on page 47
 Use the wbm-htds Ant task to repackage the HTDS EAR to include the SupportPac runtime files.

“Repackaging the EJB3 archive” on page 48
 Use the wbm-ejb3 Ant task to repackage the EJB3 rule session 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.

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";
};
```

Chapter 9. Deploying the monitoring artifacts

To make sure that your monitor model works as expected, you deploy your client application and the Java EE monitor project to the Business Monitor environment.

About this task

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.

Procedure

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

1. In the Servers view, select **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.

Results

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

Related tasks:

“Starting the server” on page 28

Before deploying the RuleApp, you must start the server.

“Deploying a RuleApp to Rule Execution Server” on page 29

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

Chapter 7, “Generating Java EE monitor projects,” on page 43

Create Java EE monitor projects to test the monitor model in the test environment before sending it to production.

Chapter 10. Opening Business Space

To view the KPIs that you have created, you set up a monitoring space and add widgets in Business Space.

About this task

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

Procedure

To open Business Space from Integration Designer 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 **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.
6. Click **Login**.

This might take a few seconds to complete.

Results

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.

Related tasks:

“Starting the server” on page 28

Before deploying the RuleApp, you must start the server.

Chapter 7, “Generating Java EE monitor projects,” on page 43

Create Java EE monitor projects to test the monitor model in the test environment before sending it to production.

Related information:

Chapter 3, “Decision monitoring sample,” on page 15

This sample shows how to monitor a decision event and how to create and view business metrics and KPIs.

Chapter 11. Customizing the trace

You can customize the trace by setting specific ruleset properties, or by adding filters on the decision trace.

You can set ruleset properties to enable or disable monitoring, Decision Warehouse persistence, and event emission. This can be done either in Rule Designer 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 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

Chapter 12. Using a customized trace DAO

If you create a custom Data Access Object (DAO), you must also customize how trace data can be queried and loaded.

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 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 `IlrTraceDAO` interface) in the Rule Execution Server Console archive.

The customization uses the `IlrTraceDAOFactory` 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.IlrdatasourceTraceDAOFactory`, the class creates the custom DAO. The `createDAO` method is called to return an `IlrTraceDAO` object. If the ruleset property `monitoring.enabled` is set to `true`, `IlrTraceDAO.saveTrace(IlrDWTrace, IlrSessionRequest, IlrSessionResponse)` is called each time the ruleset is executed.

Related information:

“SupportPac runtime files” on page 45

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

Chapter 13. Troubleshooting

You can add code to a client application to help you solve problems.

If you encounter persistence or event emission problems, add the following code 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.

Related information:

Chapter 12, “Using a customized trace DAO,” on page 57

If you create a custom Data Access Object (DAO), you must also customize how trace data can be queried and loaded.

Chapter 11, “Customizing the trace,” on page 55

You can customize the trace by setting specific ruleset properties, or by adding filters on the decision trace.

Index

A

archives
 EJB3 46
 HTDS 46
 MDB 46

B

business
 KPIs 33
 metrics 15, 33, 36
Business Monitor 1
 configuring 10
 profile 7
Business Monitor Development
 Toolkit 1
Business Space 15, 53

C

cells
 installing, across multiple cells 9
client applications
 Java SE 46
 POJO 46
configuring
 Business Monitor 10
creating
 Key Performance Indicators
 (KPIs) 37
 metrics 36
 rule projects 25
 RuleApps 25
customizing
 trace 55

D

decision
 events 15
 trace 15
decision event 15, 32
decision trace
 metric 39
Decision Warehouse 39, 55
 installing 8
deploying
 client application 51
 Java EE monitor application 51
 RuleApps 29
development environment
 installing 5
dimensions 15
dvs-dw.properties 10, 45

E

EJB3
 archive 48

event parts 32
events
 decision 15
 emission 59
Execution Object Model (XOM) 25

F

filters
 trace 55

G

generating
 monitor models 34

H

HTDS
 archive 47

I

installation 1
 prerequisites 1, 2
installing
 across multiple cells 9
 Decision Warehouse 8
 development environment 1, 5
 Rule Designer, in Integration
 Designer 5
 Rule Designer, in Rational Application
 Developer 5
 Rule Execution Server 8
 runtime environment 1, 7
 SupportPac 4, 6

J

Java EE monitor
 projects 15, 43
Java EE monitor application
 deploying 51

K

Key Performance Indicators (KPIs) 15,
 33
 creating 37
KPI context 37
KPI model 37
KPIs
 See Key Performance Indicators (KPIs)
 business 33

M

MDB
 archive 49
metrics 15, 33
 business 15, 33, 36
 creating 36
 decision trace 39
 predefined 33
monitor
 contexts 31, 38
 models 31, 34
 projects 31
monitor models
 generating 34
monitoring
 rulesets 27

O

optimizing
 trace content 55

P

predefined
 metrics 33
prerequisites
 installation 2
profiles
 Business Monitor 7
projects
 Java EE monitor 15, 43
 rule 25
 RuleApps 27

R

Rational Application Developer 1
repackaging
 EJB3 archive 48
 HTDS archive 47
 MDB archive 49
Rule Designer
 installing 5
Rule Execution Server 8
 installing 8
Rule Execution Server Console 29
rule projects 34
 creating 25
rule sessions 45
RuleApps 27
 creating 25
 deploying 29
ruleset
 monitoring 27
 parameters 32
 path 29, 38
 properties 27

- runtime
 - environment, installing 7
 - files 45

S

- server
 - starting 28
- SupportPac
 - installing 4, 6
 - runtime files 45

T

- trace
 - customizing 55
 - filters 55
 - optimizing content 55
- troubleshooting 59

W

- WebSphere Application Server 1, 7

X

- XOM
 - See* Execution Object Model (XOM)

Notices

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

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

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

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

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

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

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

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

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

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

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

Trademarks

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

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

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

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

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



Printed in USA