SeeBeyond ICAN Suite

WebSphere MQ eWay Intelligent Adapter User's Guide

Release 5.0.1

SᴇᴇBᴇʏᴏɴᴅ®

Contents

Introducing the WebSphere MQ eWay

This chapter introduces you to SeeBeyond™ eWay Intelligent Adapter for WebSphere MQ™. It includes an overview of this manual and a list of system requirements for installation.

1.1 Overview

WebSphere MQ (formerly MQSeries™) from IBM™ is a client-server message broker supporting an open API (application programming interface), available on a variety of operating systems including AIX™, Solaris™, HP-UX™, and Windows™. WebSphere MQ is "middleware" that provides commercial messaging and queuing services. Messaging enables programs to communicate with each other via messages rather than direct connection. Messages are placed in queues for temporary storage, freeing up programs to continue to work independently. This process also allows communication across a network of dissimilar components, processors, operating systems, and protocols.

The SeeBeyond eWay Intelligent Adapter for WebSphere MQ (referred to as the MQSeries eWay throughout this document) allows the eGate system to exchange data with IBM's WebSphere MQ. The eGate Integrator, using the MQSeries eWay, uses business logic within a Collaboration or Business Process to perform operations for data identification, manipulation, and transformation. Messages are tailored to meet the communication requirements of specific applications or protocols. Queues or Topics provide non-volatile storage for data within the eGate system allowing applications to run independently of one another at different speeds and times.

The MQSeries eWay transparently integrates existing systems with IBM's WebSphere MQ. This document explains how to install and configure the MQSeries eWay.

1.2   Supported Operating Systems

The MQSeries eWay is available on the following operating systems. See any notes below regarding your specific operating system.

- Windows 2000 SP3
- Sun Solaris 8 and 9
- AIX 5.1 and AIX 5.2
- HP-UX 11.0 and HP-UX 11i (RISC)
- HP Tru64 5.1A
- Red Hat Linux 8 (Intel)
- Red Hat Linux Advanced Server 2.1 (Intel)

*For **AIX** operating systems, the environmental variable LDR_CNTRL for JVM may need to be adjusted in order to accommodate WebSphere MQ shared memory. Java uses 8 segments by default (this is the maximum value allowed; each segment is 256 MB). For example, the following setting changes the number of segments to 3:*

```
setenv LDR_CNTRL MAXDATA=0x30000000
```

*For **HP-UX 11** operating systems, HP-UX Java binding support is only available for systems running the POSIX draft 10 threaded version of WebSphere MQ. The HP-UX Developers kit for Java 1.1.7, Release C.01.17.01 or above is also required.*

1.3   System Requirements

To use the MQSeries eWay, you need the following:

- A TCP/IP network connection.
- Windows XP SP1a, Windows 2003 Server, or Windows 2000 SP3. This is required for the User Interface.
- Microsoft Internet Explorer 6.0 SP1 or above.

*Note: Open and review the **Readme.txt** for any additional requirements prior to installation.*

1.3.1. External System Requirements

The MQSeries eWay requires the following installed on the participating host:

- IBM WebSphere MQ V5.3 or MQSeries V5.2 with SP MA88

Note: Install either the WebSphere MQ server or the WebSphere MQ client code. If the WebSphere MQ server is installed on the participating host, the eWay connects to the server in bindings mode. If the WebSphere MQ client code is installed on the

participating host, the eWay connects to the server in client mode. A client mode connection cannot participate in XA transactions.

Install the following after installing IBM MQSeries V5.2:

- IBM MQSeries classes for Java 5.2.0.

- Classes for Java Message Service 5.2.0.0

WebSphere MQ V5.3 includes the WebSphere MQ classes for Java and JMS. The use of SupportPac MA88 with WebSphere MQ V5.3 product is not supported. MQSeries V5.2 requires the installation of SupportPac MA88 for all supported platforms.

The **MA88 SupportPac** download and installation information can be found at: **http://www-4.ibm.com/software/ts/mqseries/txppacs/ma88.html.**

The MA88 patch includes updates for several jar files and DLL's/shared libraries. Most notably, **com.ibm.mq.jar**, **mqjbnd02.dll** and **mqxai01.dll**. It is important that the patch overwrites the existing versions of these files if they are present on your machine. Alternatively, if they do not overwrite the existing versions, it is important that the new versions of these files exist on your classpath and path before the old versions. Once you have downloaded the SupportPac, make sure that all .jar files installed as part of the SupportPac are included in the classpath.

Requirements for the Topic Publish/Subscribe Connection Type

IBM **SupportPac MAOC** is required by both MQSeries V5.2 and WebSphere MQ V5.3. The SupportPac MAOC installation information and download can be found at: **http://www-3.ibm.com/software/ts/mqseries/txppacs/ma0c.html**

Installing the WebSphere MQ eWay

This chapter explains how to install the MQSeries eWay.

Chapter Topics

2.1 Installing the MQSeries eWay

During the eGate Integrator installation process, the Enterprise Manager, a web-based application, is used to select and upload eWays (eWay.sar files) from the eGate installation CD-ROM to the Repository.

When the Repository is run on a UNIX operating system, the eWays are loaded using the Enterprise Manager on a Windows computer connected to the Repository server using Internet Explorer.

Before installing the eWay

Open and review the **Readme.txt** for any additional information or requirements, prior to installation.

Installing the MQSeries eWay on an eGate supported system

The MQSeries eWay is installed during the installation of the eGate Integrator. The eGate installation process includes the following operations:

- Installing the eGate Repository
- Uploading products to the Repository
- Downloading components (including eGate Enterprise Designer and Logical Host)
- Viewing product information home pages

Follow the instructions for installing the eGate Integrator in the eGate Integrator Installation Guide, and include the following steps:

1 During the procedures for uploading files to the eGate Repository using the Enterprise Manager, after uploading the **eGate.sar** file, select and upload the following files:

- ◆ **MQSerieseWay.sar** (to install the MQSeries eWay)
- ◆ **FileeWay.sar** (to install the File eWay, used in the sample Project)
- ◆ **WebSphereMQeWay.sar** (to download the WebSphere MQ eWay user's guide)

2 Continue installing the eGate Integrator as instructed in the *eGate Integrator Installation Guide.*

After Installation

Once the eWay is installed and configured it must then be incorporated into a Project before it can perform its intended functions. See the *eGate Integrator User's Guide* for more information on incorporating the eWay into an eGate Project.

Configuring the eWay Properties

This chapter describes how to create and configure the MQSeries eWay.

Chapter Topics

3.1 Creating and Configuring the MQSeries eWay

All eWays contain a set of parameters with properties unique to that eWay type. After the eWays are established and an MQSeries External System is created in the Project's Environment, the eWay parameters can be modified for your specific system. The MQSeries eWay properties are modified from two locations:

- From the **Connectivity Map**. These properties most commonly apply to a specific eWay, and may vary from other eWays (of the same type) in the project. The MQSeries Connectivity Map properties will vary depending on whether the MQSeries is an Inbound or Outbound eWay

- From the **Environment Explorer tree**. These properties are commonly global, applying to all eWays (of the same type) in the project. The saved properties are shared by all eWays in the MQSeries External System window.

The properties for the MQSeries eWay must be set in both locations.

3.1.1 Selecting WebSphere MQ as the External Application

To create a MQSeries eWay, you must first create an MQSeries External Application in your Connectivity Map. MQSeries eWays are located between an MQSeries External Application and a Service. Services are containers for Java Collaborations, Business Processes, eTL processes, and so forth.

To create the MQSeries External Application

1 From the Connectivity Map toolbar, click the **External Applications** icon.

2   Select the **MQSeries External Application** from the menu (see Figure 1). The selected MQSeries External Application icon appears on the Connectivity Map toolbar.

Figure 1   External Applications Selection Menu
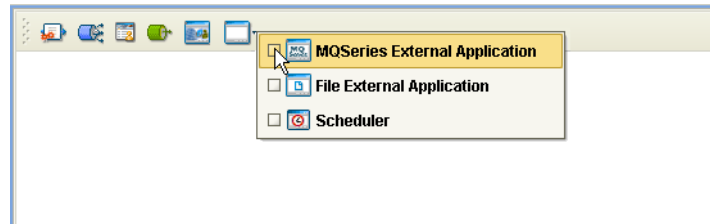
3   Drag the new **MQSeries External Application** from the toolbar onto the Connectivity Map canvas. This represents an external WebSphere MQ system.

From the Connectivity Map, you can associate (bind) the External Application with the Service to establish an eWay (see Figure 2).

Figure 2   eWay Location
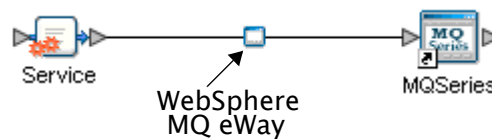
When MQSeries is selected as the External Application, it automatically applies the default MQSeries eWay properties, provided by the OTD, to the eWay that connects it to the Service. These properties can then be or modified for your specific system using the **Properties Sheet**.

3.1.2 Creating Custom Properties for a MQSeries eWay

A Project's eWay properties can be modified after the eWays have been established in the Connectivity Map and the Environment has been created.

Modifying the MQSeries eWay (Connectivity Map) Properties

1   From the Connectivity Map, double click the eWay icon, located in the link between the associated External Application and the Service. A Templates dialog box appears.

2   From the Templates dialog box, select **Inbound** or **Outbound MQSeries eWay** and click **OK**.

3   The eWay **Properties Sheet** opens to the Inbound or Outbound MQSeries Connectivity Map properties. Make any necessary modifications and click **OK**.

Modifying the MQSeries eWay (Environment Explorer) Properties

1   From the Environment Explorer tree, right-click the Inbound or Outbound WebSphere MQ external system. Select **Properties** from the shortcut menu. The **Properties Sheet** appears.

2   Make any necessary modifications to the Environment parameters of the MQSeries eWays, and click **OK** to save the settings.

3.1.3. Using the Properties Sheet

Modifications to the eWay configuration properties are made from the MQSeries eWay Properties Sheet.

To modify the default eWay configuration properties

1   Open the Properties Sheet to the MQSeries eWay default properties. An eWay has two different sets of parameters: those specific to that particular eWay (accessed from the Connectivity Map), and those that are common to all eWays of this type (accessed from the Environment Explorer tree).

2   From the upper-right pane of the Properties Sheet, select a section of the configuration directory. The parameters contained in that section are now displayed in the Properties pane of the Properties Sheet. For example, clicking on the **MQSeries** subdirectory displays the editable parameters in the right pane, as shown in Figure 3.

Figure 3   Properties Sheet -- WebSphere MQ Properties
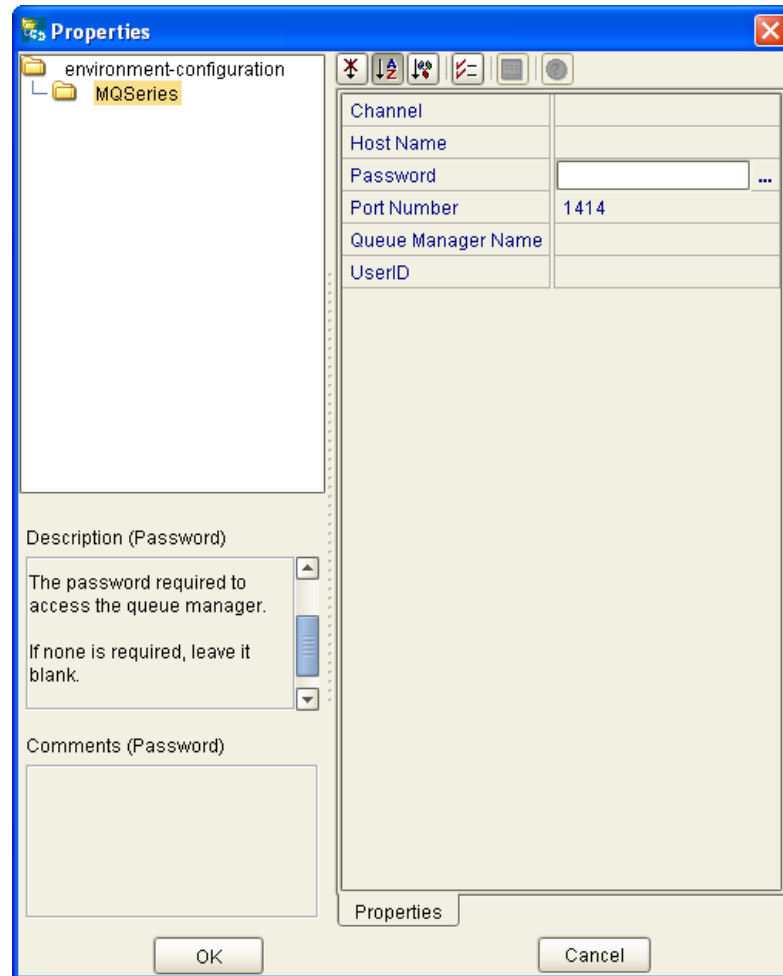
3   Click on any property field to make it editable. For example, click on the **Password** parameter to edit the Password value. If a parameter's value is true/false or multiple choice, the field, when selected, reveals a submenu of property options .

Click on the ellipsis (. . .) in the properties field to open a separate configuration dialog box. This is helpful for large values that cannot be fully displayed in the parameter's property field. Enter the property value in the dialog box and click **OK**. The value is now displayed in the parameter's property field.

4   A description of each parameter is displayed in the **Description** pane when that parameter is selected, providing an explanation of any required settings or options.

5   The **Comments** pane provides an area for recording notes and information regarding the currently selected parameter. This is saved for future referral.

6   After modifying the configuration properties, click **OK** to close the Properties Sheet and save the changes.

3.2   Outbound MQSeries eWay Connectivity Map Properties

The MQSeries eWay parameters, accessed from the Connectivity Map, are organized into the following sections:

- **General Settings** on page 15
- **Connector** on page 16

3.2.1. General Settings

This section contains the following parameters:

- **Transaction Type** on page 15

Transaction Type

Note: The XA-compliant Transaction Type is not supported for the WebSphere MQ eWay, version 50.

Description

String-set. Specifies the Transaction Type. There are two transaction types.

- **Non-Transactional**. Provides the highest level of performance, with the minimum level of message protection. No rollback is available during the send and receive period, causing the possible loss of data in the case of a system error. When the send() method is called the transaction is immediate. Non-Transactional (single-phase transaction) is subject to the user calling the commit and backout methods.

- **XA-compliant**. (two-phase transactional behavior) Highest level of transaction protection, providing rollback for internal and XA compliant transactions. The

transaction is also extended to other XA supported data exchange applications, such as Oracle, DB2, and WebSphere MQ. When the send() method is called the transaction takes place at the end of the Collaboration.

XA can only be used in Binding mode. This means that a Host Name is not entered in the properties. A Host Name in the properties implies that the API is setup in Client mode. XA cannot be used in Client mode. In addition, when using XA:

- ◆ There must be only one queue manager.
- ◆ The method **connectToQueueManager** cannot be called.
- ◆ The methods **commit** and **backout** cannot be called. In Bindings mode, eGate is the transaction coordinator and is in control of these methods.

Note: Consult the XA Transaction Processing section of the eGate Integrator User's Guide for information on XA use and restrictions.

Required Value

- ▪ Select Non-Transactional or XA-compliant. Non-Transactional is the configured default.

3.2.2. Connector

This section contains the following top level parameters:

- ▪ **class** on page 16
- ▪ **Property.Tag** on page 17
- ▪ **type** on page 17

Note: The Connector properties are not applicable for WebSphere MQ eWay, version 50.

class

Note: The Connector - class property is not applicable for WebSphere MQ eWay, version 50.

Description

Specifies the connector class for MQSeriesOTD. The default value should always be used.

Required Value

The configured default is **com.stc.eways.MQSeriesOTD.MQSeriesConnector**.

Property.Tag

Note: *The Connector - Property.Tag property is not applicable for WebSphere MQ eWay, version 50.*

Description

Specifies the data source identity. This parameter is required by the current EBobConnectorFactory.

Required Value

The data source package name.

type

Note: *The Connector - type property is not applicable for WebSphere MQ eWay, version 50.*

Description

Specifies the connector type for WebSphere MQ OTD. The default value should always be used.

Required Value

The connector type. The default value should always be used. The configured default is **MQSeriesOTD**.

3.3 Inbound WebSphere eWay Connectivity Map Properties

The Inbound MQSeries eWay parameters, accessed from the Connectivity Map, are organized into the following sections:

- **Settings** on page 17

3.3.1. Settings

This section contains the following parameters:

- **Queue Name** on page 17
- **Schedule Interval** on page 18

Queue Name

Description

Specifies the IBM WebSphere MQ Queue name.

Required Value

The name of the WebSphere MQ Queue.

Schedule Interval

Description

Specifies the polling interval by the number of milliseconds at which the WebSphere MQ Queue is polled for messages.

Required Value

The number of milliseconds at which the WebSphere MQ Queue is polled. The configured default is 5000.

3.4 MQSeries eWay Environment Explorer Properties

The MQSeries eWay parameters, accessed from the Environment Explorer tree, are organized into the following sections:

- **MQSeries** on page 18

3.4.1. MQSeries

This section contains a set of top level parameters:

- **MQ Channel Name** on page 18
- **MQ Host Name** on page 19
- **MQ Queue Manager Name** on page 19
- **Password** on page 19
- **Port Number** on page 19
- **UserID** on page 19

MQ Channel Name

Description

Specifies the name of the channel being used. This option is only relevant with transport type 'Client' and is ignored for transport type 'Bindings'. If no channel is specified the default channel is used.

Required Value

The name of the channel.

MQ Host Name

Description

Specifies the name of the host on which the Queue Manager resides. This option is only relevant with transport type "Client" and is ignored for transport type "Bindings". As for "Bindings", the Queue Manager must be on the same host.

Required Value

The name of the specific Queue Manager host.

MQ Queue Manager Name

Description

Specifies the name of the IBM WebSphere MQ Queue Manager.

Required Value

The name of the WebSphere MQ Queue Manager.

Password

Description

Specifies the user password required to access the Queue Manager. This parameter is optional.

Required Value

A user password that grants access to a specific WebSphere MQ Queue Manager.

Port Number

Description

Specifies the number of the port to connect to. This option is only relevant with a transport type 'Client' and is ignored for transport type 'Bindings'. If this option is left empty the default port is used.

Required Value

A port number, in the range of 1000 and 65536. The configured default is 1414.

UserID

Description

Specifies the User ID required to access the WebSphere MQ Queue Manager. This parameter is optional.

Required Value

A User ID required for the Specific WebSphere MQ Queue Manager.

Using the WebSphere MQ eWay With eInsight

This chapter describes how to use the MQSeries eWay with the ICAN Suite's eInsight Business Process Manager and its engine's Web Services interface.

*Note: You must have the **eInsight.sar** file installed to use the Web Services interface.*

Chapter Topics

4.1   eInsight Engine and Components

eGate components can be deployed as Activities in eInsight Business Processes. Once a component is associated with an Activity, eInsight invokes it using a Web Services interface. eGate components that can interface with eInsight in this way include the following:

- Java Messaging Service (JMS)
- Object Type Definitions (OTDs)
- eWays
- Collaborations

Using the Enterprise Designer and eInsight, you can add an Activity to a Business Process, then associate that Activity with an eGate component, for example, an eWay. Then, when eInsight runs the Business Process, it automatically invokes that component via its Web Services interface.

See the *eInsight Business Process Manager User's Guide* for details.

4.2 The MQSeries eWay With eInsight

An eInsight Business Process Activity can be associated with the MQSeries eWay during the system design phase. To make this association, select the desired operators under the eWay in the Enterprise Explorer and drag it onto the eInsight Business Process Designer canvas.

The MQSeries eWay has the following operators available:

- **receive**
- **mqget**
- **mqput**

The operation is automatically changed to an Activity with an icon identifying the component that is the basis for the Activity. At run time, eInsight invokes each step in the order defined in the Business Process. Using eInsight's Web Services interface, the Activity in turn invokes the MQSeries eWay.

4.3 MQSeries eWay eInsight Sample Projects

Two sample projects for the MQSeries eWay using eInsight Business Processes, are included with the installation. This chapter also walks you through the creation of these sample Projects:

- **The MQ_BP_Get_ Sample Project** on page 22
- **The MQ_BP_Put_ Sample Project** on page 33

4.4 Importing a Sample Project

Sample eWay Projects are included as part of the installation CD-ROM package.

To import a sample eWay Project to the Enterprise Designer do the following:

1 The sample files are uploaded with the eWay's documentation .sar file and downloaded from the Enterprise Manager's Documentation tab. Extract the samples from the Enterprise Manager to a local file.

2 From the Enterprise Designer's Project Explorer pane, right-click the Repository and select **Import Project** from the shortcut menu. The **Select File to Import** dialog box appears.

3 Browse to and select the sample file (for this sample, **WebSphereMQSeries_BPEL_Get_Sample.zip**) and click **Open**.

4 From the File Destination dialog box, select **Import to a new Project**, enter the name of the Project (for this sample, **MQ_BP_Get_ Sample**), and click **OK**.

5   When the import has successfully completed, click the **Refresh All from Repository** icon from the Enterprise Designer toolbar.

6   Before an imported sample Project can be run you must do the following:

  ◆ Install and configure a **WebSphere MQ Queue** on the localhost

  ◆ Create an **Environment** (see **Creating an Environment** on page 29)

  ◆ Configure the eWay properties for your specific system (see **Configuring the eWay Properties** on page 30)

  ◆ Create a **Deployment Profile** (see**Creating and Activating the Deployment Profile** on page 31)

4.5   The MQ_BP_Get_ Sample Project

The **MQ_BP_Get_ Sample** Project demonstrates the following:

1   The MQSeries eWay, in receive mode, retrieves a message from an MQ Queue. The message is published to an outbound File eWay.

2   The File eWay published the message to an external directory.

The following pages provide step by step directions for manually creating the **MQ_BP_Get_ Sample** Project.

4.5.1.   Creating a Project

The first step is to create a new Project in the SeeBeyond Enterprise Designer.

1   Start the Enterprise Designer.

2   From the Enterprise Explorer's Project Explorer tab, right-click the Repository and select **New Project** (see Figure 4). A new Project (Project1) appears on the Project Explorer tree.

Figure 4   Enterprise Explorer - New Project
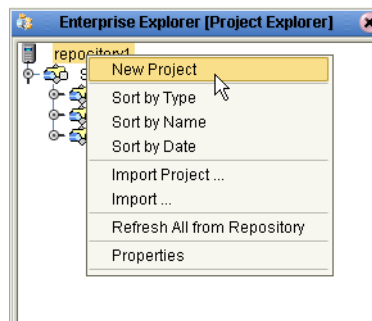
3   Click twice (not double-click) on **Project1** and rename the Project (for this sample, **MQ_BP_Get_ Sample**).

4.5.2 Creating the BusinessProcess_MQGet Business Process

Creating the Business Process Flow

1 From the Enterprise Designer's Project Explorer tree, right-click **MQ_BP_Get_ Sample**, and select **New** > **Business Process** from the shortcut menu. The eInsight Business Process Designer appears and **BusinessProcess1** is added to the Project Explorer tree. Rename the Business Process to **BusinessProcess_MQGet**.

2 From the Project Explorer tree, expand the **SeeBeyond > eWays > MQSerieseWay > MQSeries** nodes. In the same way, expand and **SeeBeyond > eWays > File > FileClient** nodes to expose the available Business Process elements.

3 Populate the eInsight Business Process Designer's canvas with the following elements from the Project Explorer tree, as displayed in Figure 5:

   ◆ **receive**, under SeeBeyond > eWays > MQSerieseWay > MQSeries

   ◆ **write**, under SeeBeyond > eWays > File > FileClient

Figure 5   eInsight Business Process Designer - Populate the Canvas
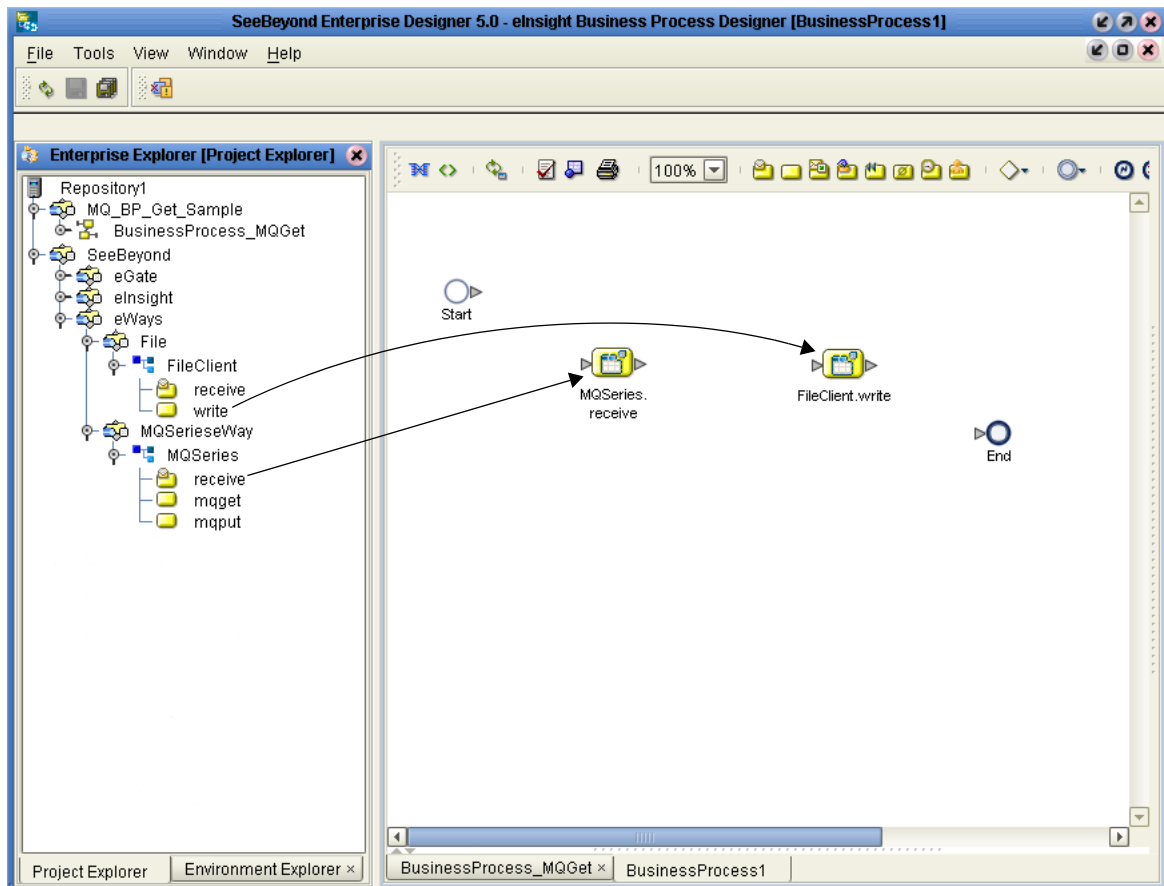
4 Link the modeling elements by clicking on the element's connector and dragging the cursor to the next element's connector, making the following links as displayed in Figure 6.

- ◆ Start -> MQSeries.receive
- ◆ MQSeries.receive -> FileClient.write
- ◆ FileClient.write -> End

Figure 6 eInsight Business Process Designer - Link the Modeling Elements
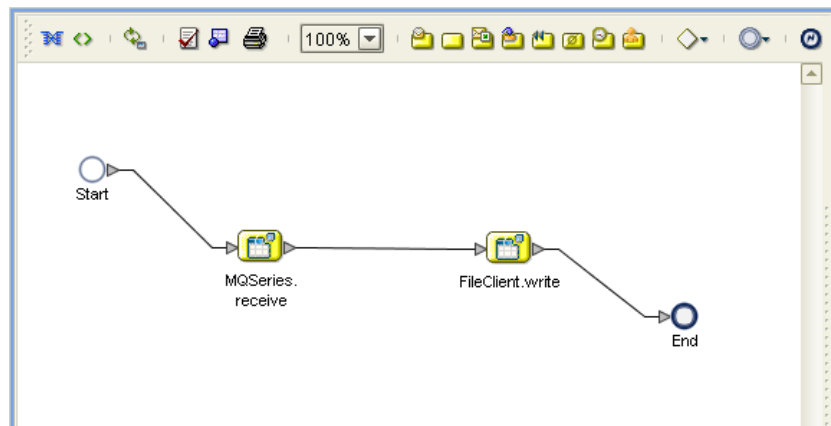
Configuring the Modeling Elements

Business Rules, created between the Business Process elements, define the relationship between the input and output Attributes of the elements.

1 Right-click the link between the **MQSeries.receive** and **FileClient.write** Activities and select **Add Business Rules** from the shortcut menu as displayed in Figure 7.

Figure 7 eInsight Business Process Designer - Adding Business Rules
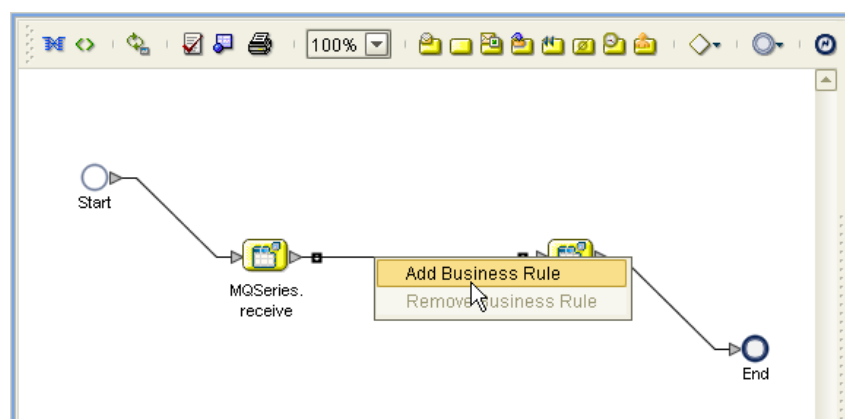
2 From the eInsight Business Process Designer toolbar, click the **Map Business Process Attributes** button. The Business Rule Designer appears at the bottom of the eInsight Business Process Designer.

3 Click on the **Business Rules** icon in the link between **MQSeries.receive** and **FileClient.write** to display the Business Rule Output and Input Attributes in the Business Rule Designer. These Attributes can now be modified.

4 From the Business Rule Designer toolbar, click the Method Palette icon (see Figure 8). The Method Palette appears. From the **String** tab of the Method Palette, select **bytes to text** and click **Close**. The **bytes to text** icon is added to the toolbar.

5 From the Business Rule Designer toolbar, drag and drop the **bytes to text** icon onto the Business Rule Designer canvas. The **bytes to text** method box appears (see Figure 8).

Figure 8   eInsight Business Rule Designer
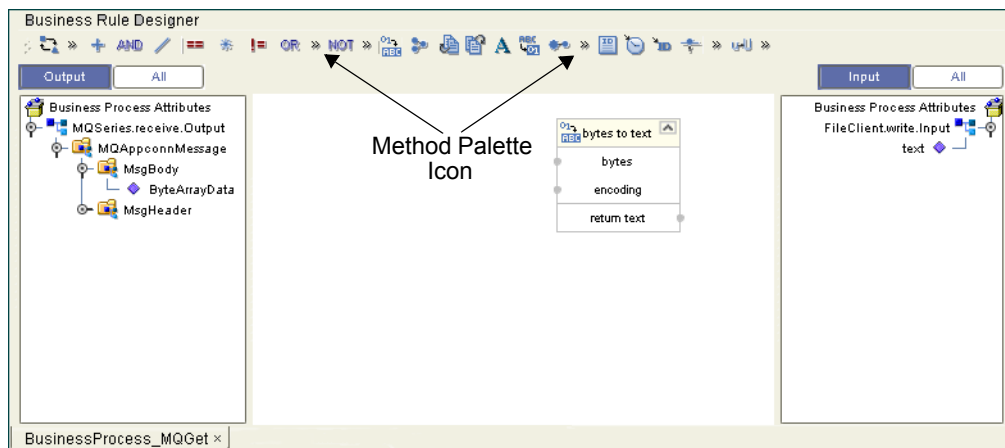
6 Map **ByteArrayData,** under **MQAppconnMessage > MsgBody** in the Output pane of the Business Rule Designer, to the **bytes** input node of the **bytes to text** method box. This is done by clicking on **ByteArrayData** and dragging the cursor to the **bytes** input node of the **bytes to text** method box.

7 Map the **return text** output node of the **bytes to text** method box to **text**, under **FileClient.write.input** in the Input pane of the Business Rule Designer. The Business Process Designer (see Figure 9).

Figure 9   eInsight Business Rule Designer
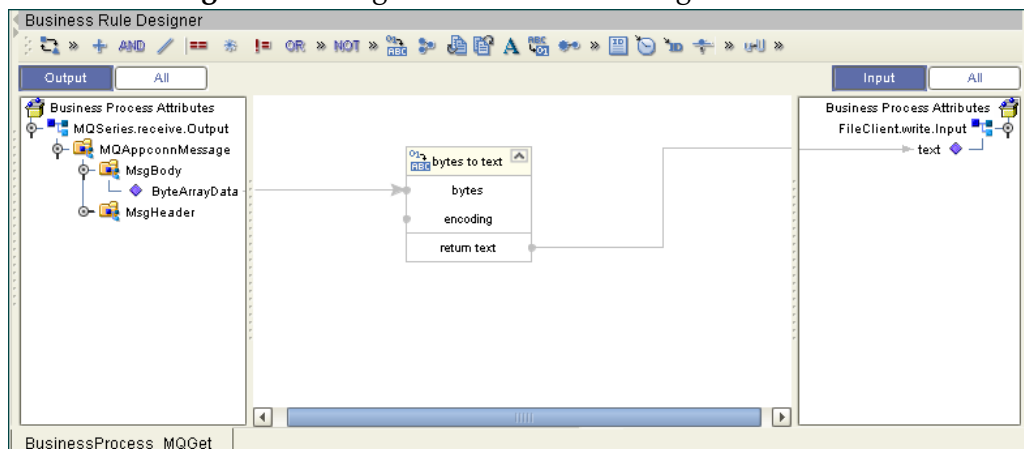
8 From the Business Process Designer toolbar, click the **Synchronize Graphical Model and Business Process Code** icon to synchronize the graphical interface to the Business Process code.

9 Click the Enterprise Designer's **Save All** icon to save your current changes.

4.5.3 Creating a Connectivity Map

The Connectivity Map provides a canvas for assembling and configuring a Project's components.

1 From the Project Explorer tree, right-click the **MQ_BP_Get_ Sample** Project and select **New > Connectivity Map** from the shortcut menu.

2 The new Connectivity Map appears and the **CMap1** (Connectivity Map) node is added Project Explorer tree. Rename the **CMap1** Connectivity Map to **CMap_BPEL_MQGet**.

Select the External Applications

In the Connectivity Map, eWays are associated with the External Applications. To create the External Applications used by the **MQ_BP_Get_ Sample** Project do the following:

1 Click the **External Application** icon on the Connectivity Map toolbar (see Figure 10).

The icons in the toolbar represent the available components used to populate the Connectivity Map canvas.

Select the External Applications

When creating a Connectivity Map, the eWays are associated with External Systems. For example, to establish a connection to WebSphere MQ, you must first select MQSeries as an External System to use in your Connectivity Map (see Figure 10).

Figure 10 Connectivity Map - External Applications
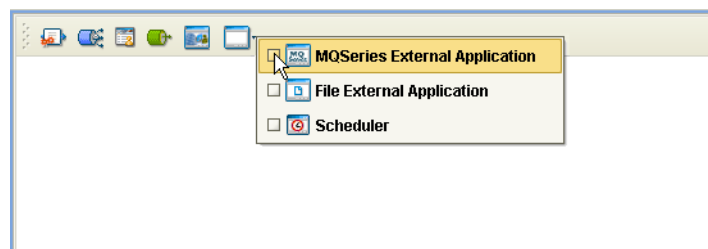
To create the External Applications used by the **MQ_BP_Get_ Sample** Project do the following:

1 Click the **External Application** icon on the Connectivity Map toolbar.

2 Select the applications needed for your Project (for this sample, the **File** and **MQSeries** External Applications). Icons representing the selected applications are added to the Connectivity Map toolbar.

Populate the Connectivity Map

Add the Project components to the Connectivity Map by dragging the icons from the toolbar to the canvas.

1 Drag the following components onto the Connectivity Map canvas as displayed in Figure 11:

◆ **MQSeries External Application**

◆ **Service** (A service is a container for Java Collaborations, Business Processes, eTL processes, and so forth) The Service may also be created by dragging the Business Process to the Connectivity Map canvas from the Project Explorer tree.

◆ **File External Application**

Figure 11 Connectivity Map with Components
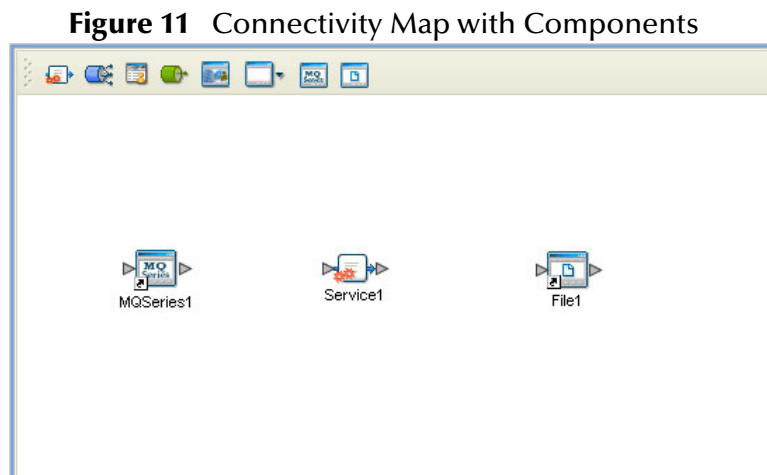
2 Rename the **MQSeries1** External Application to **MQSeriesGet** by right-clicking the object, selecting **Rename** from the shortcut menu, and typing in the new name.

3 Rename the **File1** External Application to **FileOut**.

4 Rename **Service1** to **BusinessProcess_MQGet1**.

5 Save your current changes to the Repository.

4.5.4. Binding the Project Components

Components are associated and the bindings are created from the Connectivity Map.

1 Open the **CMap_BPEL_MQGet** Connectivity Map.

2 Drag and drop the **BusinessProcess_MQGet** Business Process from the Project Explorer tree to the **BusinessProcess_MQGet1** service (see **Figure 12 on page 28**).

Figure 12 Connectivity Map - Binding the Components
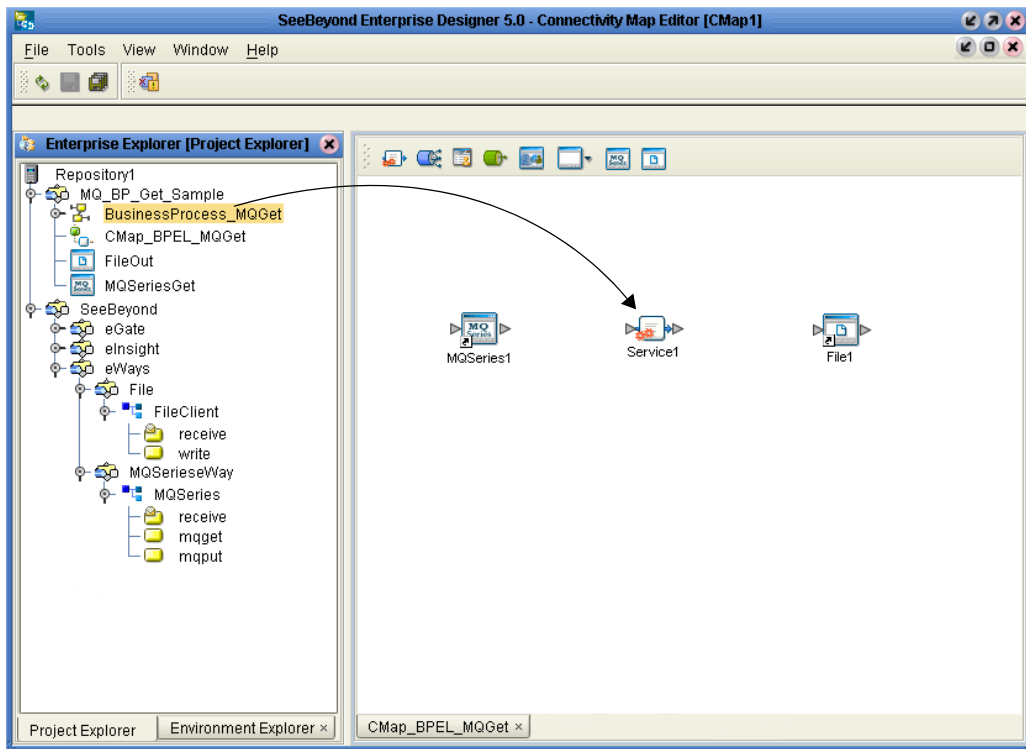
3   Double-click the **BusinessProcess_MQGet1** service. The
    BusinessProcess_MQGet1 binding box appears with the **BusinessProcess_MQGet**
    Rule.

4   From the **BusinessProcess_MQGet1** binding box, map **MQSeriesSender** (under
    Implemented Services) to the **MQSeriesGet** application.

5   From the **BusinessProcess_MQGet1** binding box, map **FileReceiver** (under
    Invoked Services) to the **FileOut** External Application (see Figure 13). .

Figure 13 Connectivity Map - Binding the Components
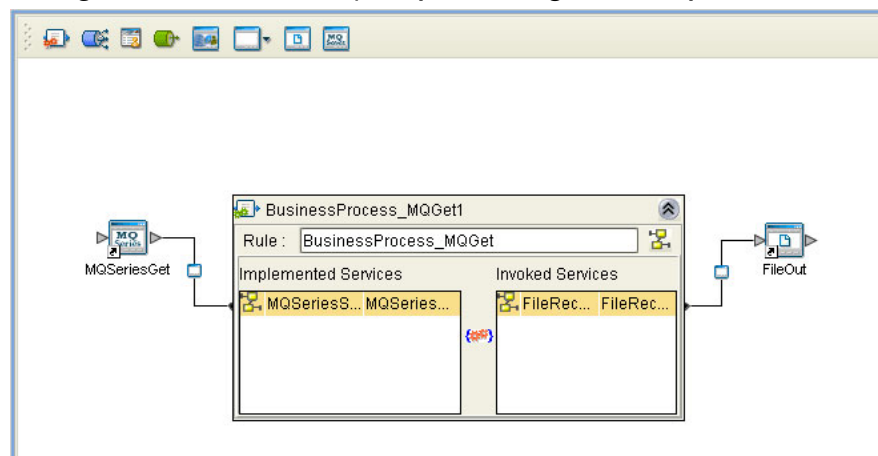
6   Minimize the **BusinessProcess_MQGet1** binding box by clicking the chevrons in
    the upper-right corner and save your current changes to the Repository.

4.5.5. Creating an Environment

Environments include the external systems, logical hosts, integration servers and JMS IQ Managers used by a Project and contain the configuration information for these components. Environments are created using the Enterprise Designer's Environment Explorer and Environment Editor.

1 From the Enterprise Designer's Enterprise Explorer, click the **Environment Explorer** tab.

2 Right-click the Repository and select **New Environment**. A new Environment is added to the Environment Explorer tree.

3 Rename the new Environment to **MQ_BP_Get_Sample_Env**.

4 From the Project Explorer tree, right-click **MQ_BP_Get_Sample_Env** and select **New File External System**. Name this External System **FileExtSysOut** and select **Outbound File eWay** as the External System Type. The **FileExtSysOut** window is added to the Environment Editor.

5 From the Project Explorer tree, right-click **MQ_BP_Get_Sample_Env** and select **New MQSeries External System**. Name this External System **MQExtSysIn** and select **Inbound MQSeries eWay** as the External System Type. The **MQExtSysIn** window is added to the Environment Editor.

6 From the Project Explorer tree, right-click **MQ_BP_Get_Sample_Env** and select **New Logical Host**. The **LogicalHost1** box is added to the Environment and **LogicalHost1** is added to the Environment Editor tree.

7 From the Environment Explorer tree, right-click **LogicalHost1** and select **New SeeBeyond Integration Server**. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under LogicalHost1.

8 Save changes to the Repository. The Environment Explorer and Environment Editor now appear as displayed in Figure 14.

Figure 14 Environment Editor
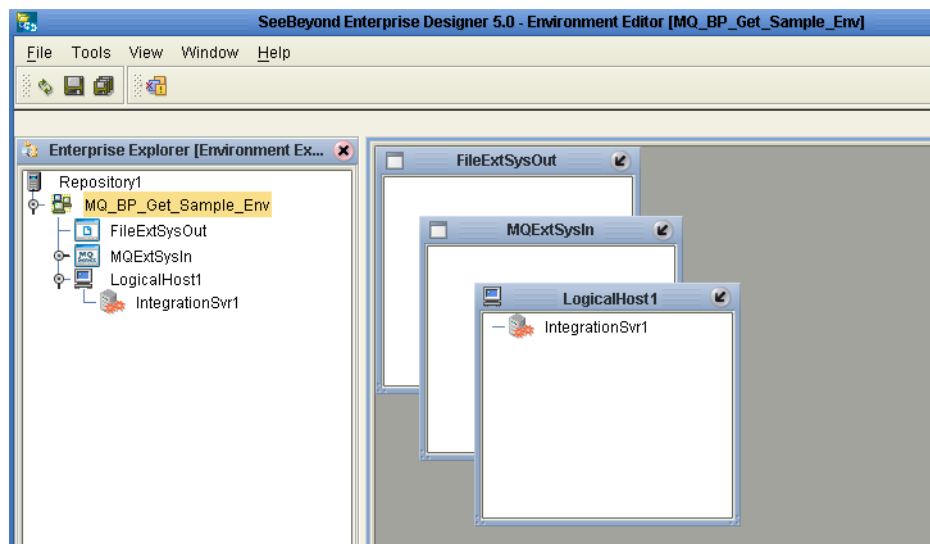
4.5.6. Configuring the eWay Properties

The **MQ_BP_Get_ Sample** Project contains two eWays, each represented in the Connectivity Map as a node between an External Application and a Service. eWays facilitate communication and movement of data between the external applications and the eGate system (see Figure 15).

Figure 15 eWay Configurations
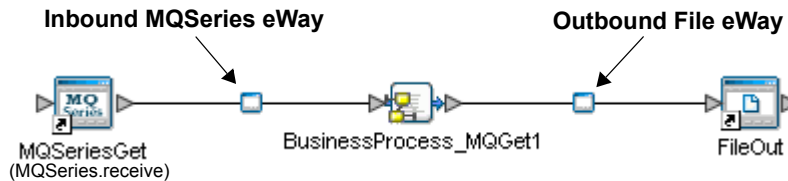
The File eWay properties are configured from the Connectivity Map. The MQSeries eWay properties are set from both the Project Explorer's Connectivity Map and the Environment Explorer tree. To configure the eWays do the following:

Configuring the File eWay Properties

1 Double-click the **Outbound File eWay**, select **Outbound File eWay** in the Templates dialog box and click **OK**.

2 The Properties Sheet opens to the Outbound File eWay properties. Modify the properties for your system, including the settings for the **Outbound File eWay** in Table 1, and click **OK**. The properties are saved for the eWay.

Table 1 Outbound File eWay Settings

Outbound File eWay Properties	
Directory	C:/temp
Output file name	output%.dat

Configuring the MQSeries eWay Properties

The MQSeries eWay properties must be set in both the Project Explorer and Environment Explorer. For more information on the MQSeries eWay properties and the Properties Sheet, see **Creating and Configuring the MQSeries eWay** on page 12 or see the *eGate Integrator User's Guide*.

Modifying the MQSeries eWay Connectivity Map Properties

1 From the Connectivity Map, double-click the **MQSeries** eWay. The Properties Sheet opens to the MQSeries eWay Connectivity Map properties.

2 Modify the **MQSeries** eWay properties for your system and click **OK**.

Modifying the MQSeries eWay Environment Explorer Properties

1 From the **Environment Explorer** tree, right-click the Inbound MQSeries External System (**MQSeriesExtSysIn** in this sample), and select **Properties**. The Properties Sheet opens to the MQSeries eWay environment properties.

2 Modify the **MQSeries** eWay environment properties for your system, including the settings in Table 2, and click **OK**.

Table 2 MQSeries Environment Explorer eWay Settings

MQSeries eWay Environment Explorer Properties	
MQSeries Set as directed, otherwise use the default settings.	
MQ Host Name	*The name of the specific Queue Manager host*
MQ Queue Manager Name	*The name of the WebSphere MQ Queue Manager*
Port Number	*Port number to use to connect to the MQ Queue Manager. The default is 1414.*

4.5.7 Creating and Activating the Deployment Profile

A Deployment Profile is used to assign Business Processes and message destinations to the integration server and JMS IQ Manager. Deployment Profiles are created using the Deployment Editor.

1 From the Enterprise Explorer's Project Explorer, right-click the Project (**MQ_BP_Get_ Sample**) and select **New** > **Deployment Profile**.

2 Enter a name for the Deployment Profile (for this sample **MQ_BP_Get_Sample_DP**). Make sure that the selected Environment is **MQ_BP_Get_Sample_Env.** Click **OK**.

3 From the left pane of the Deployment Editor, drag **BusinessProcess_MQGet1 -> FileOut** (External Application) to the **FileExtSysOut** window.

4 From the left pane of the Deployment Editor, drag the **MQSeriesGet -> BusinessProcess_MQGet1** (External Application) to the **MQExtSysIn** window.

5 From the left pane of the Deployment Editor, drag the **BusinessProcess_MQGet1** (Business Process) to **IntegrationSvr1** in the **LogicalHost1** window (see Figure 16).

Figure 16 Deployment Profile
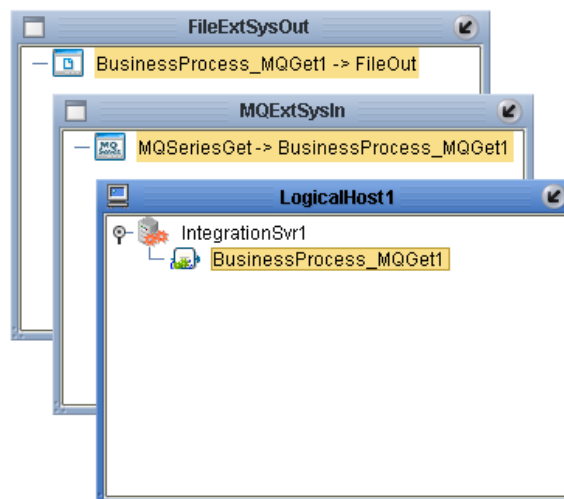
6   Click **Activate**. When activation succeeds, save the changes to the Repository.

4.5.8. Running the Project

The following directions assume that the Enterprise Designer was downloaded to C:\ican50. If this is not the case, replace that location in the following directions with the appropriate location.

1   From the Enterprise Manager Downloads tab, download **Logical Host - for win32**.

2   Extract the file to the **ican50\LogicalHost1** directory. You must specify the **LogicalHost1** directory for it to be created.

3   Navigate to **C:\ican50\LogicalHost1\bootstrap\config** directory and open the **logical-host.properties** file using Notepad™.

4   Enter the following information in the appropriate fields:

   ◆ Logical Host root directory: **ican50\LogicalHost1**

   ◆ Repository URL: **http://localhost:***port number/repository name*

   ◆ Repository user name and password: *Your user name and password*

   ◆ Logical Host Environment name: **MQ_BP_Get_Sample_Env**

   ◆ Logical Host name: **LogicalHost1**

Save your changes to **logical-host.properties** and close the file.

5   Run the **bootstrap.bat** file in the ican50\LogicalHost1\bootstrap\bin directory.

6   Copy the sample input data file to the input directory.

For more information on running a Project that utilizes eInsight from the SeeBeyond Enterprise Designer see the *eInsight Business Process Manager User's Guide* and the *eGate Integrator User's Guide*.

4.5.9 Alerting and Logging

eGate provides an alerting and logging feature. This allows monitoring of messages, and captures any adverse messages in order of severity based on configured severity level and higher. To enable Logging, please see the ***eGate Integrator User's Guide***.

Note: *The alerts/status notifications for the WebSphere MQ eWay version 50 are currently limited to Started, Running, Stopping, and Stopped.*

4.6   # The MQ_BP_Put_ Sample Project

The **MQ_BP_Put_ Sample** Project demonstrates the following:

1   The inbound File eWay subscribes to an external directory. The received message is the name of an MQ Queue.

2   The message is concatenated with the string "**This is a message for queue**" and the message from the file.

3   The queueName is set to the message from the file.

4   The message is published by the outbound MQSeries eWay to an external directory.

The MQ_BP_Put_Sample is found in the Samples file as **WebSphereMQSeries_BPEL_Put_Sample.zip**. To Import the Project, see **Importing a Sample Project** on page 21**.**

The following pages provide step by step directions for manually creating the **MQ_BP_Put_ Sample** Project components.

4.6.1   ## Creating a Project

The first step is to create and name a new Project in eGate Enterprise Designer.

1   Start the Enterprise Designer.

2   From the Enterprise Explorer's Project Explorer tab, select your Repository on the Project Explorer tree, right-click the Repository and select **New Project**. A new Project appears on the Project Explorer tree.

3   Click twice on **Project1** and rename the Project (for this sample, **MQ_BP_Put_ Sample**).

4.6.2   ## Creating the BusinessProcess_MQPut Business Process

Creating the Business Process Flow

1   From the Enterprise Designer's Project Explorer tree, right-click **MQ_BP_Put_ Sample**, and select **New** > **Business Process** from the shortcut menu. The eInsight Business Process Designer appears and **BusinessProcess1** is added to the Project Explorer tree. Rename the Business Process to **BusinessProcess_MQPut**.

2   Populate the eInsight Business Process Designer's modeling canvas with the following elements from the Project Explorer tree, as displayed in **Figure 17 on page 34**:

   ◆ **receive**, under SeeBeyond > eWays > File > FileClient

   ◆ **mqput**, under SeeBeyond > eWays > MQSerieseWay > MQSeries

Figure 17   eInsight Business Process Designer - Populate the Canvas
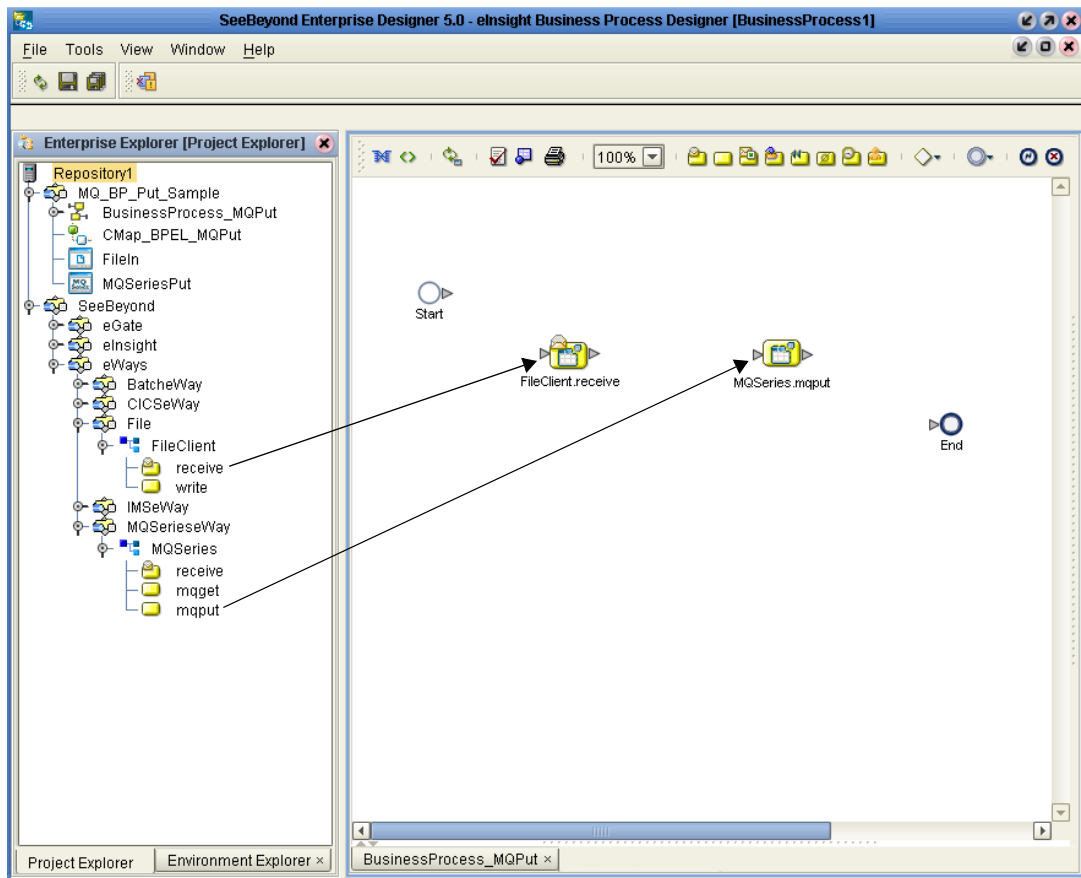
3   Link the modeling elements by clicking on the element's connector and dragging the cursor to the next element's connector, making the following links as displayed in Figure 18.

- ◆ Start -> FileClient.receive

- ◆ FileClient.receive -> MQSeries.mqput

- ◆ MQSeries.mqput -> End

Figure 18   eInsight Business Process Designer - Link the Modeling Elements
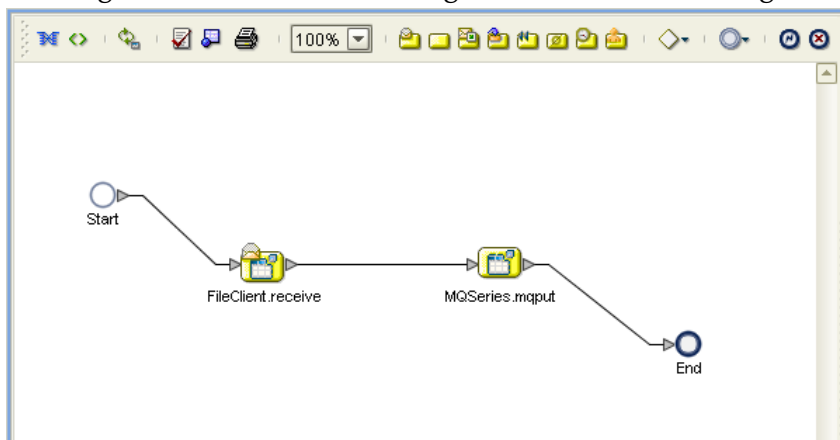
Configuring the Modeling Elements

Business Rules, created between the Business Process elements, define the relationship between the input and output Attributes of the elements.

1   Right-click the link between the **FileClient.receive** and **MQSeries.mqput** Activities and select **Add Business Rule** from the shortcut menu.

2   From the eInsight Business Process Designer toolbar, click the **Map Business Process Attributes** button. The Business Rule Designer appears at the bottom of the eInsight Business Process Designer.

3   Click on the **Business Rules** icon in the link between **FileClient.receive** and **MQSeries.mqput** to display the Business Process Output and Input Attributes in the Business Rule Designer. These Attributes can now be modified.

4   From the Business Rule Designer toolbar, drag and drop the **concat** icon to the Business Rules Designer canvas. The **concat** method box appears.

5   From the Business Rule Designer toolbar, drag and drop the **string literal** icon to the Business Rules Designer canvas. The **string literal** method box appears.

6   Map **text** under FileClient.receive.Output in the Output pane of the Business Rule Designer, to the **string2** input node of the **concat** method box.

7   Map the **'This is a message for queue'** output node of the **string literal** method box, to the **string1** input node of the **concat** method box.

8   Map the **return string** output node of the **concat** method box, to **QPutRequestStringPayLoad** under MQSeries.mqput.Input > input in the Input pane of the Business Rule Designer (see Figure 19).

Figure 19   eInsight Business Rule Designer
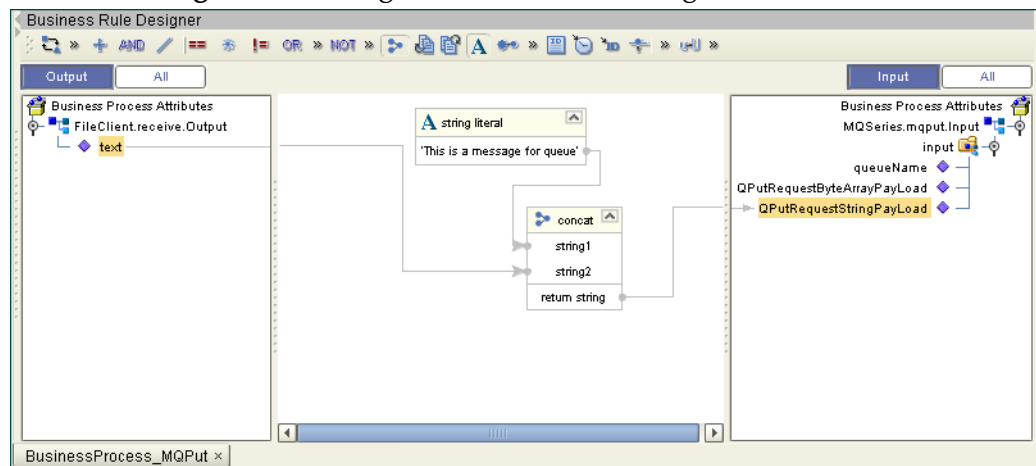
9   Map **text** under FileClient.receive.Output in the Output pane of the Business Rule Designer, to **queueName** under MQSeries.mqput.Input > input in the Input pane of the Business Rule Designer (see Figure 20).

Figure 20 eInsight Business Rule Designer
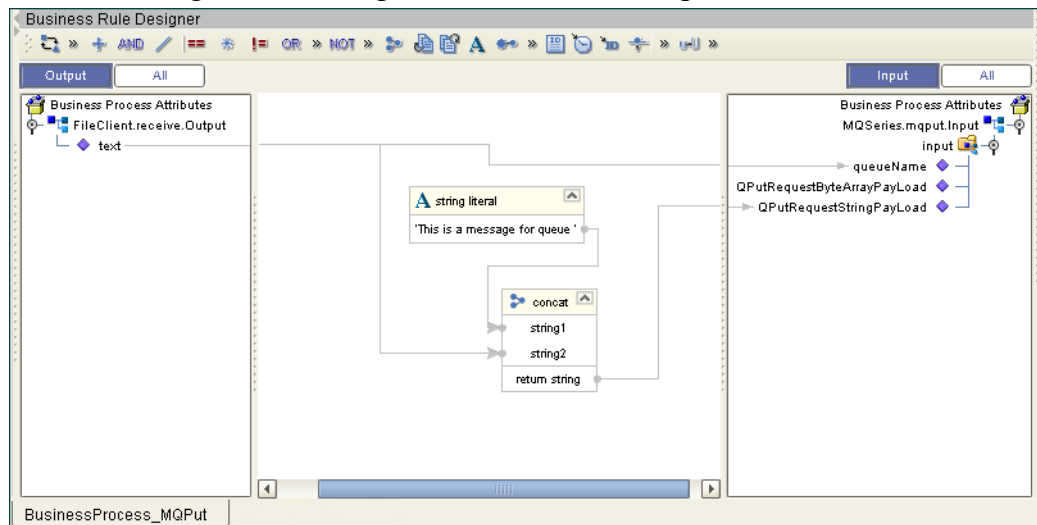
10 From the Business Process Designer toolbar, click the **Synchronize Graphical Model and Business Process Code** icon to synchronize the graphical interface to the Business Process code.

11 Click the Enterprise Designer's **Save All** icon to save your current changes.

4.6.3 Creating a Connectivity Map

The Connectivity Map provides a canvas for configuring a Project's components.

1 In Enterprise Explorer's Project Explorer, right-click the new Project (**MQ_BP_Put_ Sample**) and select **New > Connectivity Map** from the shortcut menu.

2 The New Connectivity Map appears and a node for the Connectivity Map is added under the Project on the Project Explorer tree labeled **CMap1**. Rename the Connectivity Map **CMap_BPEL_MQPut**.

Select the External Applications

1 Click the **External Application** icon on the Connectivity Map toolbar,

2 Select the applications needed for your Project (for this sample, **MQSeries External Application** and **File External Application**). Icons representing the selected applications are added to the Connectivity Map toolbar.

Populate the Connectivity Map

Add the Project components to the Connectivity Map by dragging the icons from the toolbar to the canvas.

1 For the MQ_BP_Put_ Sample Project, drag the following components onto the Connectivity Map canvas as displayed in Figure 21:

   ◆ **Service**

   ◆ **MQSeries External System**

❖ **File External System**

Figure 21   CMap_BPEL_MQPut Connectivity Map with Components
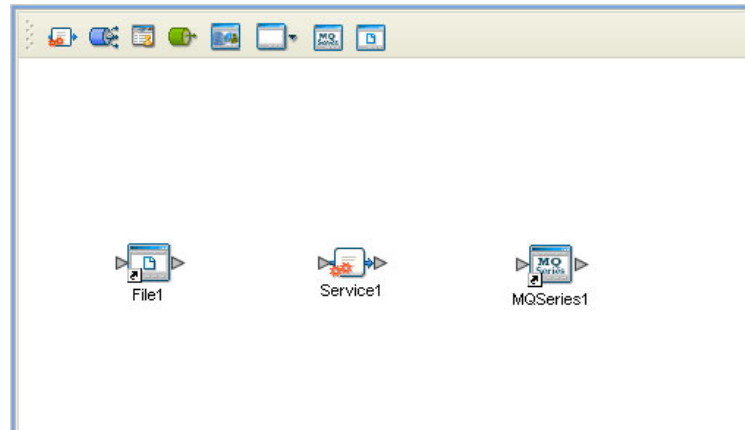
2   Rename the **File1** External Application to **FileIn** by right-clicking the object, selecting **Rename** from the shortcut menu, and typing in the new name.

3   Rename the **Service1** External Application to **BusinessProcess_MQPut1**.

4   Rename **MQSeries1** to **MQSeriesPut**.

5   Save your current changes to the Repository

4.6.4. Binding the Project Components

The components are associated and the bindings are created in the Connectivity Map.

1   From the Project Explorer, double-click the Connectivity Map **CMap_BPEL_MQPut**. The Enterprise Designer canvas now displays the **CMap_BPEL_MQPut** Connectivity Map.

2   Drag and drop the **BusinessProcess_MQPut** Business Process from the Project Explorer onto **BusinessProcess_MQPut1** in the **CMap_BPEL_MQPut** Connectivity Map.

3   Double-click **BusinessProcess_MQPut1**. The **BusinessProcess_MQPut1** binding dialog box appears.

4   From the **BusinessProcess_MQPut1** binding box, map **FileSender** (under Implemented Services) to the **FileIn** application.

5   From the **BusinessProcess_MQPut1** binding box, map the **MQSeriesReceiver** (under Invoked Services) to the **MQSeriesPut** External Application.

6   Minimize the **BusinessProcess_MQPut1** binding box, and save the current changes to your Repository.

4.6.5. # Creating an Environment

Environments include the external systems, Logical Hosts, integration servers and JMS IQ Managers used by a Project and contain the configuration information for these components.

1 From the Enterprise Designer's Enterprise Explorer, click the **Environment Explorer** tab.

2 Right-click the Repository and select **New Environment**. A new Environment is added to the Environment Explorer tree.

3 Rename the new Environment to **MQ_BP_Put_Sample_Env**.

4 Right-click **MQ_BP_Put_Sample_Env** and select **New MQSeries External System**. Name the External System **MQExtSysOut** and select **Outbound MQSeries eWay** as the External System Type. The **MQExtSysOut** window is added to the Environment Editor.

5 Right-click **MQ_BP_Put_Sample_Env** and select **New File External System**. Name this External System **FileExtSysIn** and select **Inbound File eWay** as the External System Type. The **FileExtSysIn** window is added to the Environment Editor.

6 Right-click **MQ_BP_Put_Sample_Env** and select **New Logical Host**. The **LogicalHost1** box is added to the Environment and **LogicalHost1** is added to the Environment Editor tree.

7 From the Environment Explorer tree, right-click **LogicalHost1** and select **New SeeBeyond Integration Server**. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under Localhost2.

8 Save your current changes to the Repository.

4.6.6. # Configuring the eWay Properties

The MQ_BP_Put_ Sample Project uses two eWays, each represented in the Connectivity Map as a node between an External Application and a Service.

The File eWay properties are only configured from the Connectivity Map. The MQSeries eWay, is configured from the Project Explorer's Connectivity Map, and the Environment Explorer. To configure the eWays do the following:

Configuring the File eWay Properties

1 Double-click the **inbound File eWay**, select **inbound File eWay** in the Templates dialog box, and click **OK**.

2 The Properties Sheet opens to the Inbound File eWay properties. Modify the properties for your system.

Configuring the MQSeries eWay Properties

The WebSphere MQ (MQSeries) eWay properties must be set in both the Project Explorer and Environment Explorer. For more information on the MQSeries eWay

properties and the Properties Sheet, see **Creating and Configuring the MQSeries eWay** on page 12 or see the *eGate Integrator User's Guide*.

Modifying the MQSeries eWay Connectivity Map Properties

1 From the Connectivity Map, double-click the **MQSeries eWay**. The Properties Sheet opens to the MQSeries eWay Connectivity Map properties.

2 Modify the MQSeries eWay properties for your system and click **OK**.

Modifying the MQSeries eWay Environment Explorer Properties

1 From the Environment Explorer tree, right-click the Inbound MQSeries External System (**MQSeriesExtSysOut** in this sample), and select **Properties**. The Properties Sheet opens to the MQSeries eWay environment properties.

2 Modify the MQSeries eWay environment properties for your system and click **OK**.

4.6.7 Creating and Activating the Deployment Profile

A Deployment Profile is used to assign Business Processes and message destinations to the integration server and JMS IQ Manager. Deployment profiles are created using the Deployment Editor.

1 From the Enterprise Explorer's Project Explorer, right-click the Project (**MQ_BP_Put_ Sample**) and select **New** > **Deployment Profile**.

2 Enter a name for the Deployment Profile (for this sample **MQ_BP_Put_Sample_DP**). Make sure that the selected Environment is **MQ_BP_Put_Sample_Env**. Click **OK**.

3 From the left pane of the Deployment Editor, drag **FileIn -> BusinessProcess_MQPut1** (External Application) to the **FileExtSysIn** window.

4 From the left pane of the Deployment Editor, drag the **BusinessProcess_MQPut1 -> MQSeriesPut** (External Application) to the **MQExtSysOut** window.

5 From the left pane of the Deployment Editor, drag **BusinessProcess_MQPut1** (Business Process) to **IntegrationSvr1** in the **LogicalHost1** window.

6 Click **Activate**. When activation succeeds, save the changes to the Repository.

4.6.8. Running the Project

The following directions assume that the Enterprise Designer was downloaded to C:\ican50. If this is not the case, replace that location in the following directions with the appropriate location.

1 From the Enterprise Manager Downloads tab, click on **Logical Host - for win32**.

2 Extract the file to the **ican50\LogicalHost2** directory. You must specify the **LogicalHost2** directory for it to be created.

3 Navigate to **C:\ican50\LogicalHost2\bootstrap\config** directory and open the **logical-host.properties** file using Notepad™.

4 Enter the following information in the appropriate fields:

  ◆ Logical Host rood directory: **ican50\LogicalHost2**

  ◆ Repository URL: **http://localhost:***port number/repository name*

  ◆ Repository user name and password: *Your user name and password*

  ◆ Logical Host Environment name: **MQ_BP_Put_Sample_Env**

  ◆ Logical Host name: **LogicalHost2**

Save your changes to **logical-host.properties** and close the file.

5 Run the **bootstrap.bat** file in the **ican50\LogicalHost2\bootstrap\bin** directory.

6 Copy the sample input data file to the input directory.

Implementing a WebSphere MQ eWay Project

This chapter provides an introduction to the MQSeries eWay components and information about how these components are created and implemented in an eGate project. It is assumed that the reader understands the basics of creating a project using the SeeBeyond Enterprise Designer. For more information on creating an eGate project see the *eGate Tutorial* and the *eGate Integrator User's Guide*.

Chapter Topics

- **MQSeries eWay Components** on page 41
- **MQSeries eWay Sample Projects** on page 42
- **Importing a Sample Project** on page 42

5.1 MQSeries eWay Components

This chapter presents a sample MQSeries eWay project created using the same procedures as the sample end-to-end project provided in the *eGate Tutorial*. The eWay components that are unique to the MQSeries eWay include the following:

MQSeries eWay Properties file

The Properties file for the MQSeries eWay contains the parameters necessary to connect with a specific external system. These parameters are set using the Properties dialog box. For more information about the MQSeries eWay Properties file and the Properties dialog box see **Configuring the eWay Properties** on page 12.

MQSeriesETD OTD

The MQSeriesETD OTD is provided with the eWay and contains methods and attributes used to create the Business Rules that invoke the WebSphere MQ program.

5.2 MQSeries eWay Sample Projects

Two sample projects for the MQSeries eWay using Java Collaborations, are included with the installation. This chapter also walks you through the creation of these sample Projects:

- **The MQ_JCE_Get_Sample Project** on page 43
- **The MQ_JCE_Put_Sample Project** on page 55

5.3 Importing a Sample Project

Sample eWay Projects are included as part of the installation CD-ROM package. To import a sample eWay Project to the Enterprise Designer do the following:

1 The sample files are uploaded with the eWay's documentation .sar file and downloaded from the Enterprise Manager's Documentation tab. Extract the samples from the Enterprise Manager to a local file.

2 From the Enterprise Designer's Project Explorer pane, right-click the Repository and select **Import Project** from the shortcut menu. The **Select File to Import** dialog box appears.

3 Browse to and select the sample file (for this sample, **WebSphereMQSeries_JCE_Get_Sample.zip**) and click **Open**.

4 From the File Destination dialog box, select **Import to a new Project**, enter the name of the Project (for this sample, **MQ_JCE_Get_Sample**), and click **OK**.

5 When the import has successfully completed, click the **Refresh All from Repository** icon from the Enterprise Designer toolbar.

6 Before an imported sample Project can be run you must do the following:

- Install and configure a **WebSphere MQ Queue** on the localhost
- Create an **Environment**
- Configure the eWays properties for your specific system
- Create a **Deployment Profile**

5.4 Create the WebSphere MQ Queue

The first step in creating the sample Project is to install and configure **IBM's WebSphere MQ Server** and **MQ Queue Manager** on the localhost.

It is assumed that the reader is experienced in the use of the WebSphere MQ Queue Manager. For more information on the MQ Queue Manager see the MQ Queue Related

Commands section of the SeeBeyond Message Server Reference Guide. For the sample implementation do the following:

1   Open IBM WebSphere MQ Explorer.

2   Create a new Queue Manager.

3   From the WebSphere MQ Queue Manager create a new queue.

IBM WebSphere MQ Server and Queue Manager Limits and Settings

- When using MQSeries Queue Manager on UNIX, the user must be a member of the mqm group to create and start the MQ Queue Manager.

- It is essential that the WebSphere MQ Administrator regularly monitor the number of messages in the queue. Message expiration settings should be set to allow for extended storage.

- WebSphere MQ is limited in the number of messages that can be sent before a commit is executed, and the number of physical messages that can exist on the queue at any one time. This can result in exception errors when upper limits for these numbers are exceeded. Memory and performance of the specific server may also effect the results.

5.5   The MQ_JCE_Get_Sample Project

The MQSeries eWay **MQ_JCE_Get_Sample** Project demonstrates the following:

1   The MQSeries eWay, in receive mode, retrieves a message from an MQ Queue. The message is published to an outbound File eWay.

2   The File eWay published the message to an external directory.

The following pages provides step by step procedures for building the **MQ_JCE_Get_Sample** sample Projects.

5.5.1. Create a Project

The first step is to create and name a new Project in eGate Enterprise Designer.

1   From the Project Explorer tree, right-click the Repository and select **New Project**. A new Project (**Project1**) appears on the Project Explorer tree.

2   Click twice (not double-click) **Project1** to rename the Project (for this sample, **MQ_JCE_Get Sample**).

5.5.2 Create a Connectivity Map

The Connectivity Map provides a canvas for assembling and configuring a Project's components.

1   In Enterprise Explorer's Project Explorer, right-click the **MQ_JCE_Get_Sample** Project and select **New > Connectivity Map** from the shortcut menu.

2 The New Connectivity Map appears and a node is added under the Project on the Project Explorer tree labeled **CMap1**. From the Project Explorer tree, rename the Connectivity Map to **CMap_JCE_MQGet**.

Select the External Applications

The icons on the Connectivity Map toolbar represent the available components used to populate the Connectivity Map canvas.

eWays in the Connectivity Map are associated with External Systems. For example, to establish a connection to WebSphere MQ (MQSeries), you must first select MQSeries as an External System to use in your Connectivity Map (see Figure 22).

Figure 22   Connectivity Map - External Applications
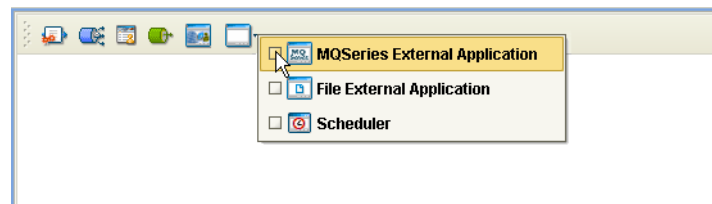
1 Click the **External Application** icon on the Connectivity Map toolbar,

2 Select the external systems needed for your Project (for this sample, the **Scheduler**, and **WebSphere MQ**, and **File** External Applications). Icons representing the selected external systems are added to the Connectivity Map toolbar.

Populate the Connectivity Map

Add the Project components to the Connectivity Map by dragging the icons from the toolbar to the canvas.

1 For this sample, drag and drop the following components onto the canvas as displayed in **Figure 23 on page 45**:

- ◆ **MQSeries External System**

- ◆ **Service** (A service is a container for Java Collaborations, Business Processes, eTL processes, and so forth)

- ◆ **File External System**

Figure 23 Connectivity Map with Components
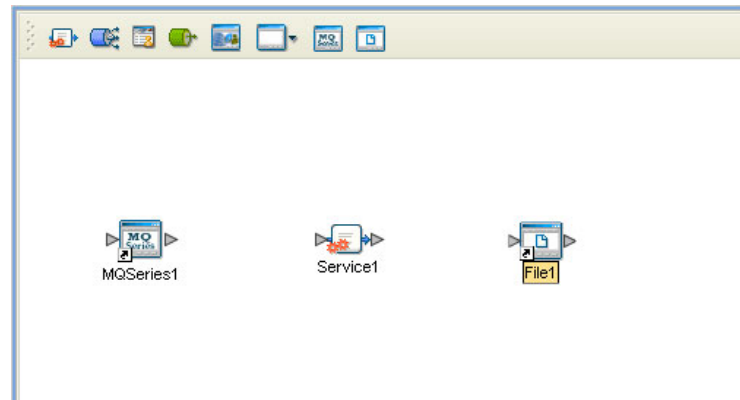
2   Rename the objects by right-clicking the object, selecting Rename from the shortcut menu, and typing in the new name. Change the names as follows:

3   **MQSeries1** to **MQSeriesGet**

4   **Service1** to **Service_JCE_MQGet**

5   **File1** to **FileOut**

6   Click the **Save All** icon to save your current changes to the Repository.

5.5.3. Creating the Java Collaboration Definition

The next step in the sample is to create a Java Collaboration using the Java Collaboration Definition Wizard. Once the Collaboration Definition has been created, the Business Rules of the Collaboration are written using the Java Collaboration Editor.

Creating the jcd_MQGet Java Collaboration

The jcd_MQGet Java Collaboration defines transactions from the WebSphere MQ External Application to the outbound File External Application.

1   From the Project Explorer, right-click the **MQ_JCE_Get_Sample** Project and select **New** > **Java Collaboration Definition** from the shortcut menu. The **Java Collaboration Definition Wizard** appears.

2   Enter a Collaboration name (for this sample **jcd_MQGet**) and click **Next**.

3   For Step 2 of the wizard, from the Web Services Interfaces selection window, double-click **SeeBeyond > eWays > MQSerieseWay > MQSeries > receive**. The File Name field now displays **receive**. Click **Next**.

4   For Step 3 of the wizard, double-click **SeeBeyond** > **eWays** > **File** > **FileClient**. The **FileClient_1** OTD is added to the Selected OTDs field (see **Figure 24 on page 46**).

Figure 24 Java Collaboration Definition Wizard - Select Web Service Interface
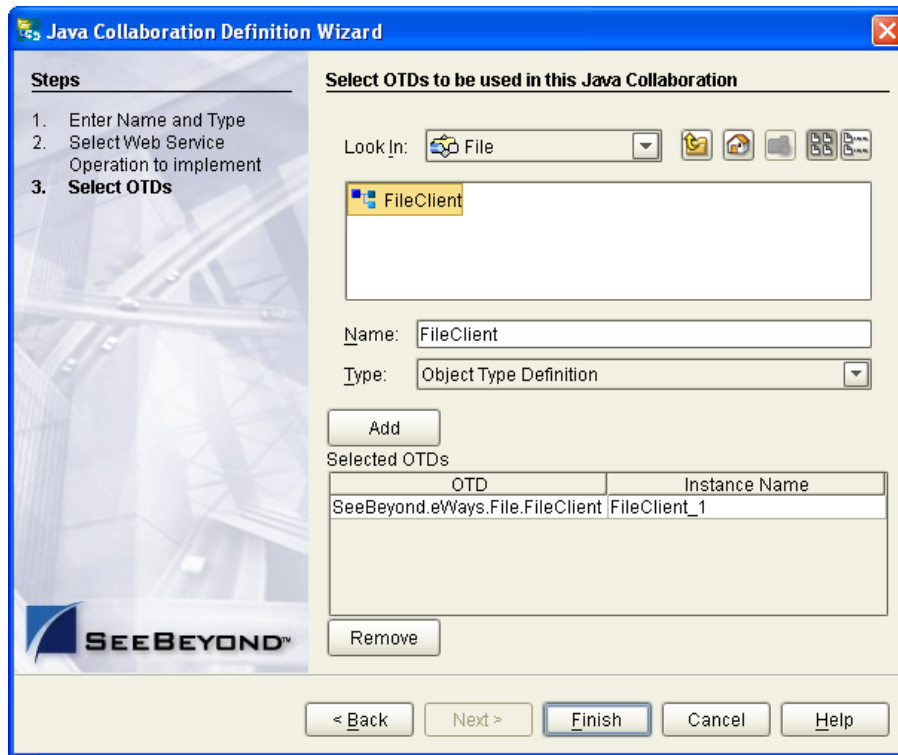
5 Click **Finish**. The Java Collaboration Editor with the new **jcd_MQGet** Collaboration appears in the right pane of the Enterprise Designer.

5.5.4. Using the Java Collaboration Editor

To open the Java Collaboration Editor, from the Project Explorer tree, double-click a Collaboration (for this sample, **jcd_MQGet**). The Java Collaboration Editor opens, in the right pane of the Enterprise Designer, to the selected Collaboration.

Creating the jcd_MQGet Collaboration Business Rule

Be careful to open all nodes specified in the directions to connect to the correct item. The jcd_MQGet Collaboration contains the Business Rule displayed in Figure 25.

Figure 25 jcd_MQGet Collaboration Business Rules
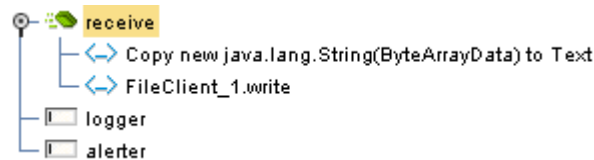
To create the jcd_MQGet Collaboration Business Rules do the following:

1 Double-click the **input**, **MQSeries_1**, and **FileClient_1** in the Business Rule Designer panes to expand the nodes.

2 To create the **Copy new java.lang.String(ByteArrayData) to Text** Business Rule do the following:

A From the Business Rule Designer toolbar, click the **Call New Constructor** icon. The **Call New Constructor** dialog box appears. Select **String** from the **All Classes** box, and **java.lang.String(byte[] bytes)** from the **Constructors** box. The **Package** value defaults to **java.lang**. Click **OK**. The **java.lang.String** method box appears in the Business Rule Designer canvas.

B Map **ByteArrayData** under **input > MsgBody** in the left pane of the Business Rule Designer, to the **bytes (byte[])** input node of the **java.lang.String** method box. This is done by clicking on **ByteArrayData** and dragging the cursor to the **bytes (byte[])** input node.

C Map the result (String) of the **java.lang.String** method box, to Text under **FileClient_1** in the right pane of the Business Rule Designer. (see Figure 26).

Figure 26 Copy new java.lang.String(ByteArrayData) to Text Business Rule
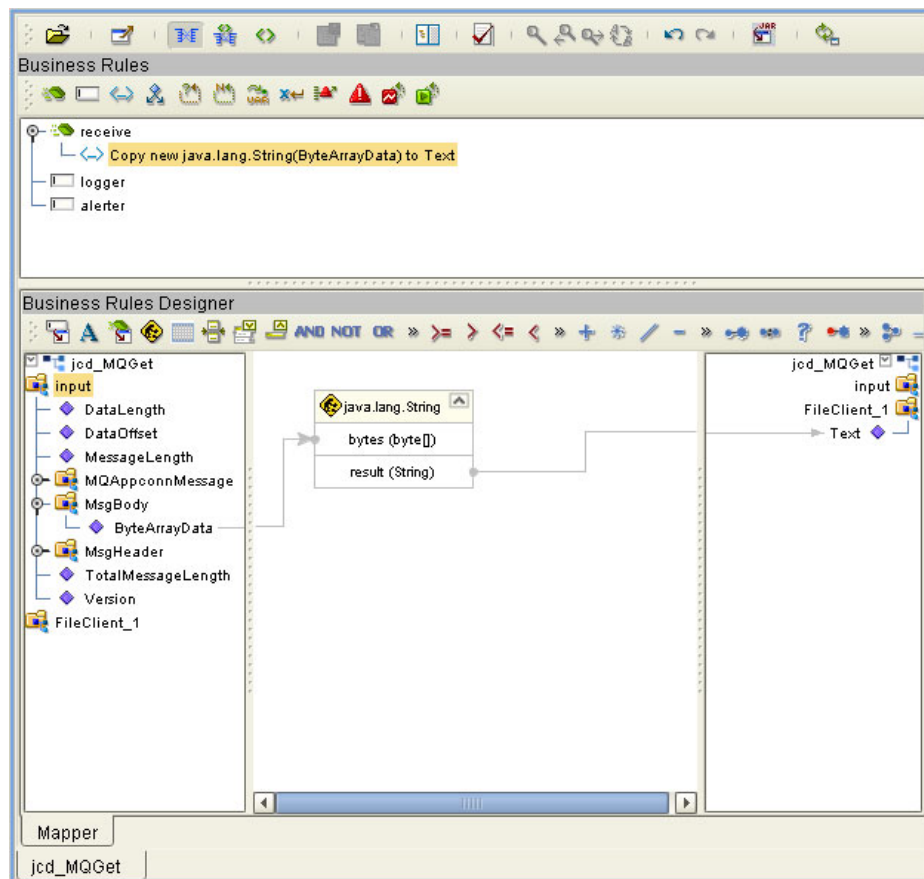
3 To create the **FileClient_1.write** Business Rule do the following:

A From the Business Rules toolbar click the **rule** button to add a new rule.

B Right-click the **FileClient_1** node in the left pane of the Business Rule Designer, and click **Select a method to call**. The method selection menu appears.

C   Select **write()** from the method selection menu. The **write** method box appears with an arrow connecting **FileClient_1**, in the left pane of the Business Rule Designer, to the **FileApplication** input node (seeFigure 27).

Figure 27   FileClient_1.write Business Rule
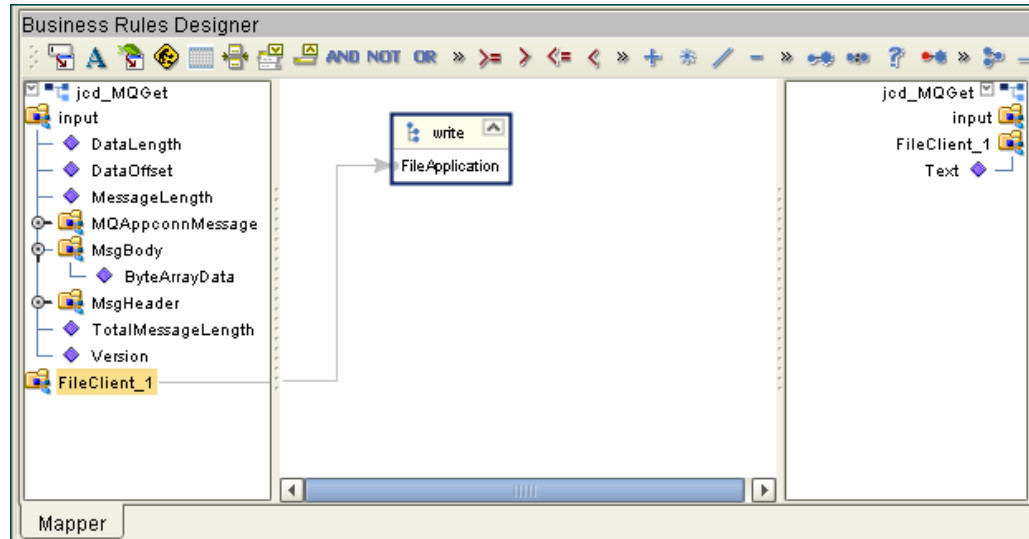
4   From the Enterprise Designer toolbar, click the **Save All** icon to save your current changes to the Repository.

For more information on how to create Business Rules using the Java Collaboration Editor see the *eGate Integrator User's Guide*.

5.5.5. Binding the eWay Components

After the Collaborations have been written, the components are associated and Bindings are created in the Connectivity Map.

1   From the Project Explorer, double-click **CMap_JCE_MQGet**. The Enterprise Designer canvas now displays the Connectivity Map.

2   Drag and drop the **jcd_MQGet** Collaboration from the Project Explorer tree onto the **Service_JCE_MQGet** service. If the Collaboration was successfully associated, the Service's "gears" icon changes from red to green (see **Figure 28 on page 49**).

Figure 28 Connectivity Map - Binding the Collaborations
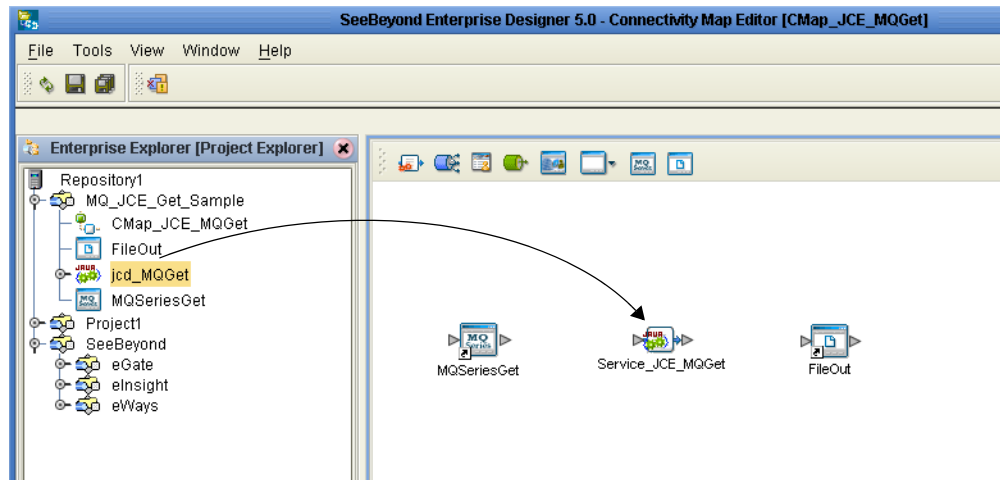
3 Double-click **Service_JCE_MQGet**. The **Service_JCE_MQGet** Binding dialog box appears using **jcd_MQGet** as the rule.

4 From the **Service_JCE_MQGet** Binding dialog box, drag **MQSeries input** (Implemented Services) to the **MQSeriesGet** External Application.

5 From the **Service_JCE_MQGet** Binding dialog box, drag **FileClient_1 FileClient** (under Invoked Services) to the **FileOut** External Application (see Figure 29).

Figure 29 Connectivity Map - Binding the Components
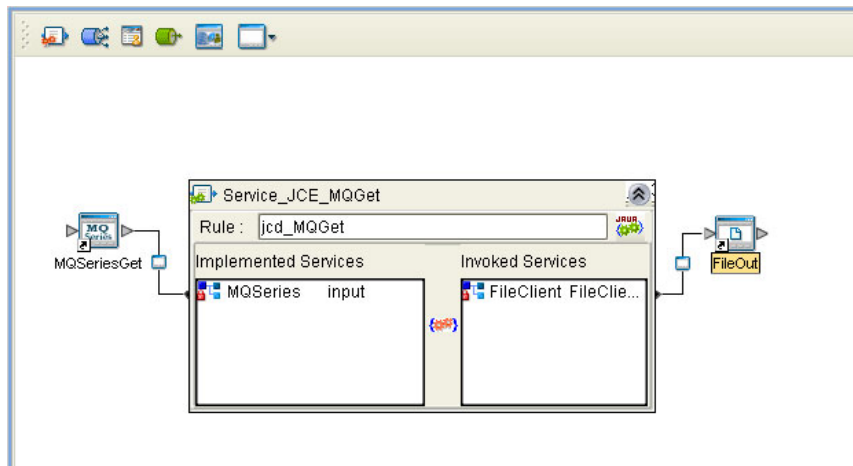
6 Click the chevrons in the upper-right corner of the **Service_JCE_MQGet** Binding dialog box to minimize the **Service_JCE_MQGet**.

5.5.6. Creating an Environment

Environments include the external systems, Logical Hosts, integration servers and message servers used by a Project, and contain the configuration information for these

components. Environments are created using the Enterprise Designer's Environment
Explorer and Environment Editor.

1   From the Enterprise Designer's Enterprise Explorer, click the **Environment
    Explorer** tab.

2   Right-click the Repository and select **New Environment**. A new Environment is
    added to the Environment Explorer tree.

3   Rename the new Environment to **MQ_JCE_Get Sample_Env**.

4   Right-click **MQ_JCE_Get Sample_Env** and select **New MQSeries External System**.
    Name the External System **MQExtSysIn** and select **Inbound MQSeries eWay** as
    the type. Click **OK**.

5   Right-click **MQ_JCE_Get Sample_Env** and select **New File External System**. From
    the Create an External System dialog box, name the External System
    FileExtSysOut, and select **Outbound File eWay** as the type. Click **OK**.

6   Right-click **MQ_JCE_Get SampleEnv** and select **New Logical Host**. The
    LogicalHost1 box is added to the Environment and **LogicalHost1** is added to the
    Environment Editor tree.

7   From the Environment Explorer tree, right-click the Logical Host environment**,
    LogicalHost1**, and select **New SeeBeyond Integration Server**. A new Integration
    Server, **IntegrationSvr1**, is added to the **LogicalHost1** box (see Figure 30).

Figure 30   Environment Editor
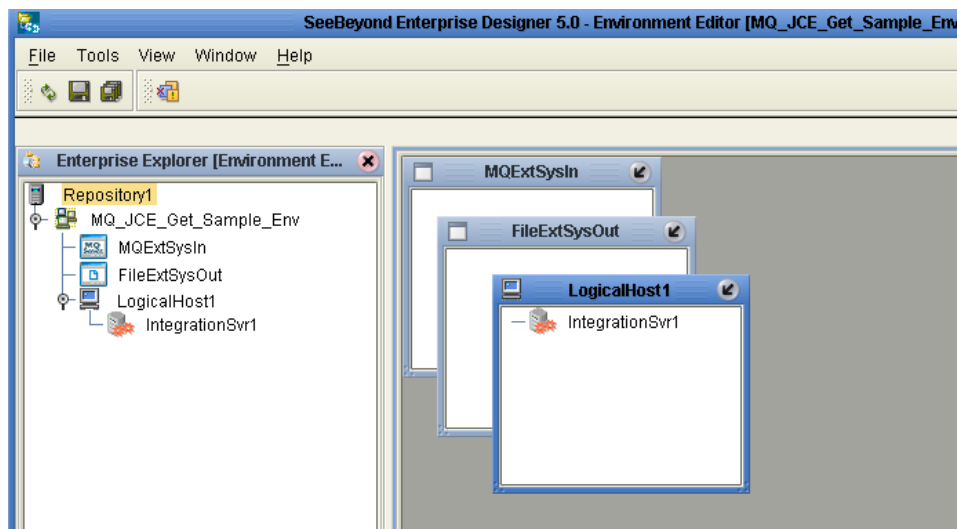
8   Save your current changes to the Repository.

5.5.7. Configuring the eWays

The MQ_JCE_Sample Project uses two component eWays, each represented in the
Connectivity Map as a node between its associated External Application and a Service
(see **Figure 31 on page 51**). eWays facilitate communication and movement of data
between the External Applications and the eGate system.

Figure 31 eWays

The File eWay properties and the Scheduler parameters are configured from the Connectivity Map. The MQSeries eWay properties are set in both the Project Explorer's Connectivity Map and the Environment Explorer. To configure the eWay properties do the following:

Configuring the File eWay Properties

1 from the Connectivity Map, double-click the **File** eWay. The Templates dialog box appears. Select **Outbound File eWay** and click **OK**. The Properties Sheet opens to the Outbound File eWay configuration.

2 Modify the parameters for your system, including the settings for the Outbound File eWay in Table 3, and click **OK**.

Table 3 Outbound File eWay Settings

Outbound File eWay Properties	
Directory	C:/eGate50/data/out
Multiple records per file	True
Output file name	mq_out.txt

Configuring the WebSphere MQ (MQSeries) eWay Properties

The MQSeries eWay parameters must be set in both the Project Explorer and Environment Explorer. For more information on the MQSeries eWay properties and the Properties Sheet, see **Configuring the eWay Properties** on page 12, or see the *eGate Integrator User's Guide*.

Modifying the MQSeries eWay Connectivity Map Properties

1 From the Connectivity Map, double-click the **MQSeriesGet** eWay. The Templates dialog box appears. Select **Inbound MQSeries eWay** and click **OK**. The Properties Sheet opens to the Inbound MQSeries eWay properties.

2 Modify the **MQSeriesGet** eWay Connectivity Map properties for your system, including the settings in **Table 4 on page 51**, and click **OK**.

Table 4 MQSeries eWay Connectivity Map Properties

Inbound MQSeries eWay Connectivity Map Properties	
Settings Set as directed, otherwise use the default settings	
Queue Name	MQ Queue Name

Inbound MQSeries eWay Connectivity Map Properties	
Schedule Interval	Polling Interval in milliseconds

Modifying the MQSeries eWay Environment Explorer Properties

1 From the **Environment Explorer** tree, right-click the MQSeries External System (**MQExtSysIn** in this sample), and select **Properties**. The Properties dialog box opens to the MQSeries eWay Environment properties.

2 Modify the MQSeries eWay Environment properties for your system, including the settings in **Table 5 on page 52**, and click **OK**.

Table 5 MQSeries eWay Environment Explorer Properties

MQSeries eWay Environment Explorer Properties	
Settings Set as directed, otherwise use the default settings	
MQ Host Name	*The name of the host on which the Queue Manager resides*
MQ Queue Manager Name	*Name of the WebSphere MQ Queue Manager*
Password	*Queue Manager Password*
Port Number	*Port number on which the Queue Manager is listening*
UserID	*User ID to access the Queue Manager*

3 Save your changes to the Repository when finished.

5.5.8 Creating and Activating the Deployment Profile

A Deployment Profile is used to assign Collaborations and message destinations to the integration server and message server. Deployment profiles are created using the Deployment Editor.

1 From the Enterprise Explorer's Project Explorer, right-click the Project (**MQ_JCE_Get Sample**) and select **New** > **Deployment Profile**.

2 Enter a name for the Deployment Profile (for this sample **MQ_JCE_Get Sample_DP**). Make sure that the selected Environment is **MQ_JCE_Get Sample_Env**. Click **OK**.

3 Drag **MQSeriesGet -> Service_JCE_MQGet** (External Application) to the **MQExtSysIn** box.

4 From the left pane of the Deployment Editor, drag the **Service_JCE_MQGet -> FileOut** (External Application) to the **FileExtSysOut** box.

5 Drag the **Service_JCE_MQGet** service to **IntegrationSvr1** in the Logical Host (**LogicalHost1**) environment box (see **Figure 32 on page 53**).

Figure 32 Deployment Profile
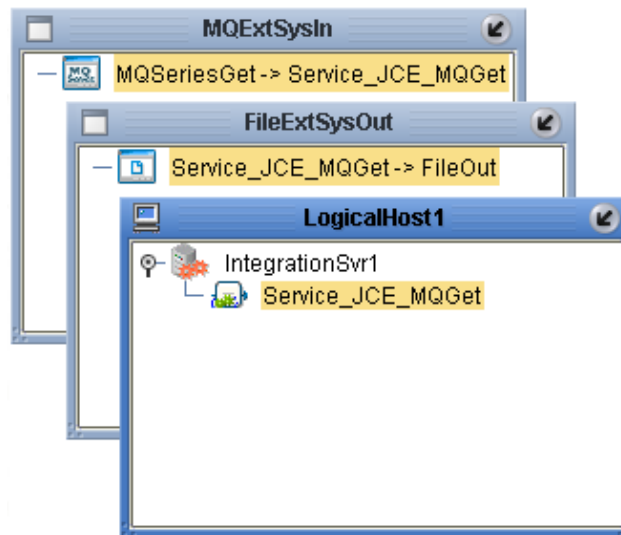
6 Click **Activate**. When activation succeeds, save the changes to the Repository.

5.5.9. Running the Project

The following directions assume that the Enterprise Designer was downloaded to C:\ican50. If this is not the case, replace that location in the following directions with the appropriate location.

1 From the Enterprise Manager Downloads tab, download **Logical Host - for win32**.

2 Extract the file to the **ican50\LogicalHost3** directory. You must specify the **LogicalHost3** directory for it to be created.

3 Navigate to **C:\ican50\LogicalHost3\bootstrap\config** directory and open the **logical-host.properties** file using Notepad™.

4 Enter the following information in the appropriate fields:

   ◆ Logical Host root directory: **ican50\LogicalHost3**

   ◆ Repository URL: **http://localhost:***port number*/*repository name*

   ◆ Repository user name and password: *Your user name and password*

   ◆ Logical Host Environment name: **MQ_JCE_Get_Sample_Env**

   ◆ Logical Host name: **LogicalHost3**

Save your changes to **logical-host.properties** and close the file.

5 Run the **bootstrap.bat** file in the ican50\LogicalHost3\bootstrap\bin directory.

6 Copy the sample input data file to the input directory.

For more information on running a Project that utilizes eInsight from the SeeBeyond Enterprise Designer see the *eInsight Business Process Manager User's Guide* and the *eGate Integrator User's Guide*.

5.5.10 Alerting and Logging

eGate provides an alerting and logging feature. This allows monitoring of messages, and captures any adverse messages in order of severity based on configured severity level and higher. To enable Logging, please see the *eGate Integrator User's Guide*.

Note: *The alerts/status notifications for the WebSphere MQ eWay are currently limited to Started, Running, Stopping, and Stopped.*

5.6   The MQ_JCE_Put_Sample Project

The **MQ_JCE_Put_Sample** Project demonstrates the following:

1 The inbound File eWay subscribes to an external directory. The received message is the name of an MQ Queue.

2 The message is concatenated with the string "This is a message for queue" and the message from the file.

3 The queueName is set to the message from the file.

4 The message is published by the outbound MQSeries eWay to an external directory.

The MQ_BP_Put_Sample is found in the Samples file as **WebSphereMQSeries_JCE_Put_Sample.zip.** To Import the Project, see **Importing a Sample Project** on page 42**.**

The following pages provide step by step directions for manually creating the **MQ_JCE_Put_Sample** Project components.

5.6.1   Create a Project

The first step is to create and name a new Project in eGate Enterprise Designer.

1 Start the Enterprise Designer.

2 From the Enterprise Explorer's Project Explorer tab, select your Repository on the Project Explorer tree, right-click the Repository and select **New Project**. A new Project appears on the Project Explorer tree.

3 Click twice on **Project1** and rename the Project (for this sample, **MQ_JCE_Put_Sample**).

5.6.2   Create a Connectivity Map

The Connectivity Map provides a canvas for configuring a Project's components.

1 In Enterprise Explorer's Project Explorer, right-click the new Project (**MQ_JCE_Put_Sample**) and select **New > Connectivity Map** from the shortcut menu.

2 The New Connectivity Map appears and a node for the Connectivity Map is added under the Project on the Project Explorer tree labeled **CMap1**. Rename the Connectivity Map **CMap_JCE_MQPut**.

Select the External Applications

1 Click the **External Application** icon on the Connectivity Map toolbar,

2 Select the applications needed for your Project (for this sample, **MQSeries External Application** and **File External Application**). Icons representing the selected applications are added to the Connectivity Map toolbar.

Populate the Connectivity Map

Add the Project components to the Connectivity Map by dragging the icons from the toolbar to the canvas.

1 For the MQ_JCE_Put_Sample Project, drag the following components onto the Connectivity Map canvas as displayed in Figure 33:

- ◆ **Service**
- ◆ **MQSeries External System**
- ◆ **File External System**

Figure 33   CMap_JCE_MQPut Connectivity Map with Components
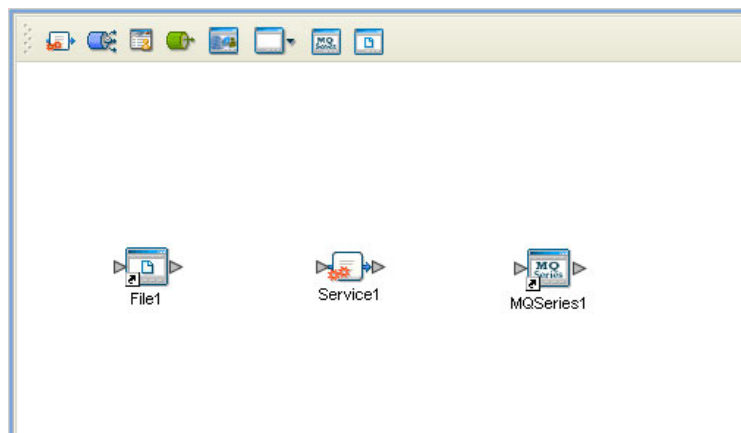
2 Rename the **File1** External Application to **FileIn** by right-clicking the object, selecting **Rename** from the shortcut menu, and typing in the new name.

3 Rename the **Service1** External Application to **Service_JCE_MQPut**.

4 Rename **MQSeries1** to **MQSeriesPut**.

5 Save your current changes to the Repository

5.6.3. Creating the Java Collaboration Definition

The next step in the sample is to create a Java Collaboration using the Java Collaboration Definition Wizard. Once the Collaboration Definition has been created, the Business Rules of the Collaboration are written using the Java Collaboration Editor.

Creating the jcd_MQPut Java Collaboration

The jcd_MQPut Java Collaboration defines transactions from the Inbound File External Application to the WebSphere MQ External Application.

1 From the Project Explorer, right-click the **MQ_JCE_Put_Sample** Project and select **New** > **Java Collaboration Definition** from the shortcut menu. The **Java Collaboration Definition Wizard** appears.

2 Enter a Collaboration name (for this sample **jcd_MQPut**) and click **Next**.

3 For Step 2 of the wizard, from the Web Services Interfaces selection window, double-click **SeeBeyond > eWays > File > FileClient > receive.** The File Name field now displays **receive.** Click **Next**.

4 For Step 3 of the wizard, double-click **SeeBeyond > eWays > MQSerieseWay > MQSeries .** The **MQSeries OTD** is added to the Selected OTDs field .

5 Click **Finish**. The Java Collaboration Editor with the new **jcd_MQPut** Collaboration appears in the right pane of the Enterprise Designer.

5.6.4. Using the Java Collaboration Editor

To open the Java Collaboration Editor, from the Project Explorer tree, double-click a Collaboration (for this sample, **jcd_MQPut**). The Java Collaboration Editor opens, in the right pane of the Enterprise Designer, to the selected Collaboration.

Creating the jcd_MQPut Collaboration Business Rule

Be careful to open all nodes specified in the directions to connect to the correct item. The jcd_MQPut Collaboration contains the Business Rule displayed in Figure 34.

Figure 34   jcd_MQPut Collaboration Business Rules
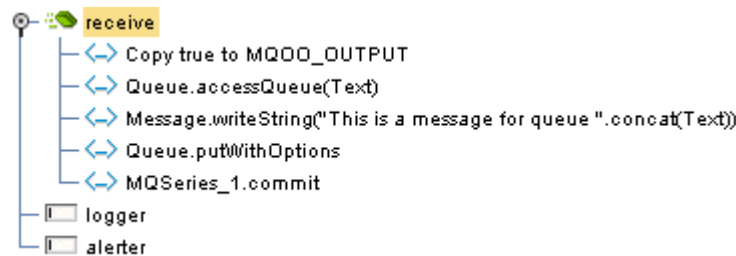
To create the jcd_MQPut Collaboration Business Rules do the following:

1 Double-click the **input**, **MQSeries_1**, and **FileClient_1** in the Business Rule Designer panes to expand the nodes.

2 To create the **Copy true to MQOO_OUTPUT** Business Rule do the following:

A From the Business Rule Designer toolbar, click the **Create Literal** icon. The **Create Literal** dialog box appears. Select **boolean** as the Type, and true as the Value. Click **OK**. The **Literal** method box appears in the Business Rule Designer.

B Map the **true** output node of the **Literal** method box, to **MQOO_OUTPUT** under **MQSeries_1 > queueAccessOptions** in the right pane of the Business Rule Designer. This is done by clicking on **true** output node, and dragging the cursor to **MQOO_OUTPUT** (see **Figure 35 on page 58**).

Figure 35   Copy true to MQOO_OUTPUT Business Rule
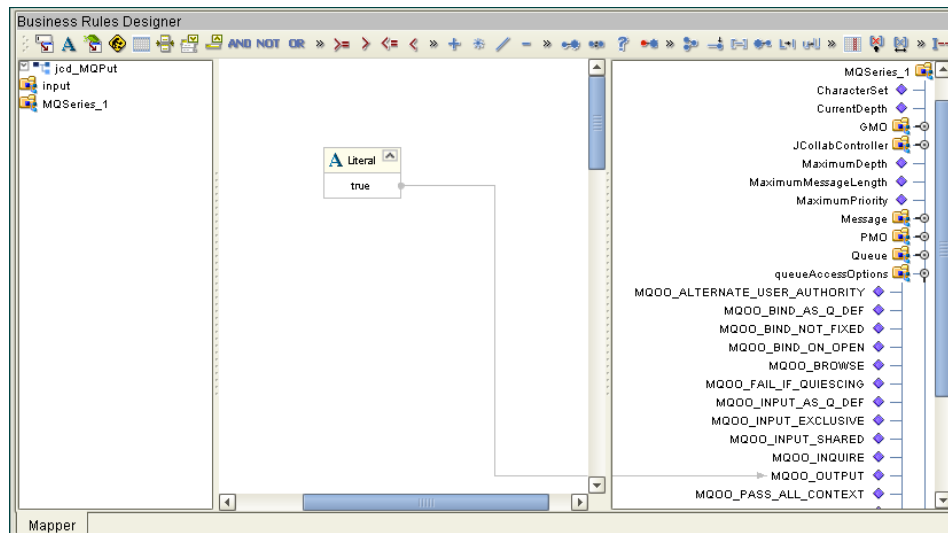
3 To create the **Queue.accessQueue(Text)** Business Rule do the following:

A From the Business Rules toolbar click the **rule** button to add a new rule.

B Right-click **Queue** underMQSeries_1 in the left pane of the Business Rule Designer, and click **Select a method to call**. The method selection menu appears.

C Select **accessQueue(java.lang.String arg0)** from the method selection menu. The **accessQueue** method box.

D Map **Text** under **input** in the left pane of the Business Rule Designer, to the **arg0** input node of the **accessQueue** method box (see Figure 36).

Figure 36   Queue.accessQueue(Text) Business Rule
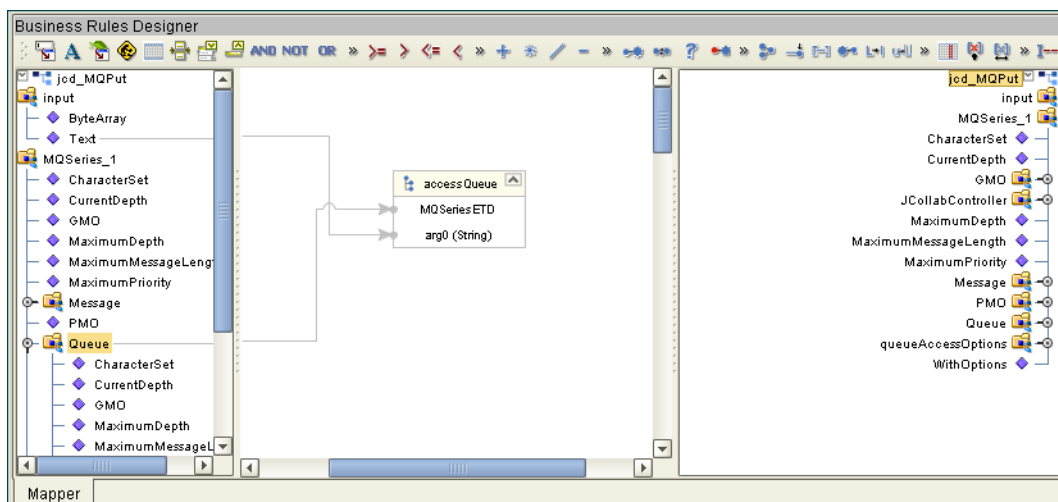
4 To create the **Message.writeString("This is a message for queue ".concat(Text))** Business Rule do the following:

A From the Business Rules toolbar click the **rule** button to add a new rule.

B   From the Business Rule Designer toolbar, drag the **concat** icon to the Business Rules Designer canvas. The **concat** method box appears.

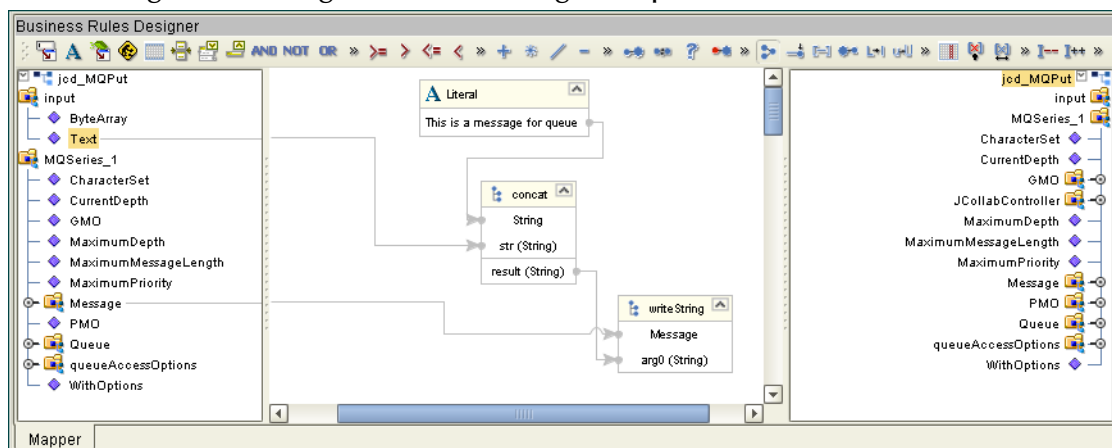C   From the Business Rule Designer toolbar, click the **Create Literal** icon. The Create Literal dialog box appears. Select **String** as the Type, enter **This is a message for queue** as the Value, and click **OK**. The **Literal** method box appears on the Business Rules Designer canvas.

D   Right-click the **Message** node under **MQSeries_1** in the left pane of the Business Rule Designer, and click **Select a method to call**. The method selection menu appears.

E   Select **writeString(java.lang.string arg0)** from the method selection menu. The **writeString** method box appears.

F   Map **Text** under **input** in the left pane of the Business Rule Designer, to the **str (String)** input node of the **concat** method box.

G   Map the **This is a message for queue** output node of the **Literal** method box, to the **String** input node of the **concat** method box.

H   Map the **result (String)** output node of the **concat** method box, to the **arg0 (String)** input node of the **writeString** method box (see Figure 37).

Figure 37   Message.writeString("This is a message for queue ".concat(Text)) Business Rule

5   To create the **Queue.putWithOptions** Business Rule do the following:

A   From the Business Rules toolbar click the **rule** button to add a new rule.

B   Right-click the **Queue** node under **MQSeries_1** in the left pane of the Business Rule Designer, and click **Select a method to call**. The method selection menu appears.

C   Select **putWithOptions()** from the method selection menu. The **putWithOptions** method box appears on the Business Rules Designer canvas (see **Figure 38 on page 60**).

Figure 38   Queue.putWithOptions Business Rule

6   To create the **MQSeries_1.commit** Business Rule do the following:

A   From the Business Rules toolbar click the **rule** button to add a new rule.

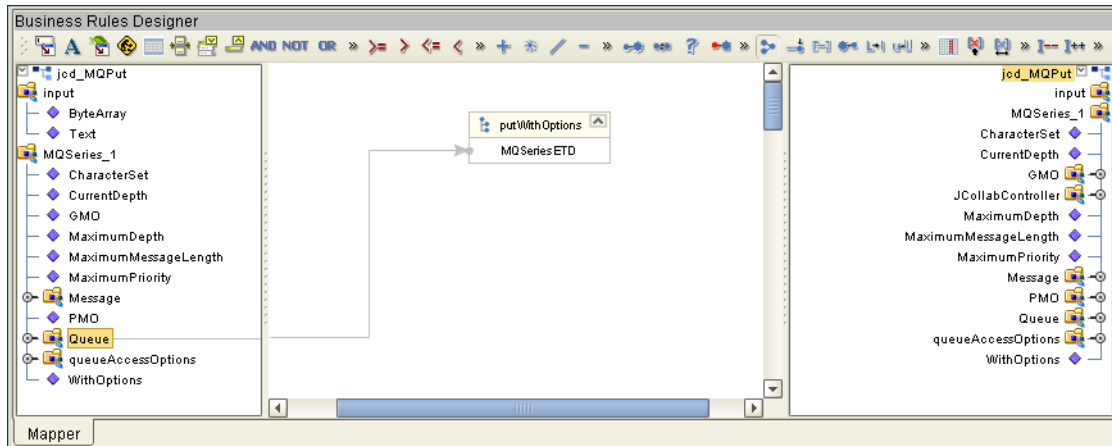B   Right-click **MQSeries_1** in the left pane of the Business Rule Designer, and click **Select a method to call**. The method selection menu appears.

C   Select **commit()** from the method selection menu. The **commit** method box appears on the Business Rules Designer canvas.

7   From the Enterprise Designer toolbar, click the **Save All** icon to save your current changes to the Repository.

8   For more information on how to create Business Rules using the Java Collaboration Editor see the *eGate Integrator User's Guide*.

5.6.5. Binding the Project Components

The eWay components are associated and the bindings are created in the Connectivity Map.

1   From the Project Explorer, double-click **CMap_JCE_MQPut** to open the Connectivity Map.

2   Drag and drop the **jcd_MQPut** Collaboration from the Project Explorer tree to **Service_JCE_MQPut** in the **CMap_JCE_MQPut** Connectivity Map.

3   Double-click **Service_JCE_MQPut**. The **Service_JCE_MQPut** binding dialog box opens.

4   From the **Service_JCE_MQPut** binding box, map **FileClient input** (under Implemented Services) to the **FileIn** application.

5   From the **Service_JCE_MQPut** binding box, map the **MQSeries_1 MQSeries** (under Invoked Services) to the **MQSeriesPut** External Application.

6   Minimize the **Service_JCE_MQPut** binding box, and save the current changes to your Repository.

5.6.6. **Creating an Environment**

Environments include the external systems, Logical Hosts, integration servers and JMS IQ Managers used by a Project and contain the configuration information for these components.

1 From the Enterprise Designer's Enterprise Explorer, click the **Environment Explorer** tab.

2 Right-click the Repository and select **New Environment**. A new Environment is added to the Environment Explorer tree.

3 Rename the new Environment to **MQ_JCE_Put_Sample_Env**.

4 Right-click **MQ_JCE_Put_Sample_Env** and select **New MQSeries External System**. Name the External System **MQExtSysOut** and select **Outbound MQSeries eWay** as the External System Type. The **MQExtSysOut** window is added to the Environment Editor.

5 Right-click **MQ_JCE_Put_Sample_Env** and select **New File External System**. Name this External System **FileExtSysIn** and select **Inbound File eWay** as the External System Type. The **FileExtSysIn** window is added to the Environment Editor.

6 Right-click **MQ_JCE_Put_Sample_Env** and select **New Logical Host**. The **LogicalHost1** box is added to the Environment and **LogicalHost1** is added to the Environment Editor tree.

7 From the Environment Explorer tree, right-click **LogicalHost1** and select **New SeeBeyond Integration Server**. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under **LogicalHost1**.

8 Save your current changes to the Repository.

5.6.7. **Configuring the eWay Properties**

The MQ_JCE_Put_Sample Project uses two eWays, each represented in the Connectivity Map as a node between an External Application and a Service.

The File eWay properties are only configured from the Connectivity Map. The MQSeries eWay, is configured from the Project Explorer's Connectivity Map, and the Environment Explorer. To configure the eWays do the following:

Configuring the File eWay Properties

1 Double-click the **inbound File eWay**, select **inbound File eWay** in the Templates dialog box, and click **OK**.

2 The Properties Sheet opens to the Inbound File eWay properties. Modify the properties for your system.

Configuring the MQSeries eWay Properties

The WebSphere MQ (MQSeries) eWay properties must be set in both the Project Explorer and Environment Explorer. For more information on the MQSeries eWay properties and the Properties Sheet, see **Creating and Configuring the MQSeries eWay** on page 12 or see the *eGate Integrator User's Guide*.

Modifying the MQSeries eWay Connectivity Map Properties

1 From the Connectivity Map, double-click the **MQSeries** eWay. The Properties Sheet opens to the MQSeries eWay Connectivity Map properties.

2 Modify the **MQSeries** eWay properties for your system and click **OK**.

Modifying the MQSeries eWay Environment Explorer Properties

1 From the **Environment Explorer** tree, right-click the Inbound MQSeries External System (**MQSeriesExtSysOut** in this sample), and select **Properties**. The Properties Sheet opens to the MQSeries eWay environment properties.

2 Modify the **MQSeries** eWay environment properties for your system and click **OK**.

5.6.8 Creating and Activating the Deployment Profile

A Deployment Profile is used to assign Collaborations and message destinations to the integration server and JMS IQ Manager. Deployment profiles are created using the Deployment Editor.

1 From the Enterprise Explorer's Project Explorer, right-click the **MQ_JCE_Put_Sample** Project and select **New** > **Deployment Profile**.

2 Enter a name for the Deployment Profile (for this sample **MQ_JCE_Put_Sample_DP**). Make sure that the selected Environment is **MQ_JCE_Put_Sample_Env**. Click **OK**.

3 From the left pane of the Deployment Editor, drag **FileIn -> Service_JCE_MQPut** (External Application) to the **FileExtSysIn** window.

4 From the left pane of the Deployment Editor, drag the **Service_JCE_MQPut -> MQSeriesPut** (External Application) to the **MQExtSysOut** window.

5 From the left pane of the Deployment Editor, drag **Service_JCE_MQPut** (Business Process) to **IntegrationSvr1** in the **LogicalHost1** window.

6 Click **Activate**. When activation succeeds, save the changes to the Repository.

5.6.9. Running the Project

The following directions assume that the Enterprise Designer was downloaded to C:\ican50. If this is not the case, replace that location in the following directions with the appropriate location.

1 From the Enterprise Manager Downloads tab, click on **Logical Host - for win32**.

2 Extract the file to the **ican50\LogicalHost4** directory. You must specify the **LogicalHost4** directory for it to be created.

3   Navigate to **C:\ican50\LogicalHost4\bootstrap\config** directory and open the
    logical-host.properties file using Notepad™.

4   Enter the following information in the appropriate fields:

    ◆ Logical Host rood directory: **ican50\LogicalHost4**

    ◆ Repository URL: **http://localhost:***port number/repository name*

    ◆ Repository user name and password: *Your user name and password*

    ◆ Logical Host Environment name: **MQ_JCE_Put_Sample_Env**

    ◆ Logical Host name: **LogicalHost4**

    Save your changes to **logical-host.properties** and close the file.

5   Run the **bootstrap.bat** file in the **ican50\LogicalHost4\bootstrap\bin** directory.

6   Copy the sample input data file to the input directory.

Java Classes and Methods for the WebSphere MQ eWay

6.1 MQSeries eWay Classes and Methods

The MQSeries eWay contains Java methods that are used to extend the functionality of the eWay. These methods are contained in the following classes:

- MQSeriesETD Class
- GMO Class
- PMO Class
- Message Class

WebSphere MQ Javadoc

The Javadoc is uploaded with the eWay's documentation file (**MQSerieseWayDocs.sar**) and downloaded from the Documentation tab of the Enterprise Manager. To access the full Javadoc, extract the Javadoc to an easily accessible folder, and double click the **index.html** file.

Mapping WebSphere MQ Header Fields

A.1 Mapping Between JMS Standard Header Items and WebSphere MQ Header Fields

JMS Standard header items and their equivalent WebSphere MQ header fields can be set using the Java Collaboration Rules Editor. For information on mapping between JMS header items and WebSphere MQ header fields see IBM MQSeries online documentation at:

http://www-4.ibm.com/software/ts/mqseries/library/manual01/csqzaw07/ csqzaw07tfrm.htm

Table 20, at the above Web site provides JMS header fields used to set or get MQSeries header fields (only some of which are available using this procedure). The Java Collaboration Rules Editor sets the header properties by calling **readProperty()** or **writeProperty()**.

For detailed information on creating Collaboration Rules using the Java Collaboration Rules Editor see the *eGate Integrator User's Guide*.

Support for WebLogic and WebSphere Application Servers

This section describes how to deploy an Enterprise Archive (EAR) file to the WebLogic™ or WebSphere™ Application Server. This includes information on installing the WebLogic or WebSphere Application Server interface and configuring the selected Application server to deploy the EAR file.

Chapter Topics:

B.1 Uploading the Application Server Interface

To support the WebLogic or WebSphere Application Server, the following files must be uploaded to your system and installed:

WebLogic Application Server

- weblogic.sar
- weblogicjmsmessageserver.sar

WebSphere Application Server

- websphereintegserver.sar
- webspherejmsmessageserver.sar

For information on uploading and installing the selected .sar files see the *SeeBeyond ICAN Suite Installation Guide.*

B.2   **Creating an EAR File**

To create an EAR file from an eGate Project, include the following steps during the creation of the Project:

1   From the JMS Properties Sheet (accessed from the Connectivity Map) set the values as follows:

 ◆ Set the **Transaction Mode** parameter (under JMS Client, Basic) to **XA**

 ◆ Set the **USE for JMS connection** parameter (under JMS Client, Basic, Run-as principal) to **false**

2   During the creation of the Project Environment do the following:

 A   Create the Logical Host in the Environment.

 B   From the Environment Explorer tree, right-click the logical host and select **New WebLogic** (or **New WebSphere**) **Integration Server** from the shortcut menu. The integration server is added to the Logical Host box and the Environment Explorer tree.

3   During the creation of the Deployment Profile do the following

 A   Drag and drop the Service containing the Collaboration to the WebSphere or WebLogic Integration Server in the Logical Host box.

 B   Drag and drop the Topic or Queue to the WebSphere or WebLogic JMS Server in the Logical Host box.

 C   Once all of the components have been placed in the environment, click **Activate**. When prompted whether to apply the Project to the logical host immediately, click **No**.

4   The Project's EAR file is now available in the following location:

```
<ICAN>/repository/data/files/<environment name>/<logical host
name>/<application server name> directory
```

where **<ICAN>** is the directory where the ICAN Suite is installed, **<environment name>** is the name of the Project Environment, **<logical host name>** is the name of the logical host, and **<application server name>** is the name of the selected Application Server.

Note: *For an eGate Project EAR file to deploy to the WebLogic or WebSphere Application Server, the Project must not contain any inbound e*Ways.*

B.3   **Deploying an EAR File**

To deploy an EAR file using the WebLogic or WebSphere Application Server, you first create the following components from the Application Server's Administrative Console:

- **Topics** or **Queues**: One Topic or Queue is needed for each Topic or Queue in the eGate Project.

- **Connection Factories**: A Connection Factory enables JMS Clients to create JMS connections with predefined attributes. If the Project contains Topics, a Topic Connection Factory is required. If it contains Queues a Queue Connection Factory is needed. If the Project contains both Topics and Queues, create both a Queue and a Topic Connection Factory.

WebSphere also requires the following:

- **Listener Ports**: A listener port must be created for every MDB the Project contains. The **MDB** is represented in the Project's Connectivity Map as the connection between a **Topic** or **Queue** and a **Service** (see Figure 39).

Figure 39   Project/EAR File
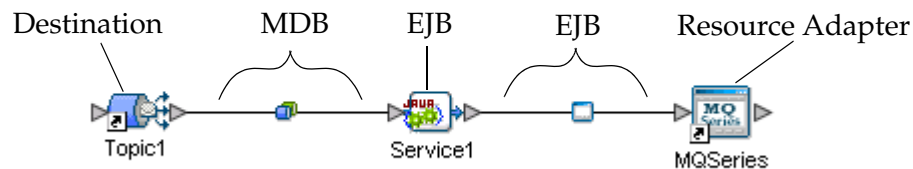
B.4   Configuring the WebLogic Application Server

To deploy an EAR file in the WebLogic Application Server do the following:

1 Start the WebLogic Server by double-clicking the following file:

```
<WL_HOME>/server/bin/startWLS.cmd
```

where <WL_HOME> is the location of WebLogic directory on your system. When Prompted, enter your user name and password.

2 Start the Server Administration Console.

Creating a JMS File Store

1 From the Console tree, click **mydomain**, click **Services**, click **JMS**, and click **Stores.**

2 From the **Create a new JMS File Store** window enter a name for the JMS Store.

3 Create a new directory on your system where the JMS file store will be kept, then enter the absolute path for the directory in the **Directory** field (see Figure 40). Click **Apply**.
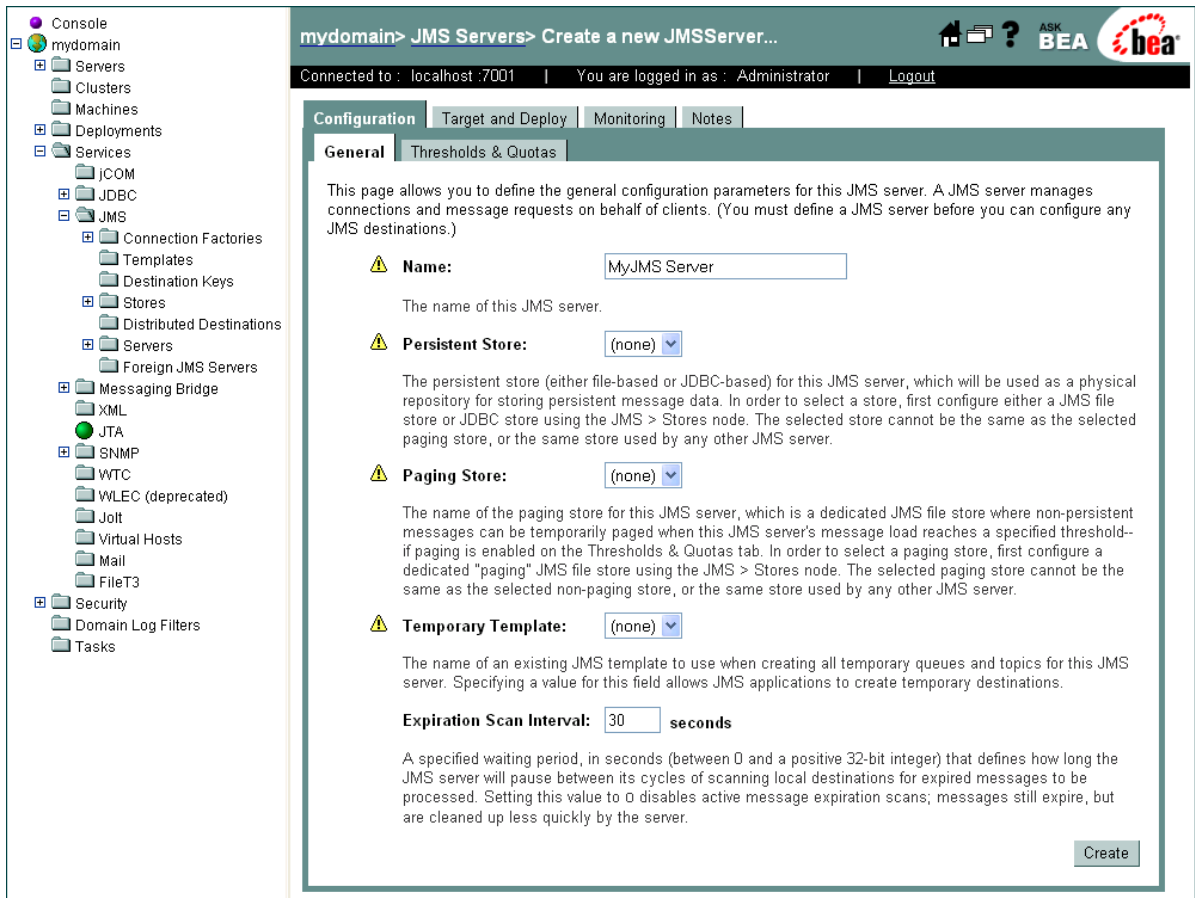
Figure 40   WebLogic Server Console - Create a JMS File Store
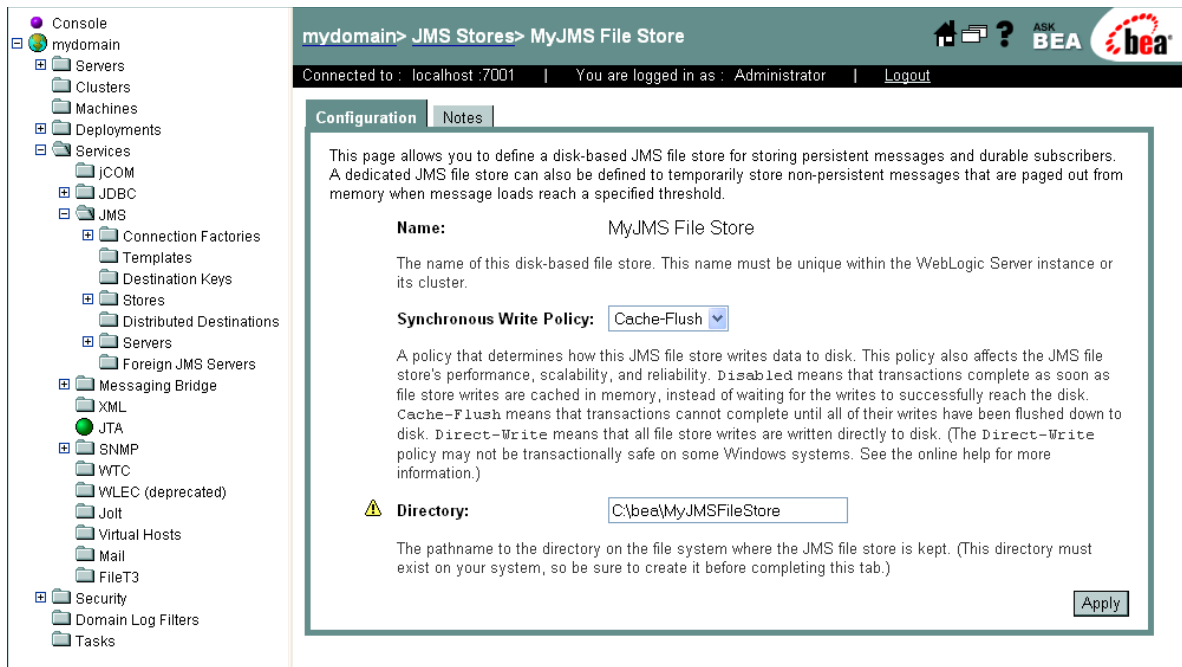
Creating a JMS Server

1 From the Console tree, click **mydomain**, click **Services**, click **JMS**, and click **Servers.**

2 From the **JMS Servers** window, click **Configure a new JMS Server**.

3 From the **MyJMS Server** window, enter a name for the JMS server (in this case, the default **MyJMS Server**) as displayed in Figure 41. Click **Create**.

Figure 41 WebLogic Server Console - Create a JMS Server

Creating a Topic or Queue

1 From the Console tree, click **mydomain**, click **Services**, click **JMS**, click **Servers**, click the new JMS server (in this example, **MyJMS Server**), and click **Destinations**.

2 From the **JSM Destinations** window, click either **Configure a new JMS Queue** or **Configure a new JMS Topic**.

3 From the Topic or Queue configuration window, enter a name for the new Topic or Queue in the Name field. Enter the same name in the JNDI Name field (see Figure 42). Click **Create**.

**Figure 42** WebLogic Server Console - Create a JMS Topic or Queue
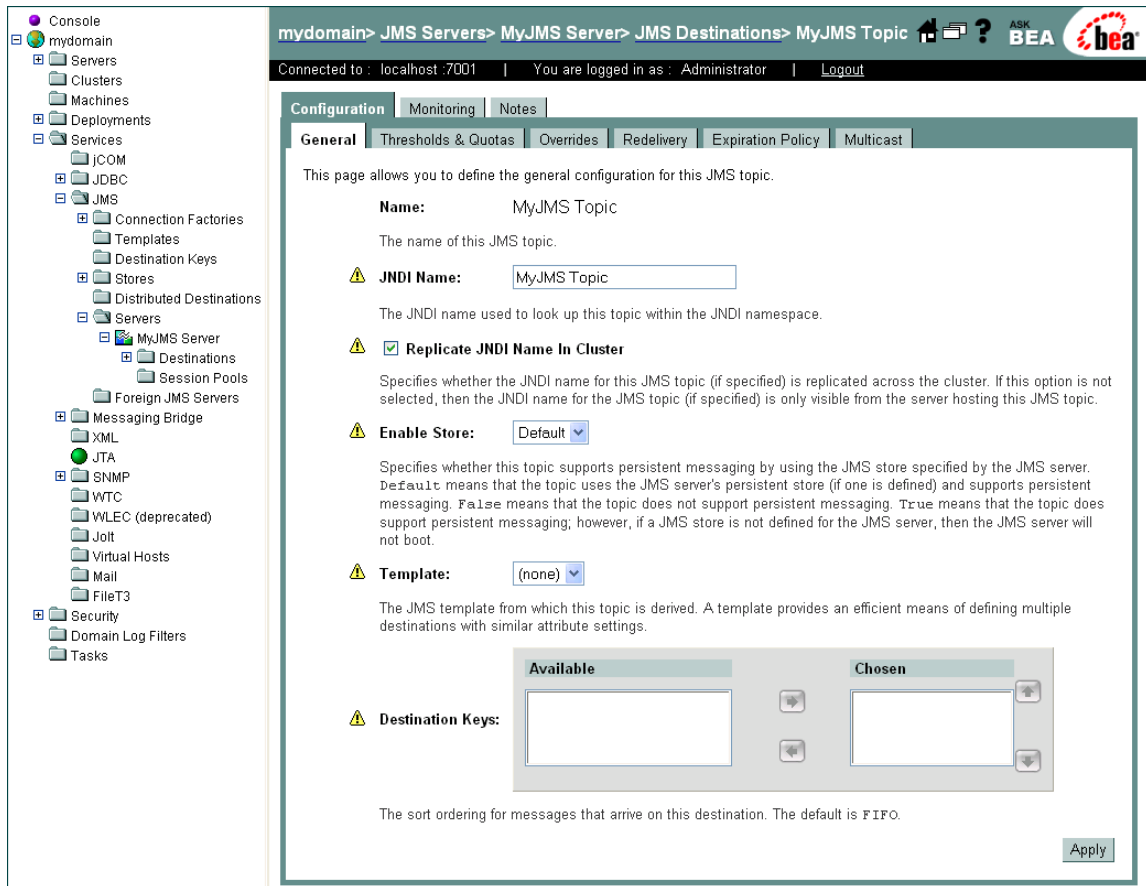
Creating a Connection Factory

1 From the Console tree, click **mydomain**, click **Services**, click **JMS**, click **Connection Factories**.

2 From the **JSM Connection Factories** window, click **Configure a new JMS Connection Factory**.

3 From the **Create a new JMSConnectionFactory** window, enter a name for the new JMS Connection Factory in the **Name** field (for this example, the default **MyJMS Connection Factory**).

4 Enter the Connection Factory JNDI Name in the **JNDI Name** field. The pattern for the JNDI Name is **jms/connectionfactory/**<*xa_topic* or *xa_queue (just topic* or *queue if xa is not selected as the Transaction Mode)>***/**<*Logical Host name>_<JMS Message Server name (from the Environment)>***.**

For this example the Connection Factory JNDI name is **jms/connectionfactory/ xa_topic/LogicalHost1_WLMessageSvr1** (see Figure 43). Click **Create**.

**Figure 43** WebLogic Server Console - Create a Connection Factory
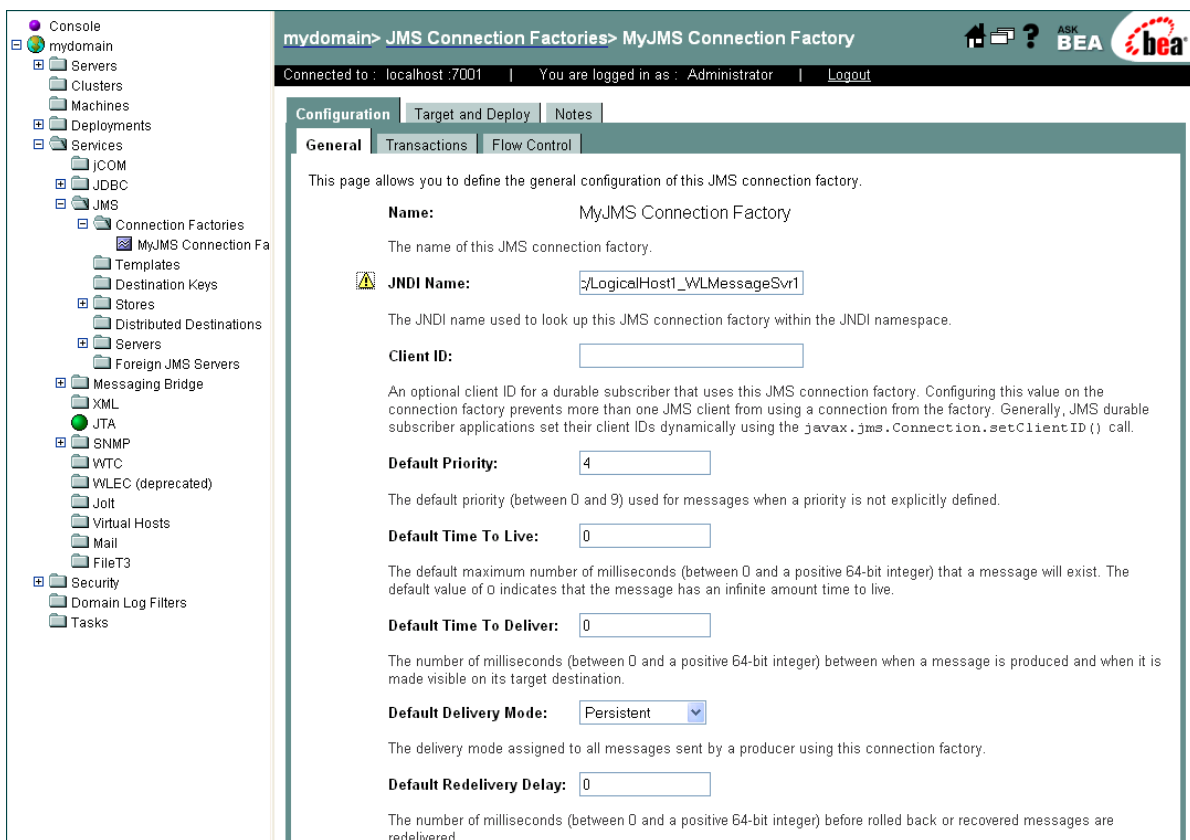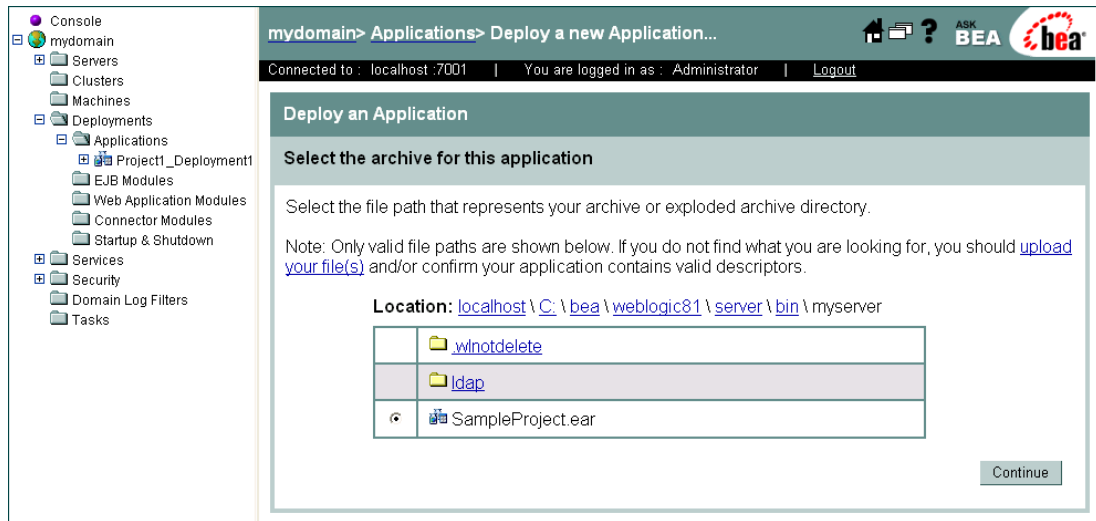
Deploying the Application

1 From the Console tree, click **mydomain**, click **Deployments**, and click **Applications**.

2 From the **Applications** window, click **Deploy a new Application**.

3 From the **Deploy a new Application** window, location of the directory that contains the EAR file (see Figure 44).

**Figure 44**   WebLogic Server Console - Select an Application
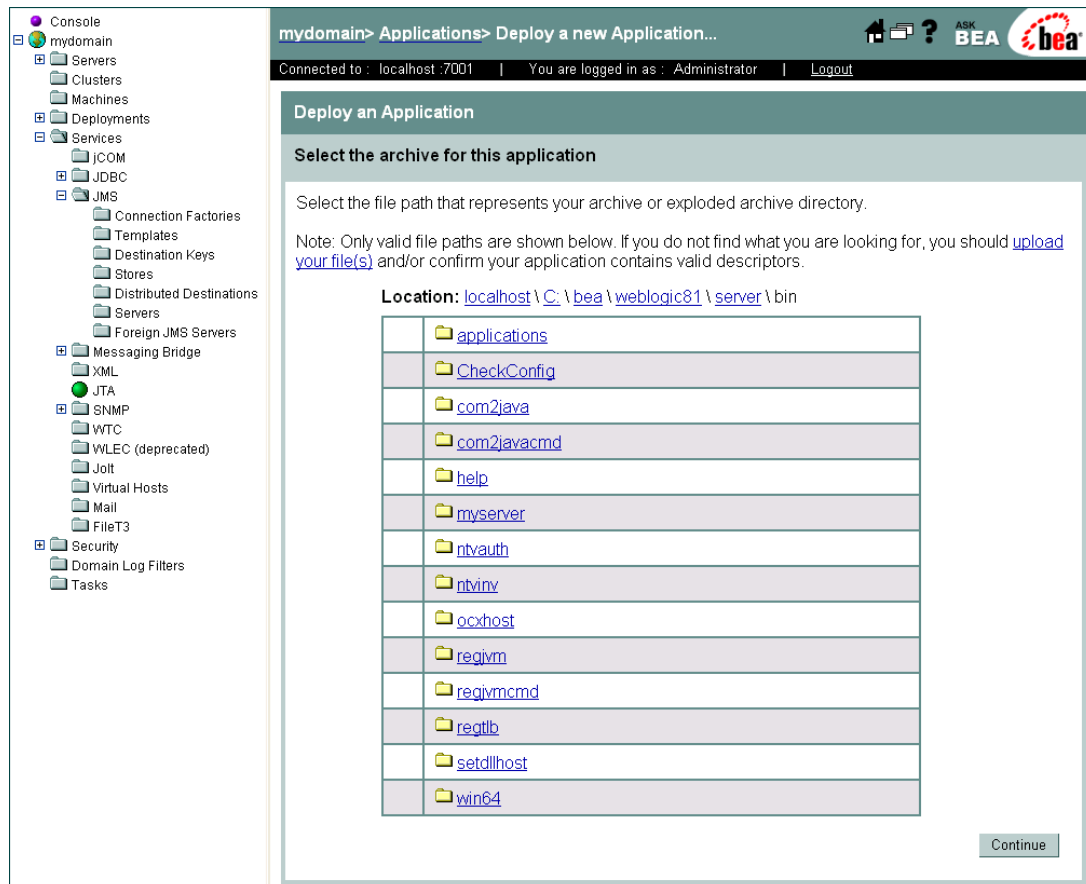
4 Select the appropriate EAR file (see **Figure 45 on page 74**)and click **Continue**.

Figure 45 WebLogic Server Console - Deploy an Application

5 From the Deploy an Application window, enter the name for the selected application and click **Deploy**.

6 From the **Depoly** tab of the Project deployment window, view the status of the applications deployment. When the application is successfully deployed the status appears as Successful (see Figure 46).

Figure 46 WebLogic Server Console - Project Deployment Status
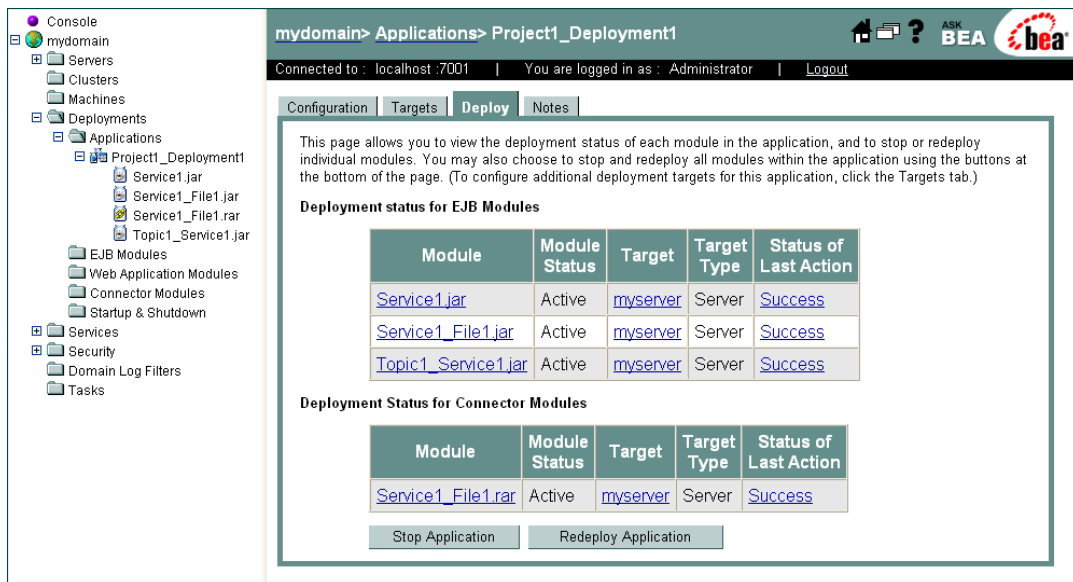
7 Run the Client program to publish to and subscribe from the destinations respectively.

B.5 # Configuring the WebSphere Application Server

To deploy an EAR file in the WebSphere Application Server, start the Administrative Console, and do the following:

Create the Topic or Queue

1 From the left pane of the WebSphere Administrative console, select **Resources** and click **WebSphere JMS Provider**.

2 From the **WebSphere JMS Provider** window, under **Additional Properties**, click **WebSphere Queue Destinations** or **WebSphere Topic Destinations**, depending on whether you are creating a topic or queue. For this example click WebSphere Topic Destinations to create a topic.

3 From the **WebSphere Topics Destinations** (or Queue Destinations) window click the **New** button.

4 From the **Topic** or **Queue Destinations** configuration window, enter a name for the new Topic or Queue in the **Name** field. Enter the same name in the **JNDI Name** field. Enter the string value used to identify the Topic in the **Topic** field (for this example, **Topic1**) as displayed in Figure 47. Click **OK**.

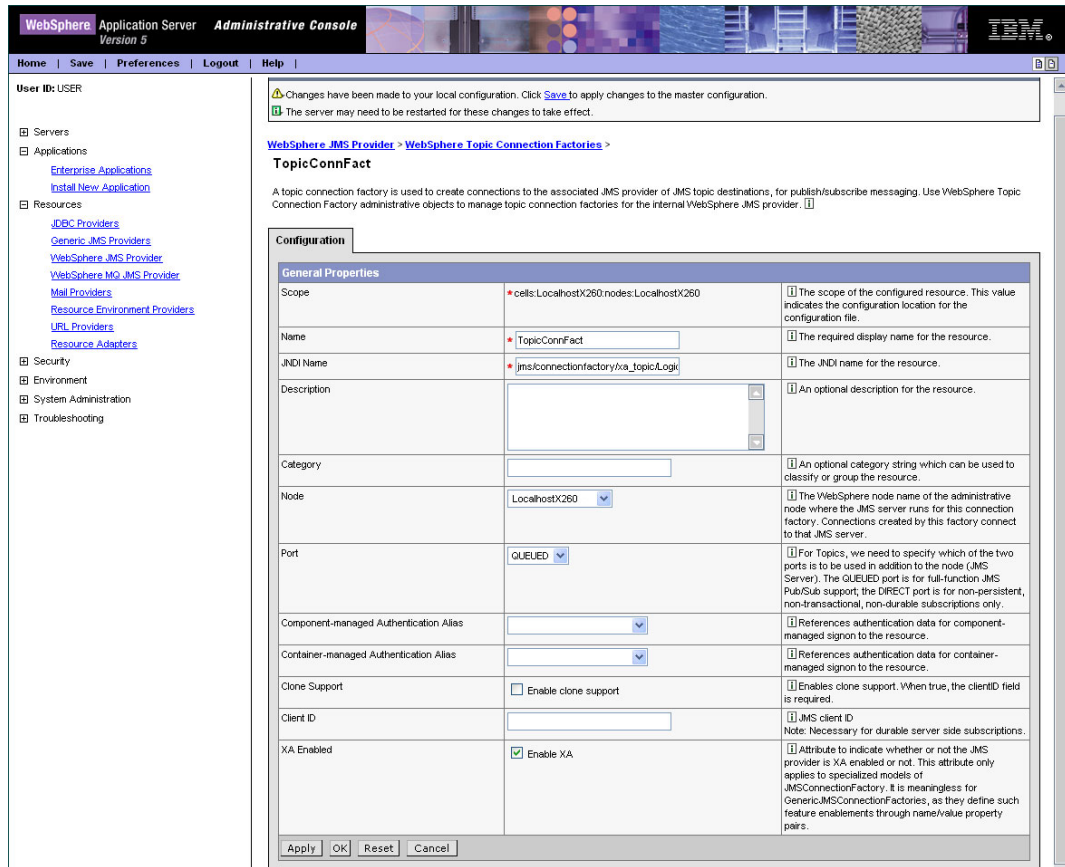**Figure 47** WebSphere Server Administrative Console - Create a Topic
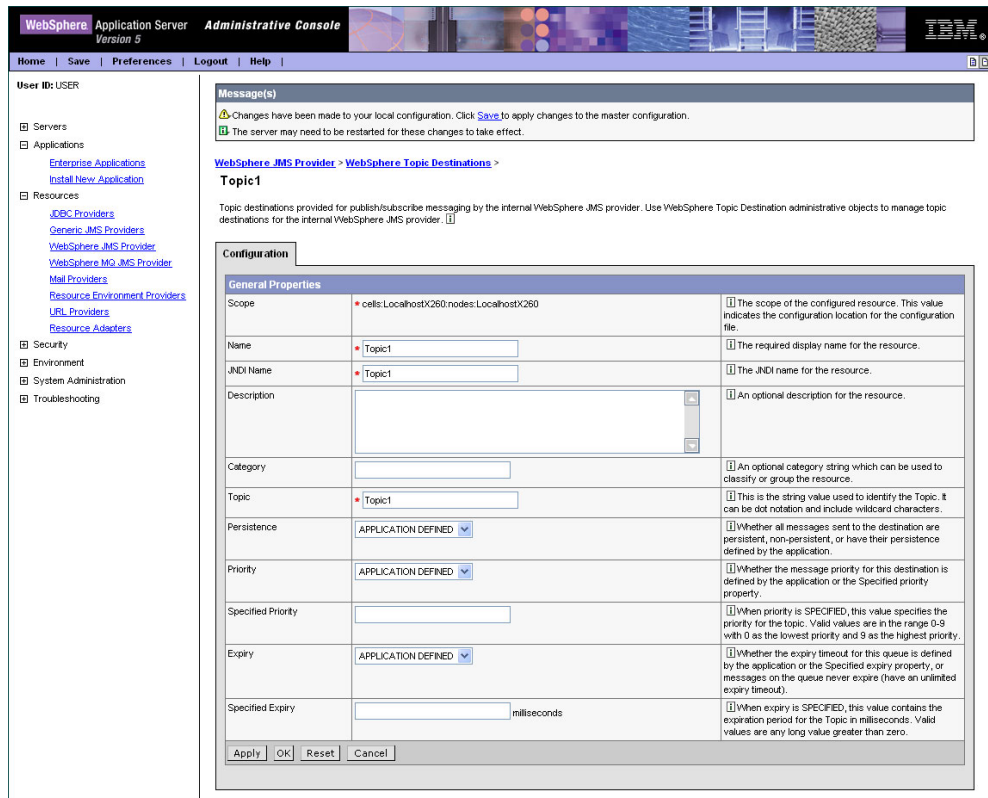
5 From the **WebSphere Topic Destinations** widow, click **WebSphere JMS Provider** to return to the JMS Provider window.

Creating a Connection Factory

1   From the **WebSphere JMS Provider** window, under **Additional Properties**, click **WebSphere Topic Connection Factories** (or Queue Connection Factories if you created a Queue).

2   From the **WebSphere Topic Connection Factories** window, click the **New** button.

3   From the **WebSphere Topic Connection Factories** configuration window, enter the Topic Connection Factory name in the **Name** field (for this example, **TopicConnFact**).

4   Enter the Connection Factory JNDI Name in the **JNDI Name** field. The pattern for the JNDI Name is **jms/connectionfactory/**<*xa_topic* or *xa_queue* (just *topic* or *queue* if xa is not selected as the Transaction Mode)>**/**<*Logical Host name>_<JMS Message Server name* (from the Environment)>**.

For this example, the Connection Factory JNDI name is **jms/connectionfactory/ xa_topic/LogicalHost1_WLMessageSvr1**

5   Make sure that **XA Enabled** is selected (see Figure 48). Click **OK**.

Figure 48   WebSphere Server Administrative Console - Create Connection Factory
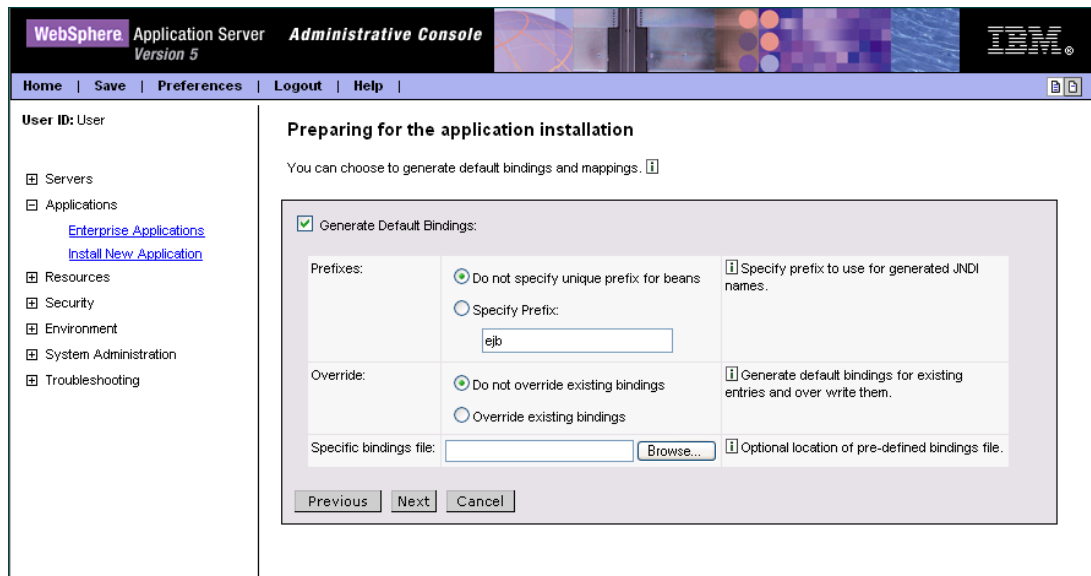

Install the Application

1   From the left pane of the WebSphere Administrative console, select **Applications**, and click **Install New Application**.

2   From the **Preparing for the application installation** window, select **Local path** or **Server path** and click **Browse**. Locate and select the appropriate EAR file (see Figure 49). Click **Next**.

Figure 49   WebSphere Server Administrative Console - Install New Application
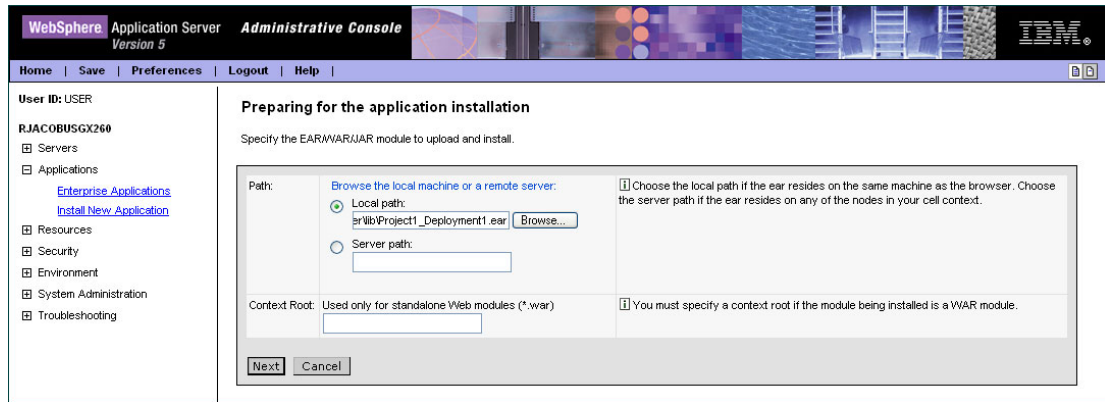
3 From the next window select **Generate Default Bindings** (see Figure 50). Click **Next**.

Figure 50   WebSphere Server Administrative Console - Generate Default Bindings

Note: The following application installation steps may differ depending on the nature of the EAR file.

4 From the **Step 1: Provide options to perform the installation** window (see Figure 51), enter a name for the application and select **Deploy EJBs** (see). Click **Next**.

Figure 51   WebSphere Administrative Console - Provide options to perform the installation
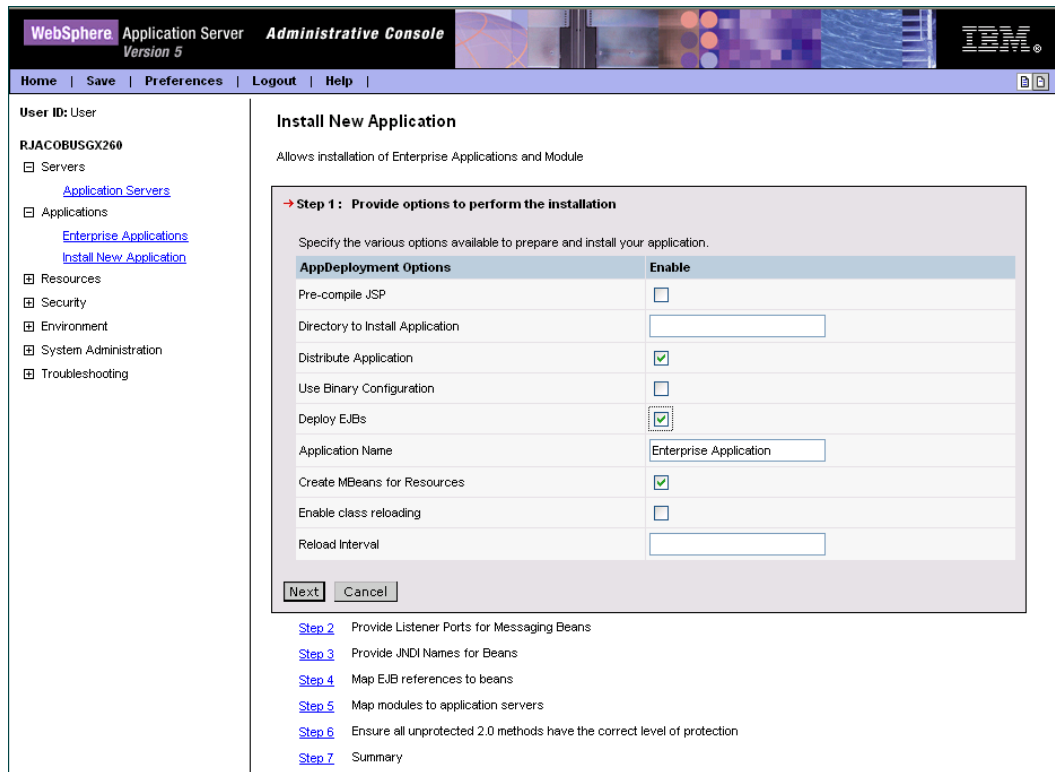
5 No changes are necessary for **Step 2** through **Step 5.** Proceed to the **Step 6: Map modules to application servers** window.

6 From the **Step 6: Map modules to application servers** window, click the Module checkbox to select all of the modules. (seeFigure 52). Click **Next**.

**Figure 52** WebSphere Administrative Console - Map Modules to Application Servers
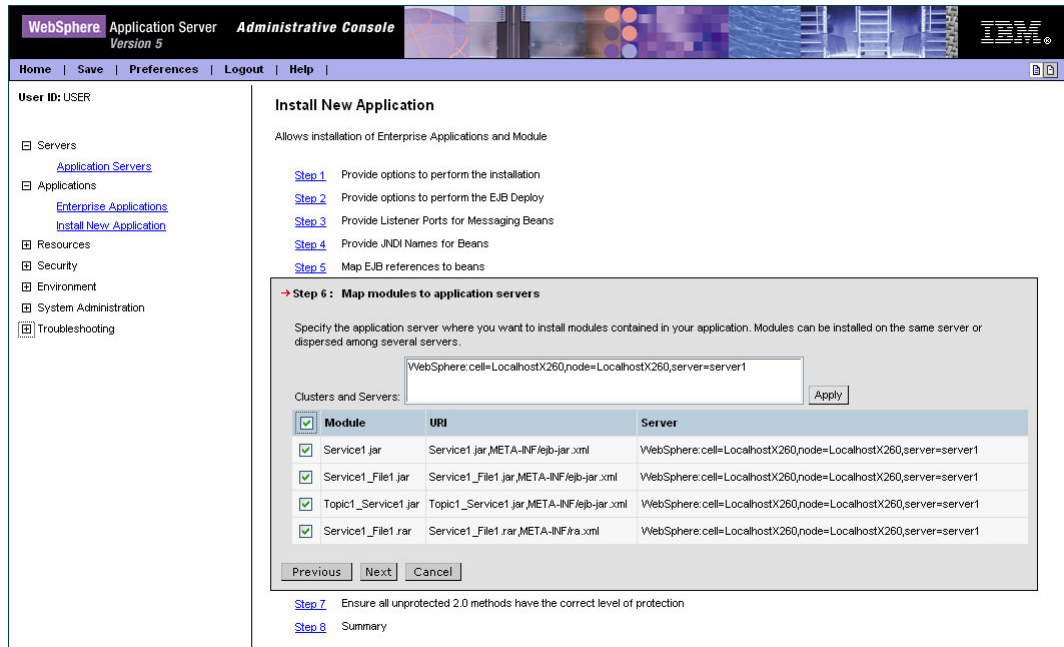
7   From the **Step 7: Ensure all unprotected 2.0 methods have the correct level of protection** window, click the Module checkbox to select all of the modules. Click **Next**

8   From the **Step 8: Summary** window, review the selected values and click **Finish**.

9   WebSphere begins installing the application. This may take several minutes. When the application installs successfully, click **Save to Master Configuration**.

10 To review the application's status or start or stop the application, from the right pane of the Administrative Console, click **Applications**, and click **Enterprise Applications** (see Figure 53).

Figure 53 WebSphere Administrative Console - Enterprise Applications
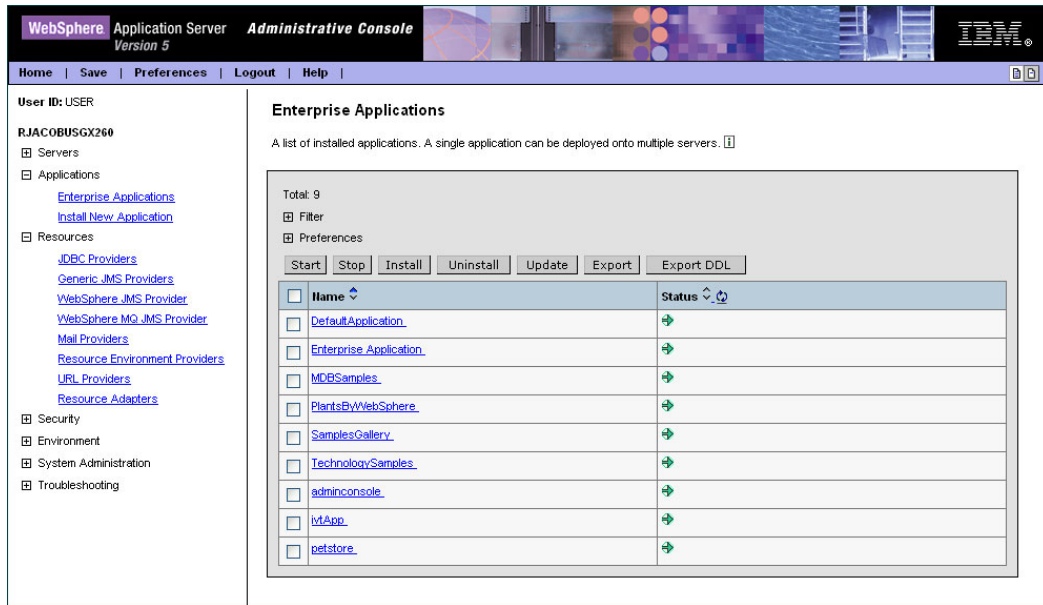
Index

Symbols

- 15

B

book **66**
building a Project **41**
Business Rules
    Java Collaboration Editor **46, 57**

C

Channel **18**
class **16**
configuration parameters
    Channel **18**
    class **16**
    Connector **16**
    General Settings **15**
    Host Name **19**
    MQSeries **18**
    Password **19**
    Port Number **19**
    Property.Tag **17**
    Queue Manager Name **19**
    Transaction Type **15**
    type **17**
    UserID **19**
configuring the WebSphere MQ eWay **12**
Connectivity Map **43**
    populating **44**
Connector **16**

D

Deployment Profile **52**
    activation **52**
    creating **52**

E

EAR file **66**
    creating **67**
eInsight

engine and components **20**
    using with the WebSphere MQ eWay **20**
Environment **49**
    creating **49**
eWay
    configuration parameters
        Channel **18**
        class **16**
        Connector **16**
        General Settings **15**
        Host Name **19**
        MQSeries **18**
        Password **19**
        Port Number **19**
        Property.Tag **17**
        Queue Manager Name **19**
        Transaction Type **15**
        type **17**
        UserID **19**
    configuring the eWays **50**
eWay components **41**
External Applications
    selecting **44**
external system requirements **8**
    MQSeries V5.2 **8**
    WebSphere MQ V5.3 **8**

F

FileeWay.sar **11**

G

General Settings **15**
GMO Class **64**

H

Header fields
    mapping **65**
Host Name **19**

I

implementation **41**
importing a Project **42**
index
    book **66**
installation **10**
    after installing **11**

J

Java Collaboration Definitions
creating **45**, **56**
wizard **45**, **56**
Java Collaboration Editor **46**, **57**
JMS Standard Header
mapping with WebSphereMQ Header fields **65**

M

MA88 patch **9**
MA88 SupportPac **9**
Message Class **64**
Methods **64**
Microsoft Internet Explorer **8**
MQSeriesETD OTD **41**
MQSerieseWay.sar **11**
MQSeriesOTD Class **64**

N

Non-Transactional transaction type **15**

O

operating systems
supported **8**
Overview **7**
overview
MQSeries **7**

P

parameters
Channel **18**
class **16**
Connector **16**
General Settings **15**
Host Name **19**
MQSeries **18**
Password **19**
Port Number **19**
Property.Tag **17**
Queue Manager Name **19**
Transaction Type **15**
type **17**
UserID **19**
Password **19**
PMO Class **64**
Port Number (JMS) **19**
Project
creating **43**
importing **42**

Property.Tag **17**

Q

Queue Manager Name **19**
Queue Manager Name (JMS) **19**

S

sample eWay Project
importing a Project **42**
supported operating systems **8**
system requirements **8**
external system requirements **8**
patches **9**

T

Topic publish/subscribe connection type
requirements for MQSeries V5.2 **9**
Transaction Type **15**
transaction type
Non-Transactional **15**
XA-compliant. **15**
type **17**

W

WebLogic Application Server **66**
creating a connection factory **72**
creating a JMS file server **70**
creating a JMS file store **68**
creating a Topic or Queue **71**
deploying an EAR file **73**
WebSphere Application Server **66**
configuring **75**
creating a connection factory **76**
creating a Queue **75**
creating a Topic **75**
WebSphere MQ eWay Classes
GMO **64**
Message **64**
MQSeriesETD **64**
PMO **64**
WebSphere MQ Header fields
mapping **65**
WebSphere MQ Javadoc
Javadoc **64**
WebSphere MQ Queue **42**
creating **42**
server and manager limits and settings **43**

X

XA-compliant transaction type **15**