



The list of topics for the preliminary tutorial document is as given below:

1. Introduction
2. Application Overview
3. Application Design
4. Installation
5. Starting
6. Testing

## Introduction

Enterprise Management Applications implement a number of Southbound protocols. Java Messaging Service (**JMS**) is one of them. This tutorial is aimed at demonstrating how JMS protocol can be used in Web NMS configuration and fault management.

## Application Overview

This application is aimed at invoking the operations in Shopping Cart application and get the result for possible operations. Receive the notifications from the Shopping Cart project and convert it to events/alerts to store in the database

This tutorial is in four parts

1. OSS (an RMI Client) to initiate configuration
2. Web NMS to handle the network management functionality (Configuration and Fault Management)
3. JBoss application to provide the JMS support
4. JMX Agent to execute the configuration tasks

## Handling Configuration

Invoke OSS, prepare the configuration task in the XML format and submit it. The XML document consists of the following information about the entity to be configured.

- Device
  - destination (object name of the MBean of the JMX agent should be given)
  - Operation
    - Name
    - Parameter
      - Name
      - Value
      - Type

Web NMS receives the XML document and converts it into configuration tasks. Further the tasks are converted to JMS messages by the JMS configuration provider. Then the JMS message is published in the Java Messaging Service of JBoss application server. JMX agent, which has a JMS adaptor, picks up the message from the JBoss server. It parses the message with the help of JMS adaptor and carries out the configuration task in the Shopping Cart application running in it.

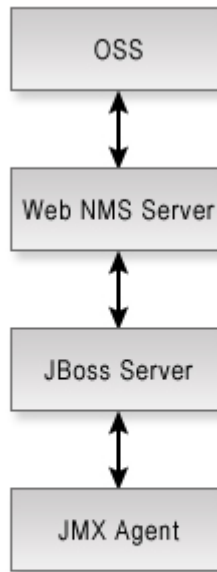
## Fault Management

In the Shopping Cart application, there is a default rule. The rule is, if any customer purchases items worth more than \$100, a discount will be offered. If your configuration task meets the condition of this rule then a notification message is published by the JMX agent. This notification is picked up by Web NMS and converts it into an Info Event and stores it in the database.

Apart from the default rule, we will be adding a rule which will notify if the number of items purchased exceed 50. If your configuration task meets the condition of this rule then a notification message is published by the JMX agent. This notification is picked up by Web NMS and converts it into an Major Event (Alert) and stores it in the database.

You can access the Web NMS database and check the result events.

## Application Design



**JMS Southbound Tutorial  
Application Architecture**

### Operations Support System (OSS)

- Send XML based configuration requests to Web NMS Server
- An RMI application sending XML information will function as OSS

### Web NMS Server

- Receives XML requests from OSS
- Interprets and translates XML request to configuration tasks
- Config Provider will convert task into a JMS request
- JMS message is sent to JMS server via JMS provider
- Publish configuration message to JMS Server
- Receive configuration result messages via JMS Provider
- Receive other asynchronous JMS messages
- Receives configuration results and stores in an object. Whenever OSS requests for the configuration status it will a file in the XML format based on the result stored in the object.
- Convert application fault notifications into Web NMS Events and store

### JBoss Application Server

- Hosts JMS Server
- Messaging server to post/publish/pickup/subscribe messages

### JMX Agent

- Receives JMS messages for which it has subscribed
- Convert JMS messages into MBean request
- Execute the configuration command on the Application components through MBeans
- Publish task execution notification (success or failure) to JMS Server
- Publish other application fault notifications to JMS Server

## Installation

### Web NMS 4.5.0

#### Installation

1. Download **Web NMS 4.5.0** from <http://adventnet.com/products/webnms/download.html>
2. Install Web NMS 4.5.0
3. JMS tutorial zip should be extracted on Web NMS Home if tutorials are unchecked during NMS installation
4. Start the Deployment Wizard to install the NAR *JMSSB\_Tutorial1.0.nar*, using the script `<Web NMS Home>/bin/DeploymentWizard.bat/sh`

#### Initialization

- Edit **jndi.properties** file present in the `<Web NMS Home>/classes` directory and change the `java.naming.provider.url` to the point to the JBoss server machine.  
Sample **jndi.properties**

```
java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
java.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interfaces
# Do NOT uncomment this line as it causes in VM calls to go over RMI!
java.naming.provider.url=<JBoss HOST>
```

### JBoss 3.2.1

#### Installation

1. Download JBoss Application server from <http://www.jboss.org>
2. Install JBoss 3.2.1 in a separate machine
3. Specify the JDK Home in `<JBoss Home>/bin/run.bat` file

#### Creating and Deploying the SAR file in JBOSS

##### Creating the SAR File

Create an XML file with the following contents and name it as **jboss-service.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
  <server>
    <mbean code="org.jboss.mq.server.jmx.Topic"
      name="jboss.mq.destination:service=Topic,name=Message">
      <depends optional-attribute-
        name="DestinationManager">jboss.mq:service=
        DestinationManager</depends>
    </mbean>
    <mbean code="org.jboss.mq.server.jmx.Topic"
      name="jboss.mq.destination:service=Topic,name=
      Notification">
      <depends optional-attribute-
        name="DestinationManager">jboss.mq:service=
        DestinationManager</depends>
    </mbean>
  </server>
```

Create a JAR file which should contain the **jboss-service.xml** in such a way that when you extract the JAR, the path of the file should be `META-INF/jboss-service.xml` and name the created JAR file as **JMSCommunication.sar**

## Deploying the SAR File

Copy the created **JMSCommunication.sar** file under the directory `<JBoss Home>/server/default/deploy`

## Agent

We will be using the Shopping Cart application with JMX agent

### Installing the JMX Studio

1. Download JMX Studio 5.1.1 from [http://www.manageengine.com/products/jmx\\_studio/download.html](http://www.manageengine.com/products/jmx_studio/download.html)
2. Unzip JMX Studio 5.1.1

**Note:** Supported JDK version is JDK 1.3.1.

## Operations Support System

### Installation

- Edit **run.bat** file present in `<OSS Home>/bin` directory and specify the JDK HOME (Please make sure you use JDK1.3.1)

### How to Start

The System has to be started in the following order JBoss Server, Agent in JMX Studio, Web NMS Server and then the Operations Support System. This is very important for getting proper result.

### JBoss Application Server

- Execute **run.bat/sh** file present in `<JBoss Home>/bin` directory (Please make sure JDK\_HOME is set prior to starting, use JDK1.3.1)

### Agent

Define a User Rule and start the Shopping Cart agent using the procedure given below:

1. Start the JMX studio by executing the `<JMX Studio Home>/bin/JMXStudio.bat/sh`
2. Open the ShoppingCart project by clicking the link labeled **Shopping Cart Project** in the right hand side panel
3. Select **Settings > Adaptor Settings** menu item
4. Select the JMS tab and enable the Enable JMS Adaptor checkbox
5. In **Provider URL** text field, type the name of the machine where the JBoss server is running and then click **OK**. In the **Initial Context Factory** text field enter the value `org.jnp.interfaces.NamingContextFactory`.

### Adding User Defined Rule

- a. In the left hand side tree select **MBean Tree > ShoppingCartMBean > Attributes > TotalItemCount** and right click on it
- b. Click the **Add Rule** menu item in the pop up menu
- c. In the *Add Rule* dialog, enter any name in the first screen and then click **Next**
- d. Select the default condition in the table and click **Modify**
- e. In the *Condition Dialog* select **greater than** in the comparator combo box
- f. Enter 50 in the *Compare Value* dialog and then click **OK** and then click **Next**
- g. In the *Action Configuration* window click on **Add** button
- h. In the *New Action*, enter a name in the **Action Name** text field and select **Send a Notification** in the **Action Type** combo box

- i. Type the message ***A customer has done a purchase of more than 50 items at a time.*** in the Message text field and then click the **OK**
  - j. Finally click the **Finish** button
6. Select **Build > Generate Agent** menu item
  7. Select **Build > Build Agent** menu item
  8. Close the JMX Studio
  9. Edit **run.bat/sh** in the *<JMX Studio Home>\example\shoppingcart\ShoppingCartProject\bin* directory and add the following jar files in the classpath

```
jboss-jmx.jar, log4j.jar, jboss-j2ee.jar, jbossmq.jar, jboss.jar,
jnpserver.jar, jboss-system.jar, jboss-common.jar, jboss-common-
client.jar, concurrent.jar, xml-apis.jar, jnet.jar
```

The above mentioned jar files are available in JBoss application server installation

10. Now start the JMX agent, by executing the **run.bat/sh** file which was added with classpath settings

### Web NMS

1. Execute **reinitialize\_nms.bat** present in *<Web NMS Home>/bin* directory to clean the database.
2. Execute **startnms.bat** file present in *<Web NMS Home>/bin* directory

### Operations Support System

- Execute **run.bat** file present in *<OSS Home>/bin* directory (Please make sure JDK\_HOME is set prior to starting, use JDK1.3.1) pass SOURCE\_ID and Web NMS Server Machine Name/IP Address as arguments  
**Usage:** *run <SOURCE\_ID> <Web NMS Server Name>*

SOURCE_ID	- Unique string that identifies the Operations Support System (can be name of the machine where the superior system is running)
Web NMS Server Name	- Name of the machine where Web NMS Server is running (Can be IP Address of the machine also)

## Testing

### How to execute a Configuration Task

You have to initiate the configuration task from the OSS. OSS comes with 4 options

1. Execute a configuration task
2. Get result for a particular message
3. Get result for all the messages that was sent from this superior system
4. Quit

### Execute a configuration task

The *Execute Configuration Task* choice takes an XML file as argument, which contains the details of the Configuration task. The XML files for configuration tasks should be present in the *<OSS Home>/conf* directory.

Specify Option 1 and provide XML name as **additem.xml** for sending it from OSS to Web NMS

**Note:** Care should be taken to specify unique ID for the message in XML file

### Get result for a particular message

Specify Option 2 and pass the Message ID for which you want the result  
Format of the Result XML Message

```
<?xml version="1.0" ?>
  <Result MessageID="12" SourceID="neena">
    <Operation OperationName="updateItem" Status="Success" />
  </Result>
```

### Get result for all messages from this OSS

Specify Option 3 and you will receive result for all the messages that were executed from this OSS.

## Notification simulation

### Task1

Carryout a configuration task. See that the configuration task action meets the condition of the default rule. A notification message will be sent by the agent. The notification contains a discount message stating that ***This customer has made purchases exceeding 100 dollars. Offer a discount.*** Open the Web NMS database and you will find the Info Event containing the discount message.

### Task2

Carryout a configuration task. See that the configuration task action meets the condition of the rule defined by you. A notification message will be sent by the agent. The notification contains a information message stating that ***A customer has done a purchase of more than 50 items at a time.*** Open the Web NMS database and you will find the Major Event (Alert) containing the discount message.