SeeBeyond ICAN Suite

WebSphere MQ eWay Intelligent Adapter User's Guide

Release 5.0.4

SEEBEYOND®

SeeBeyond Proprietary and Confidential

The information contained in this document is subject to change and is updated periodically to reflect changes to the applicable software. Although every effort has been made to ensure the accuracy of this document, SeeBeyond Technology Corporation (SeeBeyond) assumes no responsibility for any errors that may appear herein. The software described in this document is furnished under a License Agreement and may be used or copied only in accordance with the terms of such License Agreement. Printing, copying, or reproducing this document in any fashion is prohibited except in accordance with the License Agreement. The contents of this document are designated as being confidential and proprietary; are considered to be trade secrets of SeeBeyond; and may be used only in accordance with the License Agreement, as protected and enforceable by law. SeeBeyond assumes no responsibility for the use or reliability of its software on platforms that are not supported by SeeBeyond.

Version 20040721145150.

Contents

Preface

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Using the WebSphere MQ eWay With eInsight 35

Chapter 5

Chapter 6

About this Document

Welcome to the *WebSphere MQ eWay Intelligent Adapter User's Guide*. This document includes information about installing, configuring, and using the SeeBeyond® WebSphere MQ eWay Intelligent Adapter.

This chapter includes

P.1 SeeBeyond WebSphere MQ eWay Intelligent Adapter

The SeeBeyond WebSphere MQ eWay Intelligent Adapter is a add-on application that utilizes the SeeBeyond Integrated Composite Application Network (ICAN) Suite™ to enable configurable transparent data exchange with WebSphere MQ enabled applications and components.

P.2 Organization of Information

This document provides information about installing, configuring, and using the WebSphere MQ eWay Intelligent Adapter and includes the following chapters:

- **Chapter 1 "Introducing the WebSphere MQ eWay"** provides an overview of the WebSphere MQ eWay.
- **Chapter 2 "Installing the WebSphere MQ eWay"** describes how to install the WebSphere MQ eWay and lists the supported operating systems and system requirements.
- **Chapter 3 "Configuring the WebSphere eWay"** describes the process of configuring the WebSphere MQ eWay to run in your environment.

- Chapter 4 **"Using the WebSphere MQ eWay With eInsight"** describes the features and functionality of the WebSphere MQ eWay using the eInsight Business Process Manager and the eInsight Web Services interface.
- Chapter 5 **"Implementing a WebSphere MQ eWay Project"** describes the features and functionality of the WebSphere MQ eWay using the eGate Integrator and the Collaboration Editor (Java).
- Chapter 6 **"Java Classes and Methods for the WebSphere MQ eWay"** describes the WebSphere MQ eWay Java classes and provides directions for accessing the WebSphere MQ eWay Javadoc.
- Chapter A **"Mapping WebSphere MQ Header Fields"** provides information on the process of mapping between JMS standard header items and WebSphere MQ header fields.

P.3 Intended Audience

This guide is intended for experienced computer users who have the responsibility of helping to set up and maintain a fully functioning ICAN Suite system. This person must also understand any operating systems on which the ICAN Suite will be installed (Windows, UNIX, or HP NonStop Server) and must be thoroughly familiar with Windows-style GUI operations.

P.4 Writing Conventions

The following writing conventions are observed throughout this document.

Table 1 Writing Conventions

Text	Convention	Example
Button, file, icon, parameter, variable, method, menu, and object names.	**Bold** text	- Click **OK** to save and close. - From the **File** menu, select **Exit**. - Select the **logicalhost.exe** file. - Enter the **timeout** value. - Use the **getClassName()** method. - Configure the **Inbound** File eWay.
Command line arguments and code samples	`Fixed` font. Variables are shown in ***bold italic***.	`bootstrap -p` ***password***
Hypertext links	Blue text	http://www.seebeyond.com

Additional Conventions

Windows Systems

For the purposes of this guide, references to “Windows” will apply to Microsoft Windows Server 2003, Windows XP, and Windows 2000.

Path Name Separator

This guide uses the backslash (“\“) as the separator within path names. If you are working on a UNIX system, please make the appropriate substitutions.

P.5 Online Documents

The documentation for the SeeBeyond ICAN Suite is distributed as a collection of online documents. These documents are viewable with the Acrobat Reader application from Adobe Systems. Acrobat Reader can be downloaded from:

http://www.adobe.com

P.6 SeeBeyond Web Site

The SeeBeyond Web site is your best source for up-to-the-minute product news and technical support information. The site’s URL is:

http://www.seebeyond.com

Chapter 1

Introducing the WebSphere MQ eWay

This chapter introduces you to SeeBeyond™ eWay Intelligent Adapter for WebSphere MQ™.

1.1 Overview

IBM's WebSphere MQ

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.

SeeBeyond WebSphere MQ eWay Intelligent Adapter

The SeeBeyond WebSphere MQ eWay Intelligent Adapter (referred to as the WebSphere MQ eWay throughout this document) allows the eGate system to exchange data with IBM's WebSphere MQ. The eGate Integrator, using the WebSphere MQ 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 WebSphere MQ eWay transparently integrates existing systems with IBM's WebSphere MQ. This document explains how to install and configure the WebSphere MQ eWay.

Chapter 2

Installing the WebSphere MQ eWay

This chapter explains how to install the WebSphere MQ eWay and lists the supported operating systems and system requirements for installation.

Chapter Topics

2.1 Supported Operating Systems

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

- Windows 2000 and Windows 2003
- HP NonStop Server G06.22
- HP Tru64 5.1A
- HP-UX 11.0 and 11i (PA-RISC)
- IBM AIX 5.1L and 5.2
- Red Hat Linux 8 (Intel x86)
- Red Hat Enterprise Linux AS 2.1 (Intel x86)
- Sun Solaris 8 and 9
- Suse Linux Enterprise Server 8 (Intel x86)

*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.*

2.2 System Requirements

The system requirements for the WebSphere MQ eWay are the same as those for eGate Integrator. For more information, refer to the *SeeBeyond ICAN Suite Installation Guide*. It is also helpful to review the **Readme.txt** for any additional requirements prior to installation. The **Readme.txt** is located on the installation CD-ROM.

Although the WebSphere MQ eWay, the Repository, and Logical Hosts run on the platforms listed under Supported Operating Systems, the Enterprise Designer requires the Windows operating system. The Enterprise Manager can run on any platform that supports Internet Explorer 6.0.

2.2.1. External System Requirements

The WebSphere MQ eWay requires the following installed on the Logical Host:

- IBM WebSphere MQ V5.3 with CSD07, or MQSeries V5.2 with SP MA88.
- The **com.ibm.mq.jar** file specific to the operating system on which WebSphere MQ is deployed. See **Copying the System Specific mq.jar File to the WebSphere MQ eWay** on page 15.

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.

2.2.2. HP NonStop Server Requirements

To connect to MQSeries for HP NonStop the following must be installed on your HP NonStop Server:

- MQSeries 5.1
- SupportPac MA88
- MQSeries V5.1 CSD02 (downloadable from IBM)

- C2EFIX5 from IBM (C2EFIX5 is not distributed on the IBM Web site. You must contact IBM directly for C2EFIX5.)

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**

2.3 JMS Services and the WebSphere MQ eWay

The WebSphere MQ eWay does not support a JMS interface. The eWay's OTD and MQ connectivity are implemented using the WebSphere MQ base Java classes, not the WebSphere MQ classes for Java Message Service (JMS). The WebSphere MQ eWay has no intrinsic awareness or support for JMS. To use the eWay and JMS services, refer to the *eGate Integrator User's Guide* for information regarding JMS services provided by the ICAN suite for connectivity options.

2.4 Application Server Support

In addition to the above listed Operating Systems, this eWay in outbound mode is supported on WebSphere™ and WebLogic™ Application Servers when using Java Collaborations only. For additional information see the *eGate Integrator User's Guide.*

2.5 Installing the WebSphere MQ 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)

Follow the directions for installing the eGate Integrator in the *eGate Integrator Installation Guide.* After you have installed eGate and other purchased core products, do the following:

1. From the Enterprise Manager, select and upload the following files located in the **Add-ons** directory:
    - **MQSerieseWay.sar** (to install the MQSeries eWay)
    - **FileeWay.sar** (to install the File eWay, used in the sample project)
2. From the **Documentation** directory, select and upload the following file:
    - **MQSerieseWayDocs.sar** (to download the WebSphere MQ eWay User's Guide, Javadoc, and sample projects)
3. 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.

2.5.1. Copying the System Specific mq.jar File to the WebSphere MQ eWay

The WebSphere MQ eWay uses a .jar file that is specific to the operating system on which it is intended to be deployed. This file, **com.ibm.mq.jar**, must be copied from the WebSphere MQ server to the Logical Host's **stcis/lib** directory.

To install the correct .jar file, do the following:

1. From the WebSphere MQ Server, copy the **com.ibm.mq.jar** file and paste it to the Logical Host's **stcis/lib** directory.

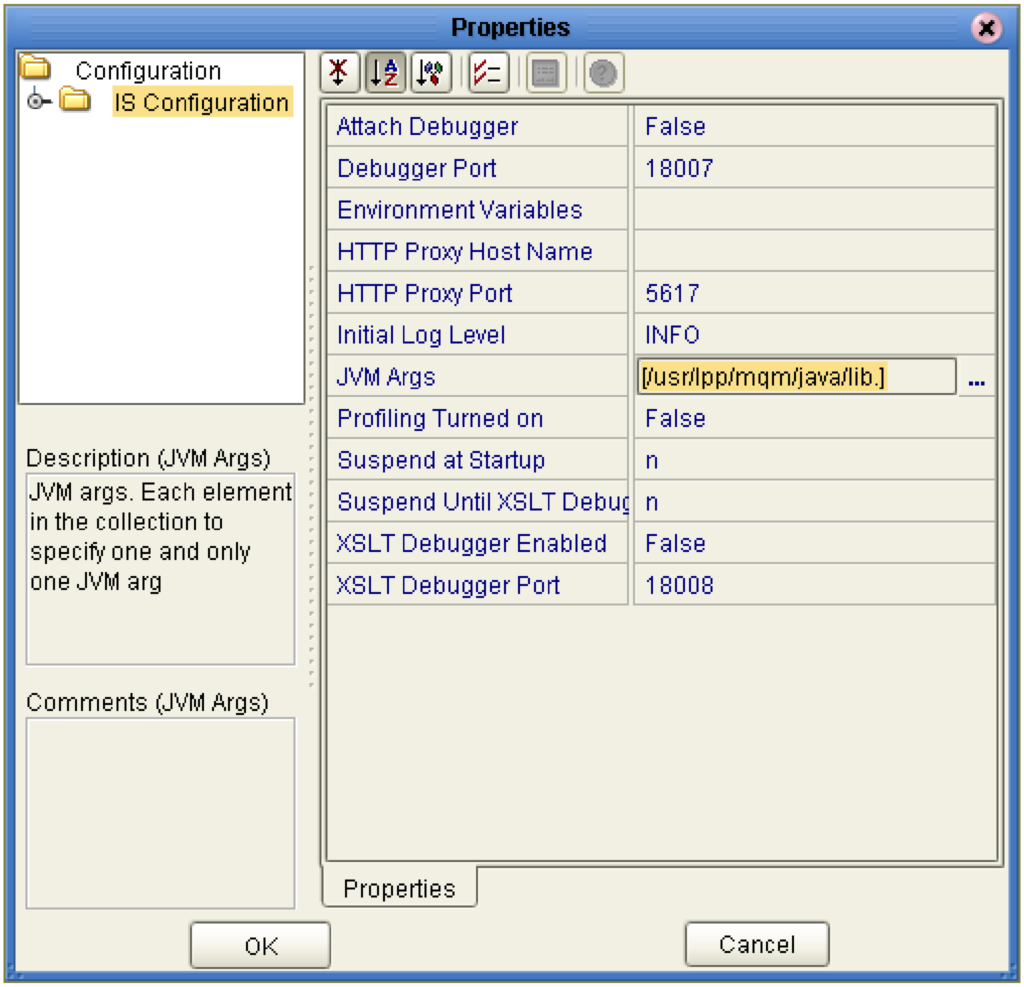
    The **com.ibm.mq.jar** file must come from the MQ server you are running against. If a project is deployed to a different MQ server running on a different operating system, the .jar file on the Logical Host must be replaced with the .jar file from that specific MQ server's operating system.
2. Prior to activating and deploying a WebSphere MQ eWay project, from the Enterprise Designer's Environment Explorer, check the project's **Integration Server**

properties (under the Logical Host in the Environment Explorer tree) to ensure that the **JVM Args** property is set as follows:

```
-Djava.library.path=<path>
```

- For **Windows**, <path> is be the location **from** (not **to**) which **com.ibm.mq.jar** was copied.
- For **z/OS** or **UNIX**, <path> is the location of the **libmqbnd05.so** file (See **Figure 1 on page 16**).

Figure 1 Integration Server Properties - JVM Args



Attach Debugger	False
Debugger Port	18007
Environment Variables	
HTTP Proxy Host Name	
HTTP Proxy Port	5617
Initial Log Level	INFO
JVM Args	[/usr/lpp/mqm/java/lib.]
Profiling Turned on	False
Suspend at Startup	n
Suspend Until XSLT Debug	n
XSLT Debugger Enabled	False
XSLT Debugger Port	18008

Description (JVM Args)

JVM args. Each element in the collection to specify one and only one JVM arg

Comments (JVM Args)

Note: *For eGate version 5.0.2 and earlier, the Integration Server configuration properties do not provide a* **JVM Args** *property. For these versions of eGate it is necessary to add the* **`-Djava.library.path=<path>`** *specification manually. For more information see* **Setting the Integration Server JVM Args Property for eGate Version 5.0.2 and Earlier** *on page 17.*

3 For **z/OS**, if **libmqbnd05.so** does not exist, create a symbolic link to **libwmqjbind.so** named **libmqbnd05.so**.

Setting the Integration Server JVM Args Property for eGate Version 5.0.2 and Earlier

eGate versions 5.0.2 and earlier do not provide the Integration Server configuration property **JVM Args** which requires the **`-Djava.library.path=<path>`** setting required by the WebSphere MQ eWay. For these eGate earlier versions it is necessary to manually add this property into the **ManagementAgent - config.xml** file located in the Logical Host's **stcma\config** subdirectory. To set this property manually, do the following:

1. From the **ManagementAgent - config.xml** file, find the process entry for the integration server:

```
<process>
    <name>IntegrationSvr1</name>
        <start-command>
        <dir>$LogicalHostRoot/stcis</dir>
        <command-line>
            <arg>$LogicalHostRoot/jre/bin/java</arg>
            <arg>-Dlogical.host.root.dir=$LogicalHostRoot</arg>
            <arg>...</arg>
[...]
```

2. Add an <arg> entry for the required java.library.path setting as follows:

```
<arg>-Djava.library.path=/directory/to/webspheremq/java/lib</arg>
```

3. Save and close the file.

Chapter 3

Configuring the WebSphere eWay

This chapter describes how to create and configure the WebSphere MQ eWay.

Chapter Topics

3.1 Creating and Configuring the WebSphere MQ eWay

All eWays contain a set of parameters with properties unique to that eWay type. After the eWays are established and a WebSphere MQ External System is created in the project's Environment, the eWay parameters can be modified for your specific system. The WebSphere MQ 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 WebSphere MQ Connectivity Map properties will vary depending on whether the eWay 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 WebSphere MQ External System window.

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

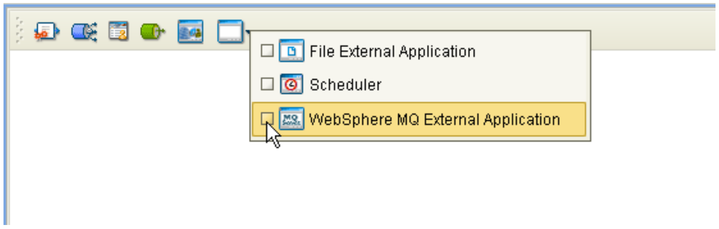
3.1.1 Selecting WebSphere MQ as the External Application

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

To create the WebSphere MQ External Application

1. From the Connectivity Map toolbar, click the **External Applications** icon.
2. Select the **WebSphere MQ External Application** from the menu (see Figure 2). The selected WebSphere MQ External Application icon appears on the Connectivity Map toolbar.

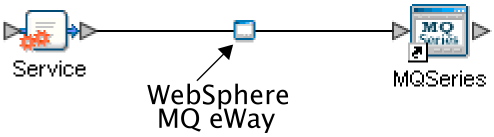
Figure 2 External Applications Selection Menu


☐ File External Application
☐ Scheduler
☐ WebSphere MQ External Application

3. Drag the new **WebSphere MQ 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 3).

Figure 3 eWay Location



When WebSphere MQ is selected as the External Application, it automatically applies the default WebSphere MQ 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 Editor**.

3.1.2 Creating Custom Properties for a WebSphere MQ 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 WebSphere MQ 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. The Templates dialog box appears.
2. From the Templates dialog box, select **Inbound** or **Outbound** as the eWay configuration type and click **OK**.
3. The eWay **Properties Editor** opens to the Inbound or Outbound WebSphere MQ Connectivity Map properties. Make any necessary modifications and click **OK** to save the settings.

Modifying the WebSphere MQ 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 Editor** appears.
2. Make any necessary modifications to the Environment parameters of the WebSphere MQ eWays, and click **OK** to save the settings.

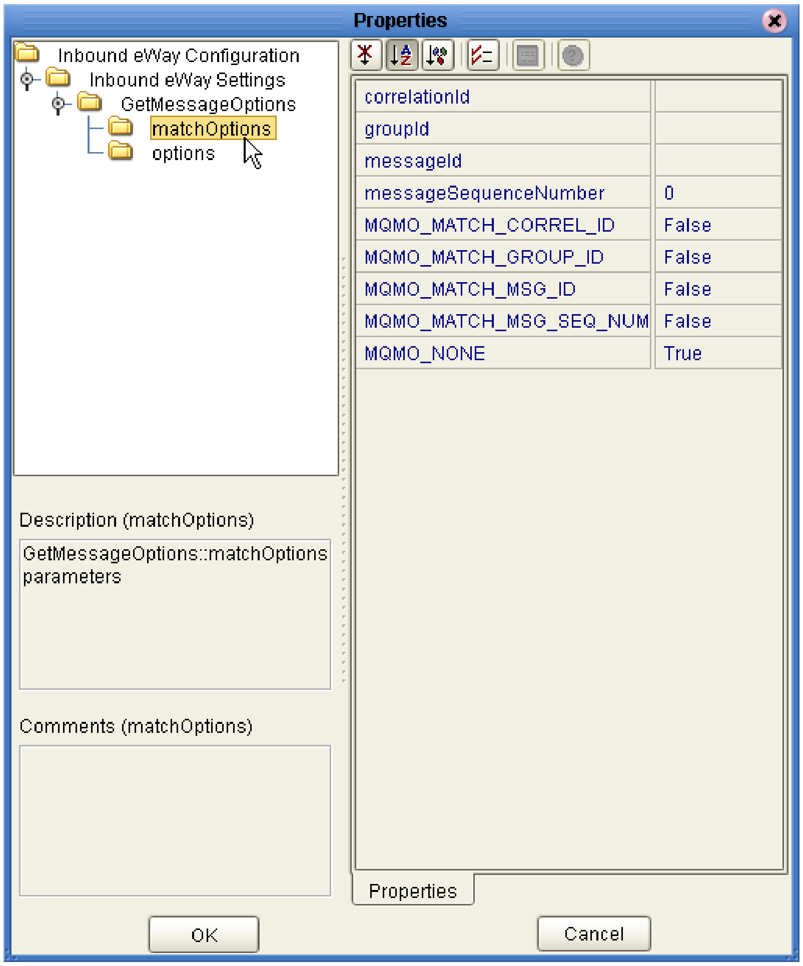
3.1.3. Using the Properties Editor

Modifications to the eWay configuration properties are made from the WebSphere MQ eWay Properties Editor.

To modify the default eWay configuration properties do the following:

1. Open the Properties Editor to the WebSphere MQ eWay properties you want to edit. The WebSphere MQ Inbound and Outbound eWays have two 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-left pane of the Properties Editor, select a properties directory. The parameters contained in that directory are now displayed in the right pane of the Properties Editor. For example, from the Inbound eWay Connectivity Map Properties, click on the **matchOptions** properties directory to display this section’s editable parameters in the right pane, as shown in Figure 4.

Figure 4 Properties Editor -- WebSphere MQ Properties



Description (matchOptions)

GetMessageOptions::matchOptions parameters

Comments (matchOptions)

correlationId	
groupId	
messageId	
messageSequenceNumber	0
MQMO_MATCH_CORREL_ID	False
MQMO_MATCH_GROUP_ID	False
MQMO_MATCH_MSG_ID	False
MQMO_MATCH_MSG_SEQ_NUM	False
MQMO_NONE	True

3 Click on any property field to make it editable. For example, click on the **messageId** parameter to edit the messageId value. If a parameter's value is true/false or multiple choice, the field, when selected, reveals a submenu of property options. If a parameter requires that you type in a value, such as a name or password, the property field provides space to type in the value and an ellipsis (. . .) button.

Click on the ellipsis (. . .) in the properties field to open a separate configuration dialog box. This is helpful for entering 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 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 Editor and save the changes.

3.2 WebSphere MQ eWay Properties

The WebSphere MQ eWay's Properties are organized as follows:

Inbound WebSphere MQ eWay Connectivity Map Properties on page 22

Outbound WebSphere MQ eWay Connectivity Map Properties on page 29

Inbound WebSphere MQ eWay Environment Properties on page 29

Outbound WebSphere MQ eWay Environment Properties on page 31

Note: *Creating customized individual OTD configuration settings can override the default eWay OTD configuration settings.*

3.3 Inbound WebSphere MQ eWay Connectivity Map Properties

The inbound WebSphere MQ eWay parameters, accessed from the Connectivity Map, are organized into the following sections:

- **Inbound eWay Settings** on page 22
- **Inbound eWay Settings, GetMessageOptions, matchOptions** on page 23
- **Inbound eWay Settings, GetMessageOptions, options** on page 26

3.3.1. Inbound eWay Settings

This section contains the following parameters:

- **Queue Name** on page 22
- **Schedule Interval** on page 23

Queue Name

Description

Specifies the name of the local queue from which messages are picked up (subscribed).

Required Value

The name of the local WebSphere MQ Queue.

Note: *Inbound (poll/receive) mode eWays will not function if a non-local queue is specified. Non-local queues include alias queues and local queue definitions to remote queues. See* **Accessing Non-Local Queue Managers and Non-Local Queues** *on page 34.*

Schedule Interval

Description

Specifies the polling interval in milliseconds at which the subscribed queue is polled for messages. This is the duration of the pause, in milliseconds, between attempts to get messages from the queue.

Required Value

The number of milliseconds at which the queue is polled. The configured default is **10000** (or 10 seconds).

3.3.2. Inbound eWay Settings, GetMessageOptions, matchOptions

This section contains the following parameters:

- **correlationId** on page 23
- **groupId** on page 23
- **messageId** on page 24
- **messageSequenceNumber** on page 24
- **MQMO_MATCH_CORREL_ID** on page 24
- **MQMO_MATCH_GROUP_ID** on page 24
- **MQMO_MATCH_MSG_ID** on page 25
- **MQMO_MATCH_MSG_SEQ_NUMBER** on page 25
- **MQMO_NONE** on page 25

correlationId

Description

Specifies the correlation identifier of the message to be retrieved. Normally the queue manager returns the first message with a message identifier and correlation identifier that matches the identifiers specified.

Required Value

The correlation identifier of the message.

groupId

Description

Specifies the byte string that identifies the message group to which the physical message belongs.

Required Value

A byte string that indicates the message group.

messageId

Description

For an MQGET call, this field specifies the message identifier of the message to be retrieved. Normally, the queue manager returns the first message with a message identifier and correlation identifier that matches those identifiers specified.

For an MQPUT call, this specifies the message identifier to use.

Required Value

The message identifier.

messageSequenceNumber

Description

Specifies the sequence number of a logical message within a group.

Required Value

The sequence number of the logical message within a group.

MQMO_MATCH_CORREL_ID

Description

Specifies that the retrieved message must have a correlation identifier that matches the value of the correlationId parameter. The values are:

- **True**: Indicates that the message must have a matching correlation identifier.
- **False**: Indicates that the correlation identifier is ignored and any correlation identifier will be accepted.

This match is in addition to any other matches that may apply (for example, the message identifier).

Required Value

True or **False**. The configured default is **False**.

MQMO_MATCH_GROUP_ID

Description

Specifies that the retrieved message must have a group identifier that matches the value of the groupId parameter. The values are:

- **True**: Indicates that the message must have a matching group identifier.
- **False**: Indicates that the group identifier is ignored and any group identifier is accepted.

This match is in addition to any other matches that may apply (for example, the correlation identifier).

Required Value

True or **False**. The configured default is **False**.

MQMO_MATCH_MSG_ID

Description

Specifies that the retrieved message must have a message identifier that matches the value of the messageId parameter. The values are:

- **True**: Indicates that the message must have a matching message identifier.
- **False**: Indicates that the message identifier is ignored and any message identifier is accepted.

This match is in addition to any other matches that may apply (for example, the correlation identifier).

Required Value

True or **False**. The configured default is **False**.

MQMO_MATCH_MSG_SEQ_NUMBER

Description

Specifies that the retrieved message must have a message sequence number that matches the value of the messageSequenceNumber parameter. The values are:

- **True**: Indicates that the message must have a matching message sequence number.
- **False**: Indicates that the message sequence number is ignored and any message sequence number is accepted.

This match is in addition to any other matches that may apply (for example, the group identifier).

Required Value

True or **False**. The configured default is **False**.

MQMO_NONE

Description

Specifies that no matches are to be used in selecting the message to be returned. All messages on the queue are eligible for retrieval (subject to some MQGMO_ options...).

Required Value

True or **False**. The configured default is **True**.

3.3.3. Inbound eWay Settings, GetMessageOptions, options

This section contains the following parameters:

- **MQGMO_ACCEPT_TRUNCATED_MSG** on page 26
- **MQGMO_COMPLETE_MSG** on page 26
- **MQGMO_FAIL_IF_QUIESCING** on page 27
- **MQGMO_SYNCPOINT** on page 27
- **MQGMO_SYNCPOINT_IF_PERSISTENT** on page 27
- **MQGMO_WAIT** on page 28
- **waitInterval** on page 28

MQGMO_ACCEPT_TRUNCATED_MSG

Description

Specifies whether a truncated message is accepted as a complete message. If the message buffer is too small to hold the complete message, this option allows the **MQGET** call to fill the buffer with as much as it can hold and complete its processing. Without this option, in the given situation, the **MQGET** call will still be filled to capacity, but the processing will not be considered completed. The values are:

- **True**: Indicates that a truncated message is accepted as a complete message.
- **False**: Indicates that a truncated message is not considered as a complete message.

Required Value

True or **False**. The configured default is **True**.

MQGMO_COMPLETE_MSG

Description

Specifies that only a complete logical message can be returned by calling **MQGET**. If the logical message is segmented, the queue manager reassembles the segments and returns the complete logical message to the application; the fact that the logical message was segmented is not apparent to the eWay. The values are:

- **True**: Indicates that only a complete logical message can be returned by calling **MQGET**.
- **False**: Indicates that a complete logical message is not required.

Required Value

True or **False**. The configured default is **False**.

MQGMO_FAIL_IF_QUIESCING

Description

Forces the **MQGET** call to fail if the queue manager is in the quiescing state. The values are:

- **True**: Indicates that calling **MQGET** fails if the queue manager is in the quiescing state.
- **False**: Indicates that calling **MQGET** does not fail if the queue manager is in the quiescing state.

Required Value

True or **False**. The configured default is **True**.

MQGMO_SYNCPOINT

Description

Forces the **MQGET** call to get the message under syncpoint control; the message is marked as being unavailable to other applications, but it is deleted from the queue only when the unit of work is committed. The message is made available again if the unit of work is backed out. **DO NOT ENABLE THIS METHOD if XA mode is to be used**. The values are:

- **True**: Indicates that calling **MQGET** gets the message under syncpoint control.
- **False**: Indicates that **MQGET**, when called does not get the message under syncpoint control.

Required Value

True or **False**. The configured default is **False**.

MQGMO_SYNCPOINT_IF_PERSISTENT

Description

Forces the **MQGET** call to get the message under syncpoint control if the message is persistent. **DO NOT ENABLE this option if XA mode is to be used**. The values are:

- **True**: Indicates that calling **MQGET** gets the message under syncpoint control if the message is persistent.
- **False**: Indicates that **MQGET**, when called does not get the message under syncpoint control if the message is persistent.

Required Value

True or **False**. The configured default is **False**.

MQGMO_WAIT

Description

Specifies that an **MQ GET** call waits (block/suspend) until a message becomes available in the queue. The values are:

- **True**: Indicates that an **MQ GET** call waits until a message becomes available in the queue.
- **False**: Indicates that an **MQ GET** call does not wait until a message becomes available in the queue.

Required Value

True or **False**.

waitInterval

Description

Specifies how long (in milliseconds) an **MQ GET** call waits for a message to become available in the queue. This parameter is used in conjunction with **MQGMO_WAIT**. If **MQGMO_WAIT** is set to false, waitInterval is not used.

Required Value

A number indicating the period of time, in milliseconds, that an **MQ GET** call waits for a message to become available in the queue. Specifying a negative value indicates that the wait will last indefinitely.

Setting this value to a negative number causes the polling eWay to execute **MQ GET** calls with a wait interval of **MQWI_UNLIMITED**. With this type of get call, the eWay will block indefinitely until a suitable message is available. If the Integration Server (in association with the logical host) is commanded to shut down or restart while the eWay is still blocked, the Integration Server will not be able to proceed until the eWay is unblocked by the availability of a suitable MQ message.

The same limitation affects the **non-polling** use of the eWay. The WebSphere MQ eWay's **OTD GMO** structure exposes a method named **setUnlimitedWait()** to Java Collaborations that, when used, sets the **waitInterval** to the value **MQWI_UNLIMITED**. If using **setUnlimitedWait()** causes the eWay to block indefinitely during a subsequent get call, the Integration Server will be unable to shut down until the eWay is unblocked.

3.4 Outbound WebSphere MQ eWay Connectivity Map Properties

The outbound WebSphere MQ eWay parameters, accessed from the Connectivity Map, are organized into the following sections:

- **Outbound eWay Settings** on page 29

3.4.1. Outbound eWay Settings

This section contains the following parameters:

- **Queue Name** on page 29

Queue Name

Description

Specifies the name of queue to which the message is published. This parameter is optional. The queue name may also be specified manually in the Business Process or Collaboration that effects the put.

Required Value

The queue name to which the message is published.

3.5 Inbound WebSphere MQ eWay Environment Properties

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

- **Inbound eWay Environment Configuration** on page 29

3.5.1. Inbound eWay Environment Configuration

This section contains a set of top level parameters:

- **Channel Name** on page 30
- **Host Name** on page 30
- **Is XA** on page 30
- **Password** on page 30
- **Port Number** on page 30
- **Queue Manager Name** on page 31
- **UserID** on page 31

Channel Name

Description

Specifies the name of the channel being used.

Required Value

The name of the channel.

Host Name

Description

Specifies name of the computer on which the queue manager resides. This property must be left blank to cause the eWay to use Bindings mode.

Required Value

The name of the specific queue manager host. Leave the value blank to cause the eWay to use Bindings mode.

Is XA

Description

Specifies whether the eWay participates in global transactions coordinated by the Integration Server. The property values are:

- **True**: Indicates that XA mode is enabled.
- **False**: Indicates that XA mode is disabled.

Required Value

True or **False**. The configured default is **False**.

Password

Description

Specifies the user password required to access the queue manager. If a password is not required, leave this parameter blank.

Required Value

A user password that grants access to a specific queue manager.

Port Number

Description

Specifies the number of the listen port on which the queue manager is bound.

Required Value

A number indicating the port on which the queue manager is bound.

Queue Manager Name

Description

Specifies the name of the local queue manager to which the eWay connects.

Required Value

The name of the local queue manager.

Note: *Use only a local queue manager name in the eWay Environment Configuration, whether bindings or client mode is used. See* **Accessing Non-Local Queue Managers and Non-Local Queues** *on page 34.*

UserID

Description

Specifies the user ID required to access the queue manager. If none is required, leave this parameter blank.

Required Value

A User ID required to access the queue manager.

3.6 Outbound WebSphere MQ eWay Environment Properties

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

- **Outbound eWay Environment Configuration** on page 31

3.6.1. Outbound eWay Environment Configuration

This section contains a set of top level parameters:

- **Channel Name** on page 32
- **Host Name** on page 32
- **Is XA** on page 32
- **Password** on page 32
- **Port Number** on page 32
- **Queue Manager Name** on page 33
- **UserID** on page 33

Channel Name

Description

Specifies the name of the channel being used.

Required Value

The name of the channel.

Host Name

Description

Specifies name of the computer on which the queue manager resides. This property must be left blank to cause the eWay to use Bindings mode.

Required Value

The name of the specific queue manager host. Leave the value blank to cause the eWay to use Bindings mode.

Is XA

Description

Specifies whether the eWay participates in global transactions coordinated by the Integration Server. The property values are:

- **True**: Indicates that XA mode is enabled.
- **False**: Indicates that XA mode is disabled.

Required Value

True or **False**. The configured default is **False**.

Password

Description

Specifies the user password required to access the queue manager. If a password is not required, leave this parameter blank.

Required Value

A user password that grants access to a specific queue manager.

Port Number

Description

Specifies the number of the listen port on which the queue manager is bound.

Required Value

A number indicating the port on which the queue manager is bound.

Queue Manager Name

Description

Specifies the name of the local queue manager to which the eWay connects.

Required Value

The name of the local queue manager.

Note: *Use only a local queue manager name in the eWay Environment Configuration, whether bindings or client mode is used. See* **Accessing Non-Local Queue Managers and Non-Local Queues** *on page 34.*

UserID

Description

Specifies the user ID required to access the queue manager. If none is required, leave this parameter blank.

Required Value

A User ID required to access the queue manager.

3.7 Polling and Reconnection Logic

The WebSphere MQ eWay runs in two modes: Polling and Non-Polling

Polling Mode

The eWay in its inbound capacity, runs in polling mode. If the eWay looses its connection, errors and monitor alerts are posted, and the eWay automatically attempts to reconnect until the connection is reestablished. Once it reconnects, polling resumes the retrieval of available messages.

Non-Poll Mode

An eWay performing a put or a get operation, runs in non-poll mode. If the eWay looses its connection, it informs the Collaboration that the connection is down by throwing an exception. It is the Collaboration author’s responsibility to add business logic to the Collaboration to catch the exception and preserve the data (for example, saving undelivered messages to a safe location or invoking **backout()** on the eWay’s Message OTD).

3.8 Accessing Non-Local Queue Managers and Non-Local Queues

Specify only **local queue manager names** in the eWay Environment configuration (for both bindings or client mode), and specify only **local queue names** for the inbound Connectivity Map configuration. Non-local queues include alias queues and local queue definitions to remote queues.

If you specify a non-local queue manager and non-local queue (either by specifying them in the eWay configuration, or by specifying them in a “connect” call in a Collaboration), be aware that, due to the inability to query non-local objects, the eWay cannot proactively validate the connection (reconnecting, if necessary) before each OTD operation. When non-local objects are used, the Collaboration is responsible for handling failures and executing recourse actions (including reestablishing the queue manager or queue connection).

Collaboration that attempt to access non-local queues must use the OTD method, **accessQueue(String, String)** rather than the earlier method **accessQueue(String)**. For more information, refer to the *WebSphere MQ eWay Javadoc.*

3.9 Alerting and Logging

eGate provides an alerting and logging feature that allows monitoring of messages, and captures any adverse messages in order of severity based on configured severity level. For information on how to enable and configure logging, see the ***eGate Integrator User’s Guide***.

Note: *The alerts/status notifications for the WebSphere MQ eWay are currently limited to started, running, stopping, stopped, and connection lost.*

Chapter 4

Using the WebSphere MQ eWay With eInsight

This chapter describes how to use the WebSphere MQ 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 WebSphere MQ eWay With eInsight

An eInsight Business Process Activity can be associated with the WebSphere MQ 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 WebSphere MQ 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 WebSphere MQ eWay.

4.3 WebSphere MQ eWay eInsight Sample Projects

Two sample projects for the WebSphere MQ eWay (using eInsight Business Processes) are included with the installation:

- **WebSphereMQSeries_BPEL_Get_Sample.zip**: See**The MQ_BP_Get_ Sample Project** on page 37
- **WebSphereMQSeries_BPEL_Put_Sample.zip**: See**The MQ_BP_Put_ Sample Project** on page 48

This chapter also walks you through the creation of these sample projects.

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** from the shortcut menu. The **Import Manager** appears.
3. Browse to the directory that contains the sample project zip file. Select the sample file (for this sample, **WebSphereMQSeries_BPEL_Get_Sample.zip**) and click **Import**. After the sample project is successfully imported, click **Close**.

4. From the Project Explorer tree, rename the project to **MQ_BP_Get_ Sample.**
5. 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 44)
   - Configure the eWay properties for your specific system (see **Configuring the eWay Properties** on page 45)
   - Create a **Deployment Profile** (see **Creating and Activating the Deployment Profile** on page 46)

4.5 The MQ_BP_Get_ Sample Project

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

1. The WebSphere MQ 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.

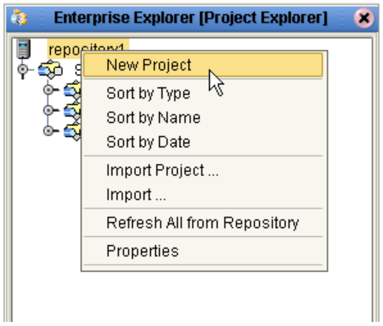
The following pages provide step by step directions for manually creating the **MQ_BP_Get_ Sample** project.

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 5). A new project (Project1) appears on the Project Explorer tree.

Figure 5 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

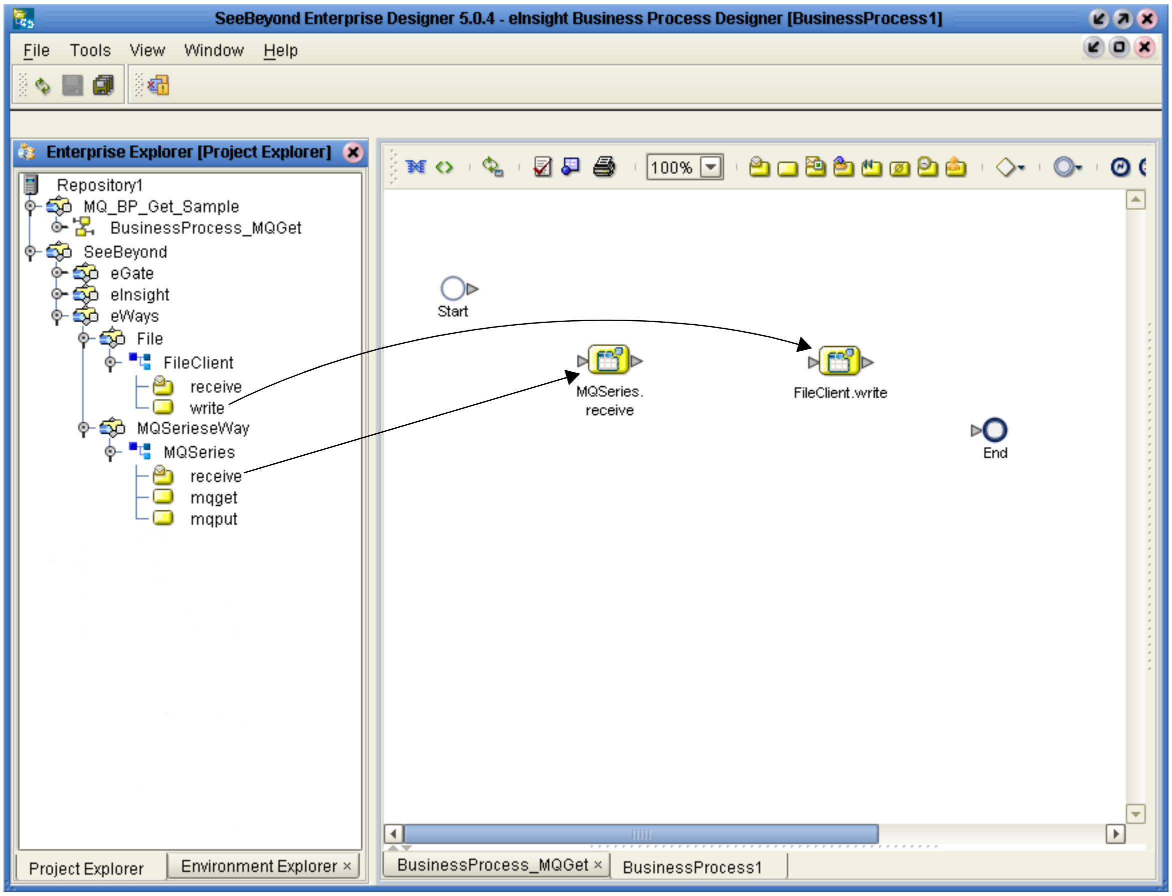
1. From the Enterprise Designer's Project Explorer tree, right-click **MQ_BP_Get_ Sample**, and select **New** > **Business Process** from the shortcut menu. The eInsight Business Process Designer appears and **BusinessProcess1** is added to the Project Explorer tree. Rename the Business Process to **BusinessProcess_MQGet**.

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 6:

   - **receive**, under SeeBeyond > eWays > MQSerieseWay > MQSeries
   - **write**, under SeeBeyond > eWays > File > FileClient

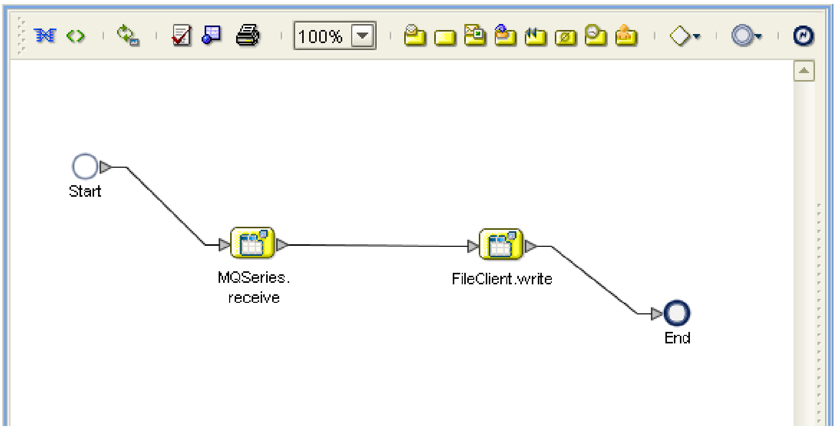
Figure 6 eInsight Business Process Designer - Populate the Canvas



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 7.

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

Figure 7 eInsight Business Process Designer - Link the Modeling Elements

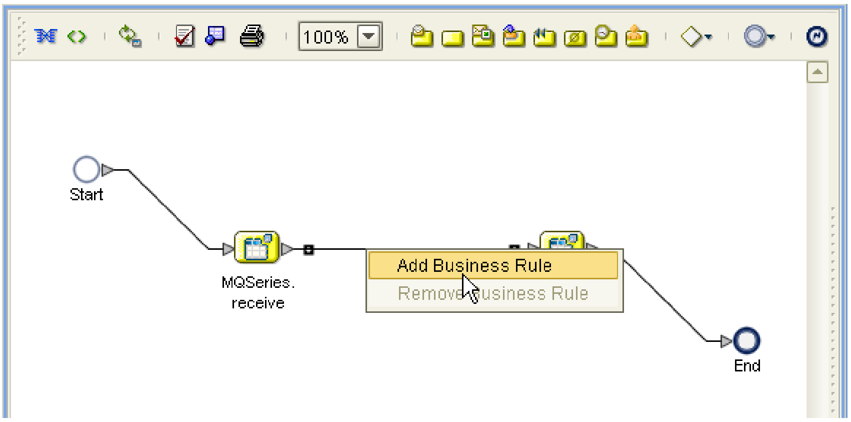


Configuring the Modeling Elements

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 8.

Figure 8 eInsight Business Process Designer - Adding Business Rules



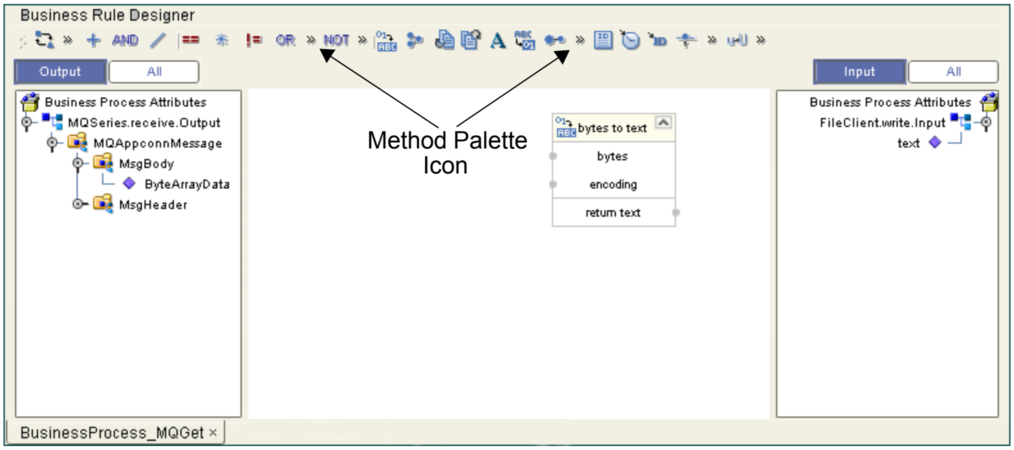
2 From the eInsight Business Process Designer toolbar, click the **Map Business Process Attributes** button. The Business Rule Designer appears at the bottom of the eInsight Business Process Designer.

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 9). 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 9).

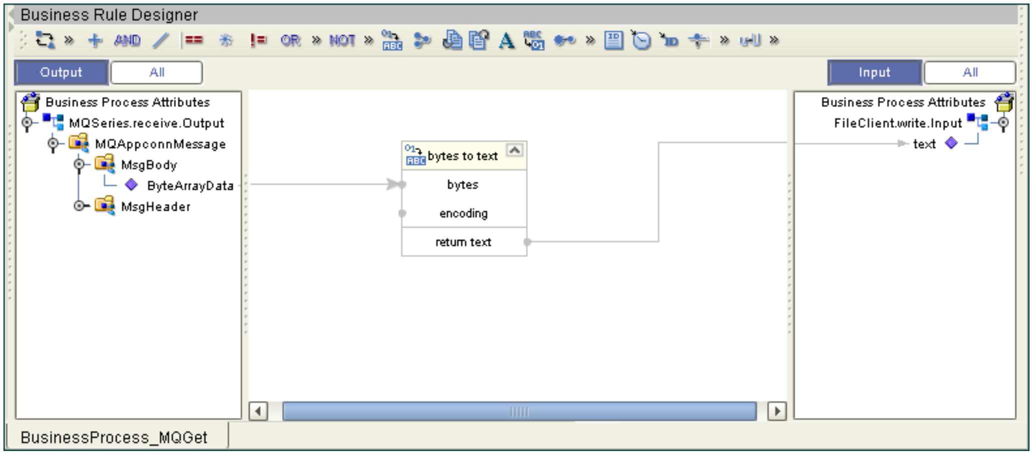
Figure 9 eInsight Business Rule Designer



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 10).

Figure 10 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:

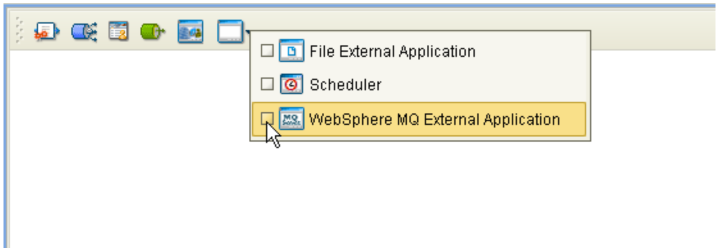
1. Click the **External Application** icon on the Connectivity Map toolbar (see Figure 11).

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 WebSphere MQ as an External System to use in your Connectivity Map (see Figure 11).

Figure 11 Connectivity Map - External Applications


□ File External Application
□ Scheduler
□ WebSphere MQ External Application

To create the External Applications used by the **MQ_BP_Get_ Sample** project do the following:

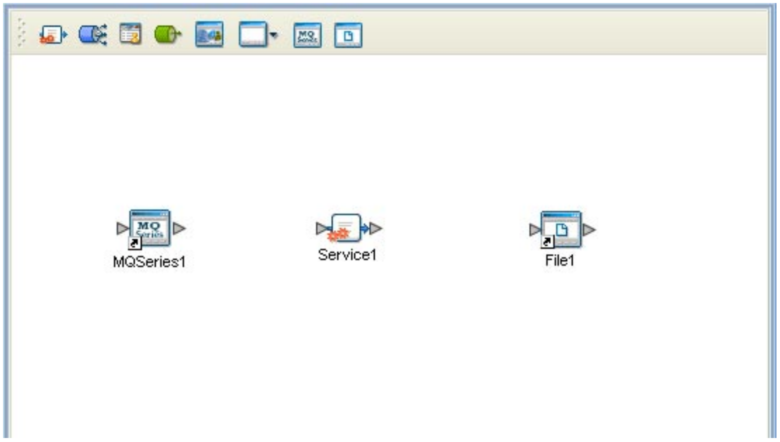
1. Click the **External Application** icon on the Connectivity Map toolbar.
2. Select the applications needed for your project (for this sample, the **File** and **WebSphere MQ** External Applications). Icons representing the selected applications are added to the Connectivity Map toolbar.

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 12:
   - **WebSphere MQ External Application**
   - **Service** (A service is a container for 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 12 Connectivity Map with Components



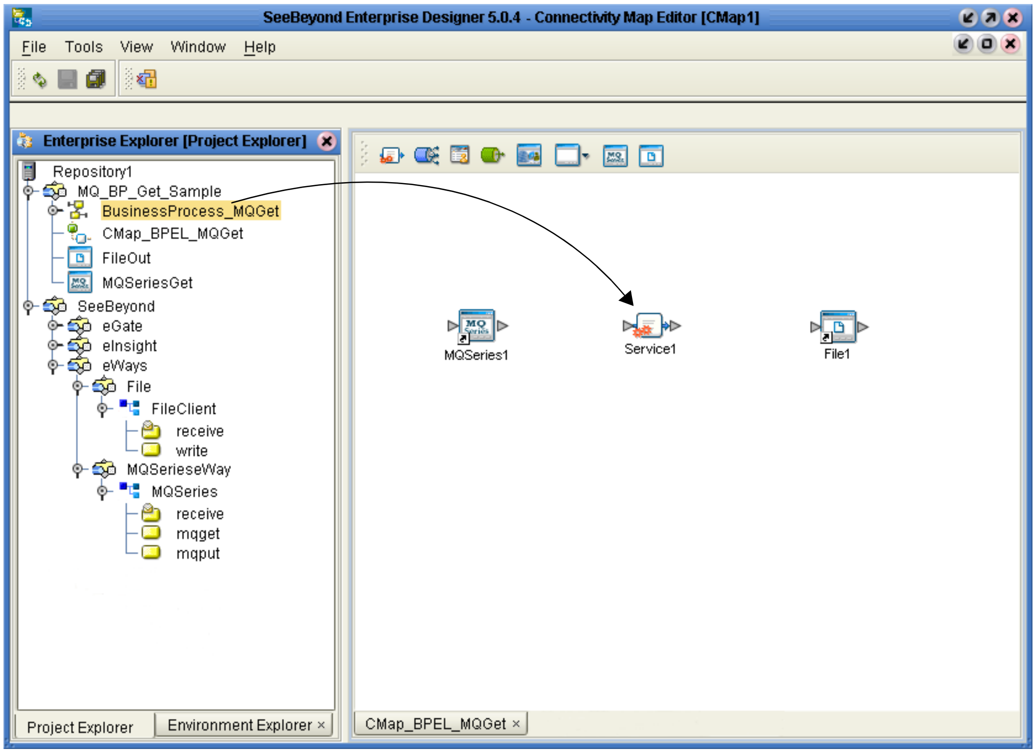
2. Rename the **MQSeries1** External Application to **MQSeriesGet** by right-clicking the object, selecting **Rename** from the shortcut menu, and typing in the new name.
3. Rename the **File1** External Application to **FileOut**.
4. Rename **Service1** to **BusinessProcess_MQGet1**.
5. Save your current 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 13 on page 43**).

Figure 13 Connectivity Map - Binding the Components

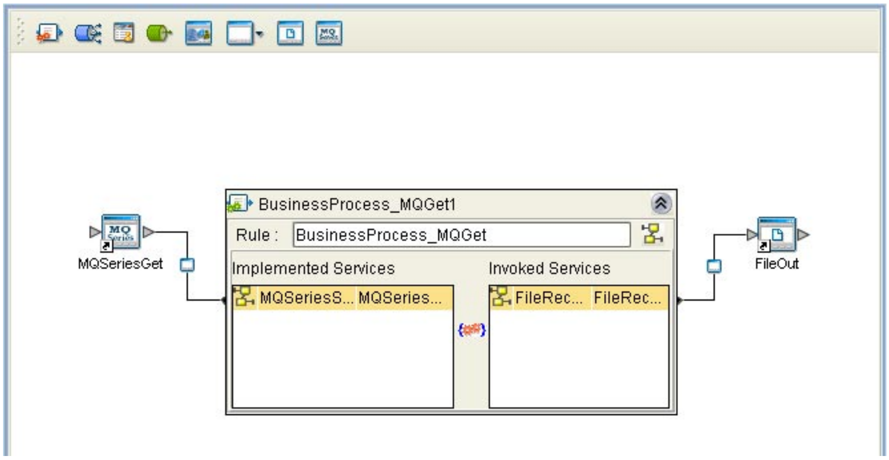


3 Double-click the **BusinessProcess_MQGet1** service. The **BusinessProcess_MQGet1** binding box appears with the **BusinessProcess_MQGet** Rule.

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 14).

Figure 14 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.

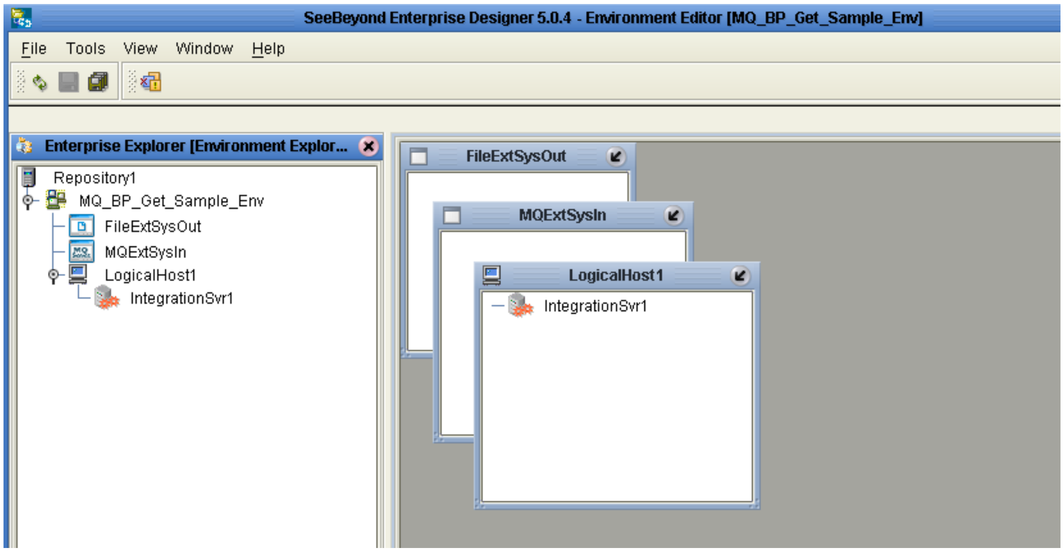
5 From the Project Explorer tree, right-click **MQ_BP_Get_Sample_Env** and select **New WebSphere MQ External System**. Name this External System **MQExtSysIn** and select **Inbound WebSphere MQ eWay** as the External System Type. The **MQExtSysIn** window is added to the Environment Editor.

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 15.

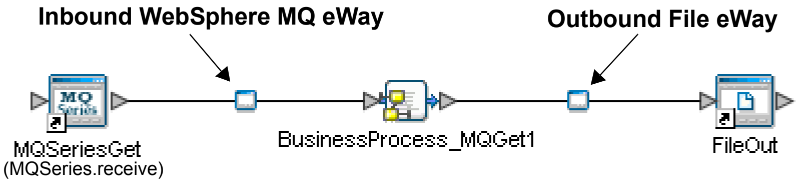
Figure 15 Environment Editor



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 16).

Figure 16 eWay Configurations



The File eWay properties are configured from the Connectivity Map. The WebSphere MQ 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 Editor opens to the Outbound File eWay properties. Modify the properties for your system, including the settings for the **Outbound File eWay** in Table 2, and click **OK**. The properties are saved for the eWay.

Table 2 Outbound File eWay Settings

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

Configuring the WebSphere MQ eWay Properties

The WebSphere MQ eWay properties must be set in both the Project Explorer and Environment Explorer. For more information on the WebSphere MQ eWay properties and the Properties Editor, see **Creating and Configuring the WebSphere MQ eWay** on page 18 or see the *eGate Integrator User's Guide*.

Modifying the WebSphere MQ eWay Connectivity Map Properties

1. From the Connectivity Map, double-click the WebSphere MQ eWay. The Properties Editor opens to the WebSphere MQ eWay Connectivity Map properties.
2. Modify the **WebSphere MQ** eWay properties for your system and click **OK**.

Modifying the WebSphere MQ eWay Environment Explorer Properties

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

2 Modify the WebSphere MQ eWay environment properties for your system, including the settings in Table 3, and click **OK**.

Table 3 WebSphere MQ 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**.

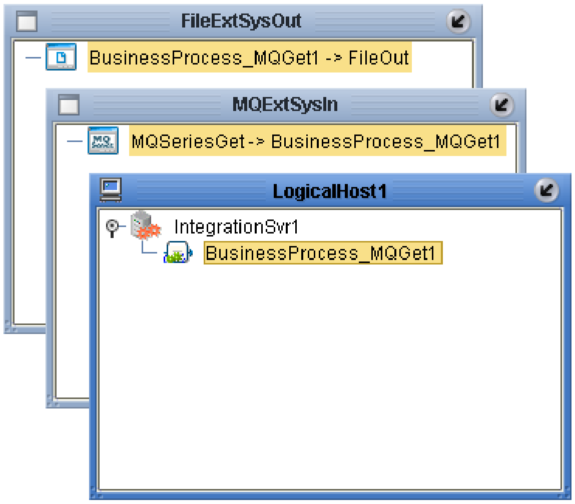
2 Enter a name for the Deployment Profile (for this sample **MQ_BP_Get_Sample_DP**). Make sure that the selected Environment is **MQ_BP_Get_Sample_Env.** Click **OK**.

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 17).

Figure 17 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*.

4.6 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 WebSphere MQ 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 36**.**

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.

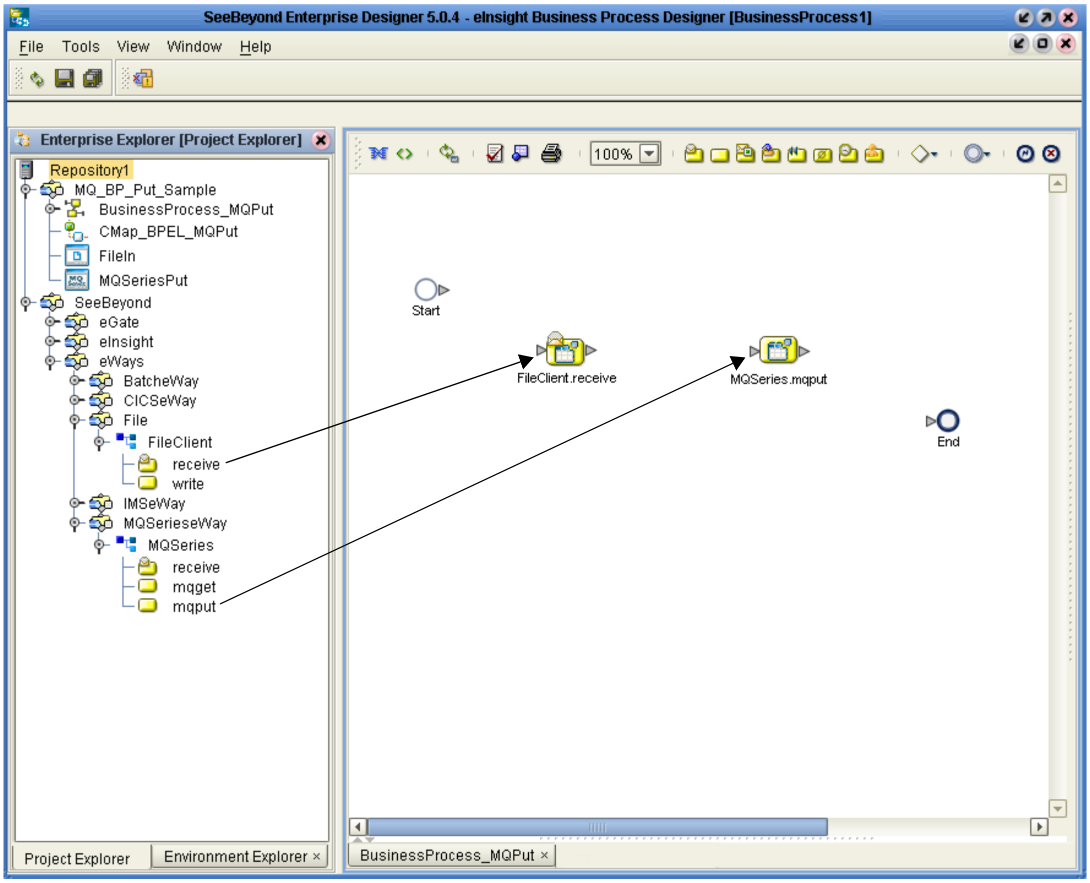
1. Start the Enterprise Designer.
2. From the Enterprise Explorer's Project Explorer tab, select your Repository on the Project Explorer tree, right-click the Repository and select **New Project**. A new project appears on the Project Explorer tree.
3. Click twice on **Project1** and rename the project (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 18 on page 49**:
   - **receive**, under SeeBeyond > eWays > File > FileClient
   - **mqput**, under SeeBeyond > eWays > MQSerieseWay > MQSeries

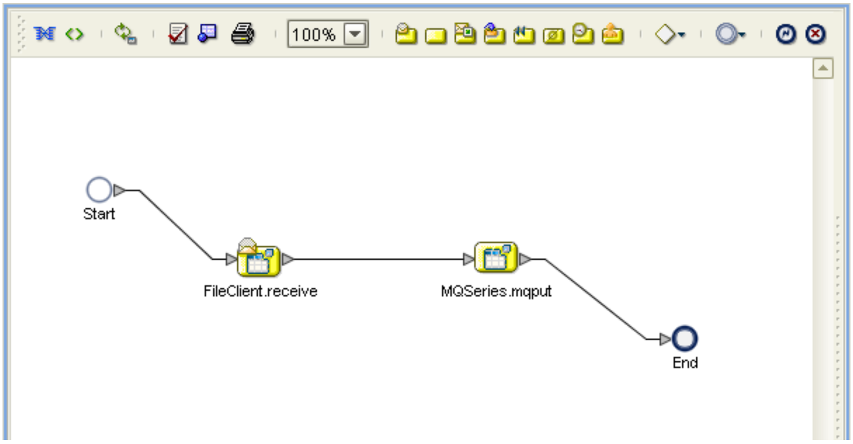
Figure 18 eInsight Business Process Designer - Populate the Canvas



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 19.

- Start -> FileClient.receive
- FileClient.receive -> MQSeries.mqput
- MQSeries.mqput -> End

Figure 19 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.

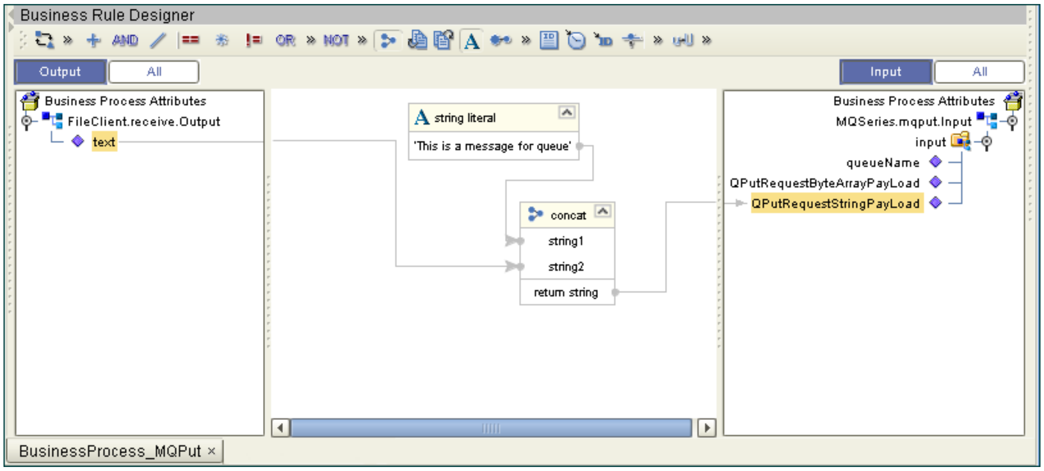
5 From the Business Rule Designer toolbar, drag and drop the **string literal** icon to the Business Rules Designer canvas. The **string literal** method box appears.

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.

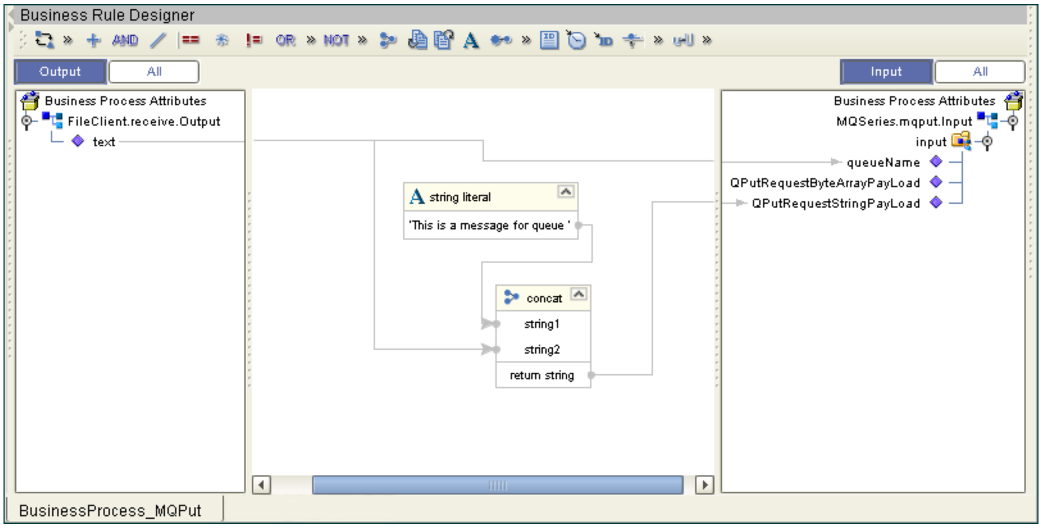
8 Map the **return string** output node of the **concat** method box, to **QPutRequestStringPayLoad** under MQSeries.mqput.Input > input in the Input pane of the Business Rule Designer (see Figure 20).

Figure 20 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 21).

Figure 21 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, **WebSphere MQ External Application** and **File External Application**). Icons representing the selected applications are added to the Connectivity Map toolbar.

Populate the Connectivity Map

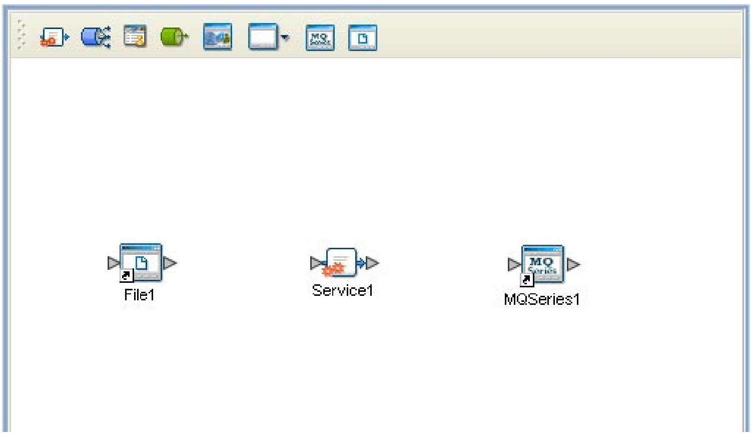
Add the project components to the Connectivity Map by dragging the icons from the toolbar to the canvas.

1 For the MQ_BP_Put_ Sample project, drag the following components onto the Connectivity Map canvas as displayed in Figure 22:

- **Service**
- **WebSphere MQ External System**

- **File External System**

Figure 22 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 WebSphere MQ External System**. Name the External System **MQExtSysOut** and select **Outbound WebSphere MQ 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 WebSphere MQ 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 Editor opens to the Inbound File eWay properties. Modify the properties for your system.

Configuring the WebSphere MQ eWay Properties

The WebSphere MQ eWay properties must be set in both the Project Explorer and Environment Explorer. For more information on the WebSphere MQ eWay properties and the Properties Editor, see **Creating and Configuring the WebSphere MQ eWay** on page 18 or see the *eGate Integrator User's Guide*.

Modifying the WebSphere MQ eWay Connectivity Map Properties

1. From the Connectivity Map, double-click the **WebSphere MQ eWay**. The Properties Editor opens to the WebSphere MQ eWay Connectivity Map properties.
2. Modify the WebSphere MQ eWay properties for your system and click **OK**.

Modifying the WebSphere MQ eWay Environment Explorer Properties

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

Chapter 5

Implementing a WebSphere MQ eWay Project

This chapter provides an introduction to the WebSphere MQ 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

5.1 WebSphere MQ eWay Components

This chapter presents a sample WebSphere MQ 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 WebSphere MQ eWay include the following:

WebSphere MQ eWay Properties file

The Properties file for the WebSphere MQ 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 WebSphere MQ eWay Properties file and the Properties dialog box see **Configuring the WebSphere eWay** on page 18.

MQSeries OTD

The MQSeries OTD is provided with the eWay and contains methods and attributes used to create the Business Rules that invoke the WebSphere MQ program. This OTD has been restructured for WebSphere MQ eWay version 5.0.4 and above. for more information on the latest version of the MQSeries OTD see **OTD Changes for WebSphere MQ eWay 5.0.4** on page 57.

5.2 OTD Changes for WebSphere MQ eWay 5.0.4

The **MQSeries OTD** for WebSphere MQ eWay 5.0.4 has been slightly restructured from the OTD structure in previous versions. These changes are reflected in the new 5.0.4 version of the sample projects as well as the 5.0.4 Javadoc. Existing projects will continue to work with the restructured OTD, but may require the following update.

Updating Project Collaborations that use the "receive" Web Service Operation

Any existing projects with Collaborations that use the MQSeries **receive** Web Service Operation must be updated for WebSphere MQ eWay 5.0.4. Internal references to the previous package still exist in the project, and even though the new eWay release uses the same package name, the following error may occur during project activation:

```
package com.stc.connector.appconn.mqseriesadapter does not exist
```

To update an existing Collaboration, do the following:

1. Locate the projects Collaboration icon in the Project Explorer tree. If the icon appears to have a red padlock, this means that the Collaboration must be checked out before it can be updated. To do this, right-click the Collaboration and select **Check Out** from the shortcut menu. The Collaboration can now be modified.
2. From the Project Explorer tree, right-click the Collaboration, and select **Properties** from the shortcut menu. The **Collaboration Definition Properties** dialog box appears.
3. From the **Operation** section of the Properties dialog box, click the **Input Message** field's ellipsis (. . .) button. The **Select** dialog box appears.
4. Nothing needs to be done in the **Select** dialog box. Simply click **Cancel** to close the dialog box.
5. Click **OK** to close the Properties dialog box.
6. Reactivate the project. The project is now associated with the current package.

5.3 WebSphere MQ eWay Sample Projects

The WebSphere MQ eWay includes two inbound and two outbound sample projects, that use Java Collaborations.

- **The MQ_JCE_Get_New_Sample Project** on page 59
- **The MQ_JCE_Get_Sample Project** on page 69
- **The MQ_JCE_Put_New_Sample Project** on page 70
- **The MQ_JCE_Put_Sample Project** on page 78

Both inbound and both outbound samples demonstrate the similar functions, but two of these, **MQ_JCE_Get_Sample_New** and **MQ_JCE_Put_Sample_New** reflect the restructured MQSeries OTD as well as the current Javadoc. This chapter provides a walk-through of the manual creation of these sample projects.

5.4 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** from the shortcut menu. The **Import Manager** appears.
3. Browse to the directory that contains the sample project zip file. Select the sample file (for this sample, **WebSphereMQSeries_JCE_Get_New_Sample.zip**) and click **Import**. After the sample project is successfully imported, click **Close**.
4. From the Project Explorer tree, rename the project to **MQ_JCE_Get_New_Sample.**
5. 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 64)
   - Configure the eWays properties for your specific system (see **Configuring the eWays** on page 65)
   - Create a **Deployment Profile** (see **Creating and Activating the Deployment Profile** on page 67)

5.5 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 the WebSphere MQ 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.

5.6 The MQ_JCE_Get_New_Sample Project

The WebSphere MQ eWay **MQ_JCE_Get_New_Sample** project demonstrates the following:

1. The WebSphere MQ 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_New_Sample** project.

5.6.1. 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. Rename **Project1** to **MQ_JCE_Get_New_Sample**.

5.6.2 Create a Connectivity Map

The Connectivity Map provides a canvas for assembling and configuring a project’s components.

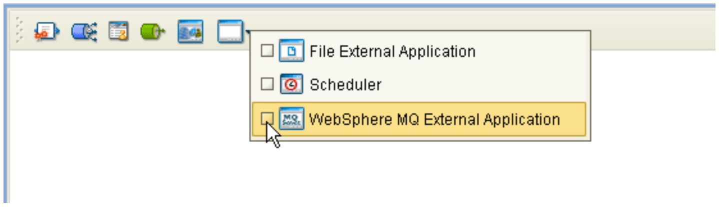
1. In Enterprise Explorer’s Project Explorer, right-click the **MQ_JCE_Get_New_Sample** project and select **New > Connectivity Map** from the shortcut menu.
2. The New Connectivity Map appears and a node is added under the project on the Project Explorer tree labeled **CMap1**. From the Project Explorer 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, you must first select WebSphere MQ as an External System to use in your Connectivity Map (see Figure 23).

Figure 23 Connectivity Map - External Applications


□ File External Application
□ Scheduler
□ WebSphere MQ External Application

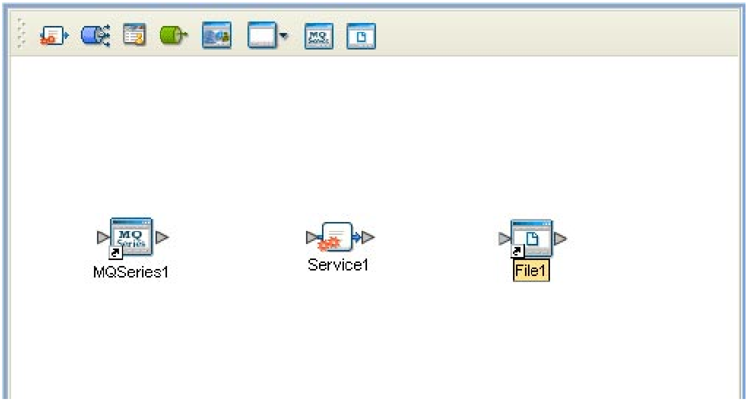
1. Click the **External Application** icon on the Connectivity Map toolbar,
2. Select the external systems needed for your project (for this sample, the **Scheduler**, and **WebSphere MQ**, and **File** External Applications). Icons representing the selected external systems are added to the Connectivity 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 24 on page 60**:
   - **WebSphere MQ External System**
   - **Service** (A service is a container for Collaborations, Business Processes, eTL processes, and so forth)
   - **File External System**

Figure 24 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.

5.6.3. Creating the Collaboration Definition

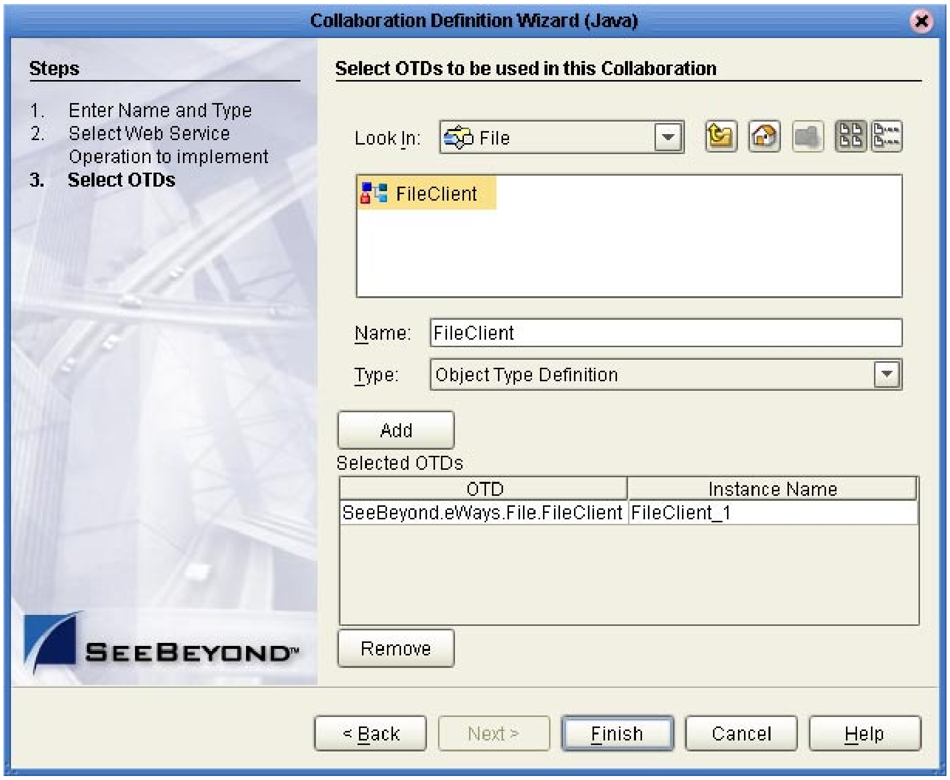
The next step in the sample is to create a Collaboration using the **Collaboration Definition Wizard (Java)**. Once the Collaboration Definition has been created, the Business Rules of the Collaboration are written using the Collaboration Editor.

Creating the jcd_MQGet Collaboration

The jcd_MQGet 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_New_Sample** project and select **New** > **Collaboration Definition (Java)** from the shortcut menu. The **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 25 on page 61**).

Figure 25 Collaboration Definition Wizard (Java) - Select Web Service Interface


Collaboration Definition Wizard (Java)

Steps

1. Enter Name and Type
2. Select Web Service Operation to implement
3. **Select OTDs**

Select OTDs to be used in this Collaboration

Look In: File

FileClient

Name: FileClient

Type: Object Type Definition

Selected OTDs

OTD	Instance Name
SeeBeyond.eWays.File.FileClient	FileClient_1

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

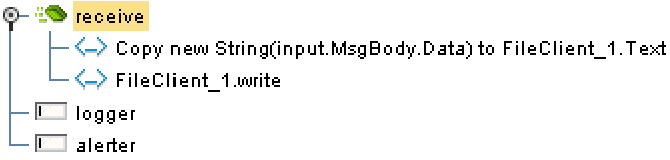
5.6.4. Creating the Business Rules

The Business Rules for the jcd_MQGet Collaboration are created using the Collaboration Editor (Java). To open the Collaboration Editor to the jcd_MQGet Collaboration, from the Project Explorer tree, double-click the **jcd_MQGet** Collaboration. The Collaboration Editor opens, to the selected Collaboration.

The jcd_MQGet Collaboration Business Rule

The jcd_MQGet Collaboration contains the Business Rules displayed in Figure 26.

Figure 26 jcd_MQGet Collaboration Business Rules

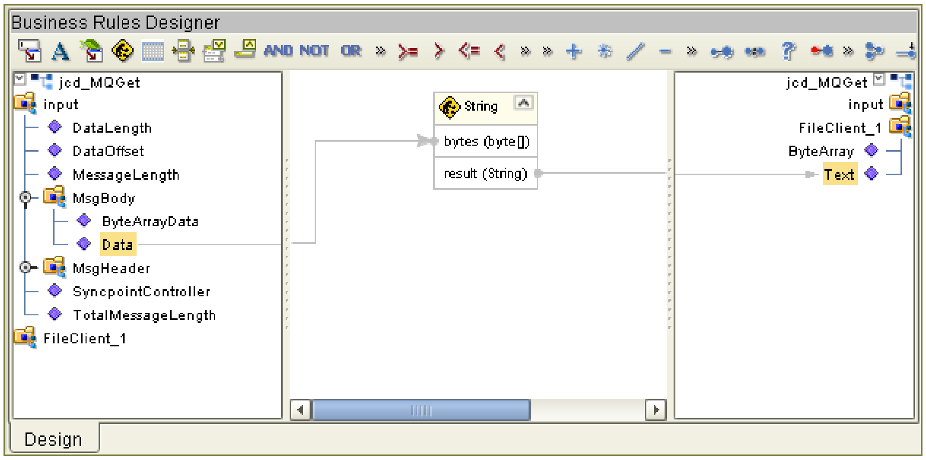


```
receive
  Copy new String(input.MsgBody.Data) to FileClient_1.Text
  FileClient_1.write
logger
alerter
```

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 String(input.MsgBody.Data) to FileClient_1.Text** Business Rule do the following:
   - A From the Business Rule Designer toolbar, click the **Call New Constructor** icon. The **Call Constructor** dialog box appears. Select **String** from the **All Classes** box, and **String(byte[] bytes)** from the **Constructors** box. The **Package** value defaults to **java.lang**. Click **OK**. The **String** method box appears in the Business Rule Designer canvas.
   - B Map **Data** under **input > MsgBody** in the left pane of the Business Rule Designer, to the **bytes (byte[])** input node of the **String** method box. This is done by clicking on **Data** and dragging the cursor to the **bytes (byte[])** input node.
   - C Map the result (String) of the **String** method box, to **Text** under **FileClient_1** in the right pane of the Business Rule Designer. (see Figure 27).

Figure 27 Collaboration Editor - Business Rules Designer



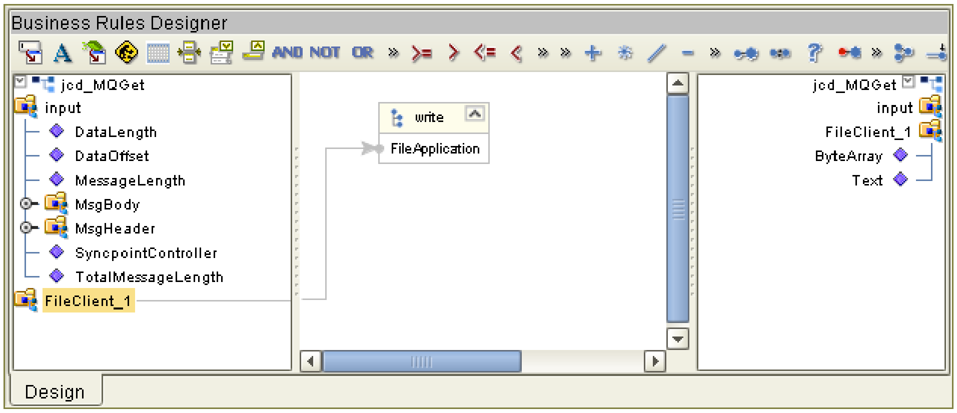
3 To create the **FileClient_1.write** Business Rule do the following:

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 28).

Figure 28 FileClient_1.write Business Rule



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

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

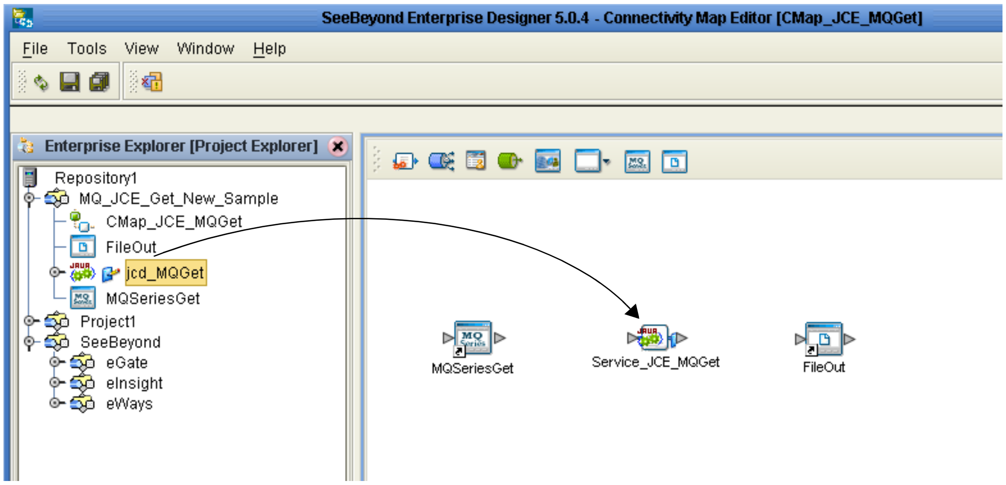
5.6.5. Binding the eWay Components

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 29 on page 64**).

Figure 29 Connectivity Map - Binding the Collaborations

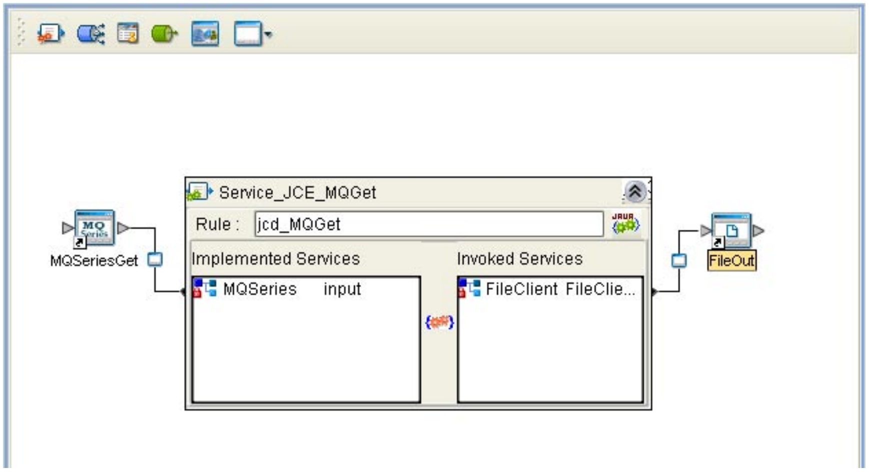


3 Double-click **Service_JCE_MQGet**. The **Service_JCE_MQGet** Binding dialog box appears using **jcd_MQGet** as the rule.

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 30).

Figure 30 Connectivity Map - Binding the Components



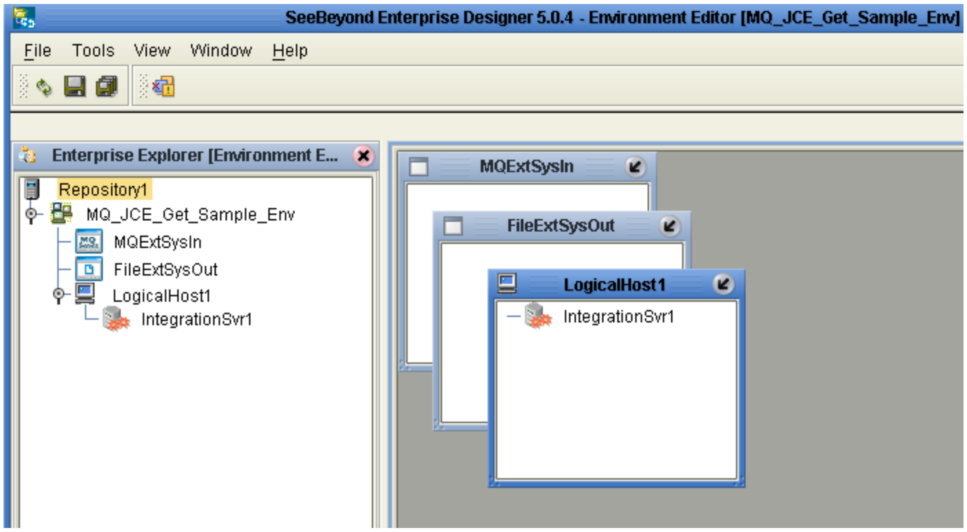
6 Click the chevrons in the upper-right corner of the **Service_JCE_MQGet** Binding dialog box to minimize the **Service_JCE_MQGet**.

5.6.6. 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 WebSphere MQ External System**. Name the External System **MQExtSysIn** and select **Inbound WebSphere MQ 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 31).

Figure 31 Environment Editor

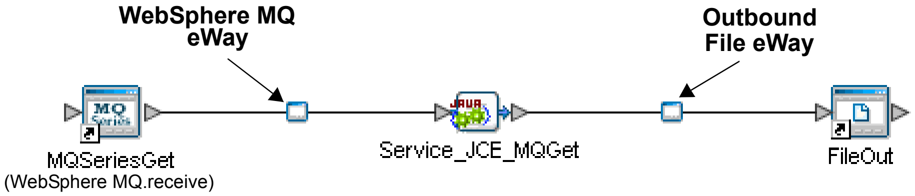


8. Save your current changes to the Repository.

5.6.7. 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 32 on page 66**). eWays facilitate communication and movement of data between the External Applications and the eGate system.

Figure 32 eWays



The File eWay properties and the Scheduler parameters are configured from the Connectivity Map. The WebSphere MQ 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 Editor opens to the Outbound File eWay configuration.
2. Modify the parameters for your system, including the settings for the Outbound File eWay in Table 4, and click **OK**.

Table 4 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 eWay Properties

The WebSphere MQ eWay parameters must be set in both the Project Explorer and Environment Explorer. For more information on the WebSphere MQ eWay properties and the Properties Editor, see **Configuring the WebSphere eWay** on page 18, or see the *eGate Integrator User's Guide*.

Modifying the WebSphere MQ eWay Connectivity Map Properties

1. From the Connectivity Map, double-click the **MQSeriesGet** eWay. The Templates dialog box appears. Select **Inbound WebSphere MQ eWay** and click **OK**. The Properties Editor opens to the Inbound WebSphere MQ eWay properties.
2. Modify the **MQSeriesGet** eWay Connectivity Map properties for your system, including the settings in **Table 5 on page 66**, and click **OK**.

Table 5 WebSphere MQ eWay Connectivity Map Properties

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

Inbound WebSphere MQ eWay Connectivity Map Properties	
Schedule Interval	Polling Interval in milliseconds

Modifying the WebSphere MQ eWay Environment Explorer Properties

1. From the **Environment Explorer** tree, right-click the WebSphere MQ External System (**MQExtSysIn** in this sample), and select **Properties**. The Properties dialog box opens to the WebSphere MQ eWay Environment properties.
2. Modify the WebSphere MQ eWay Environment properties for your system, including the settings in **Table 6 on page 67**, and click **OK**.

Table 6 WebSphere MQ eWay Environment Explorer Properties

WebSphere MQ 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*

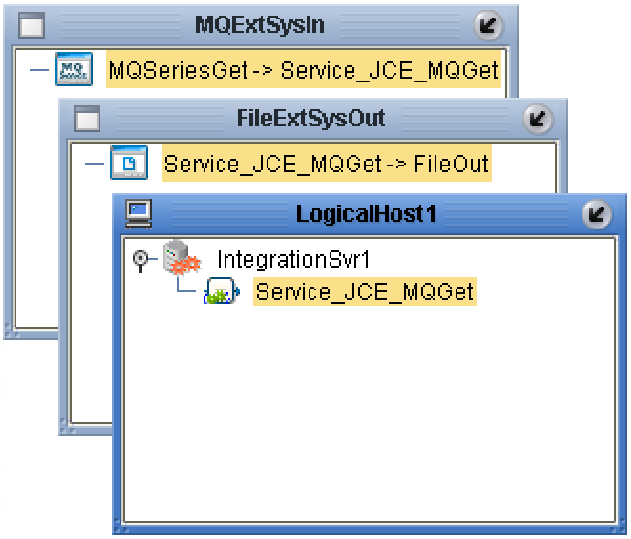
3. Save your changes to the Repository when finished.

5.6.8 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_New_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 33 on page 68**).

Figure 33 Deployment Profile



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, 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*.

5.7 The MQ_JCE_Get_Sample Project

The **MQ_JCE_Get_Sample** project performs the same functions as the **MQ_JCE_Get_New_Sample**, but reflects the MQSeries OTD from the previous version of the WebSphere MQ eWay. For this reason, the sample must be updated before it will run with the current WebSphere MQ eWay. For directions on updating the project, see **Updating Project Collaborations that use the “receive” Web Service Operation** on page 57.

5.8 The MQ_JCE_Put_New_Sample Project

The **MQ_JCE_Put_New_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 WebSphere MQ eWay to an external directory.

The MQ_BP_Put_New_Sample is found in the Samples file as **WebSphereMQSeries_JCE_Put_New_Sample.zip.** To Import the project, see **Importing a Sample Project** on page 58**.**

The following pages provide step by step directions for manually creating the **MQ_JCE_Put_New_Sample** project components.

5.8.1 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_New_Sample**).

5.8.2 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_New_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

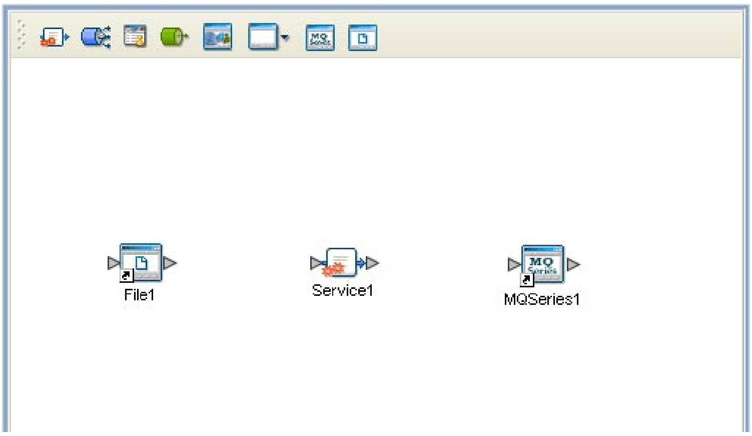
1. Click the **External Application** icon on the Connectivity Map toolbar,
2. Select the applications needed for your project (for this sample, **WebSphere MQ External Application** and **File External Application**). Icons representing the selected applications are added to the Connectivity Map toolbar.

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_New_Sample** project, drag the following components onto the Connectivity Map canvas as displayed in Figure 34:
   - **Service**
   - **WebSphere MQ External System**
   - **File External System**

Figure 34 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

5.8.3. Creating the Collaboration Definition

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

Creating the jcd_MQPut Collaboration

The jcd_MQPut 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_New_Sample** project and select **New** > **Collaboration Definition (Java)** from the shortcut menu. The **Collaboration Definition Wizard (Java)** 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 Collaboration Editor with the new **jcd_MQPut** Collaboration appears in the right pane of the Enterprise Designer.

5.8.4. Creating the Business Rules

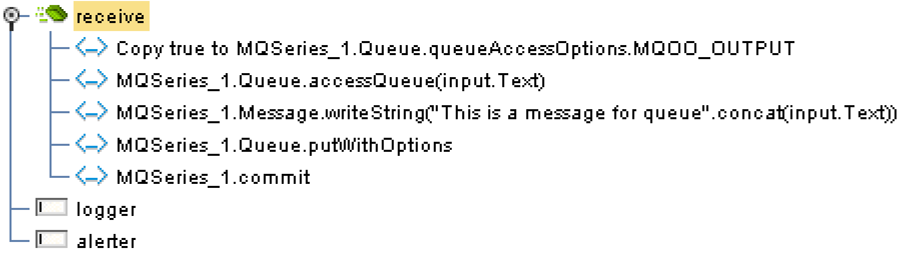
The Business Rules for the jcd_MQPut Collaboration are created using the Collaboration Editor (Java). To open the Collaboration Editor to the jcd_MQPut Collaboration, from the Project Explorer tree, double-click the **jcd_MQPut** Collaboration. The Collaboration Editor opens, to the selected Collaboration.

The jcd_MQPut Collaboration Business Rule

The the **WebSphereMQSeries_JCE_Put_Sample** project uses one Collaboration created in the previous section, **jcd_MQPut**.

The jcd_MQPut Collaboration contains the Business Rules displayed in Figure 35

Figure 35 jcd_MQPut Collaboration Business Rules



```
receive
    Copy true to MQSeries_1.Queue.queueAccessOptions.MQOO_OUTPUT
    MQSeries_1.Queue.accessQueue(input.Text)
    MQSeries_1.Message.writeString("This is a message for queue".concat(input.Text))
    MQSeries_1.Queue.putWithOptions
    MQSeries_1.commit
logger
alerter
```

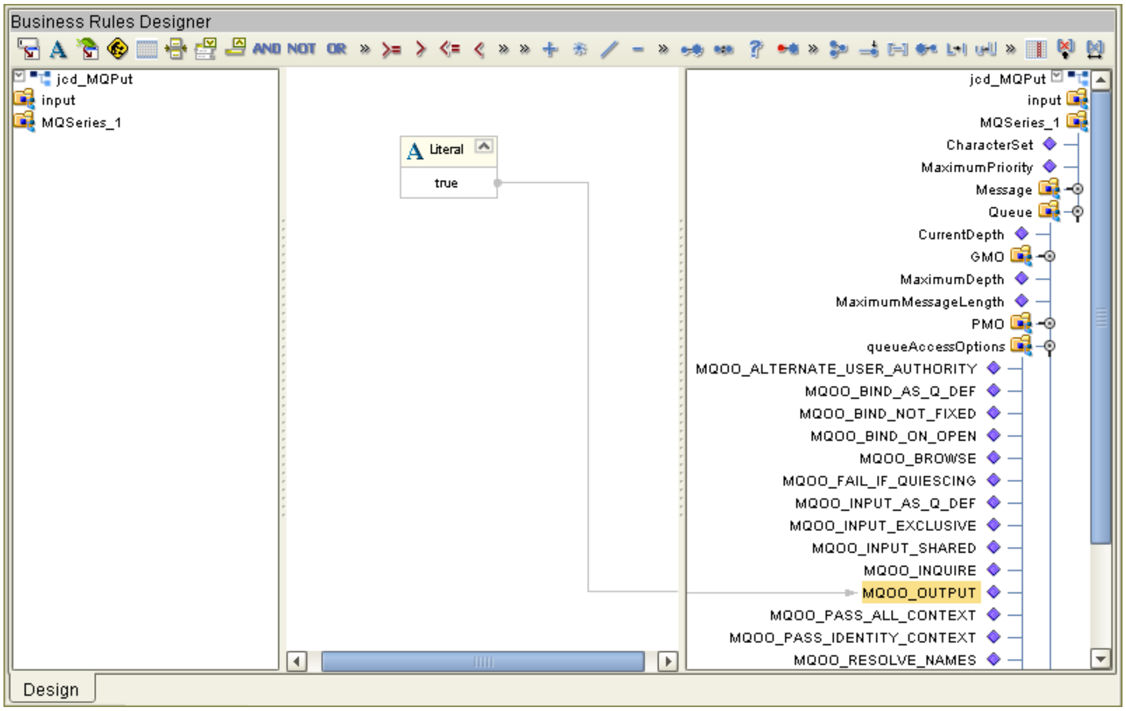
To create the **jcd_MQPut** Collaboration Business Rules do the following:

1. Double-click **input** and **MQSeries_1** in the Business Rule Designer panes to expand the nodes.
2. To create the **Copy true to MQSeries_1.Queue.queueAccessOptions. 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 > Queue > 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 36 on page 73**).

Figure 36 Copy true to MQOO_OUTPUT Business Rule



3 To create the **MQSeries_1.Queue.accessQueue(input.Text)** Business Rule do the following:

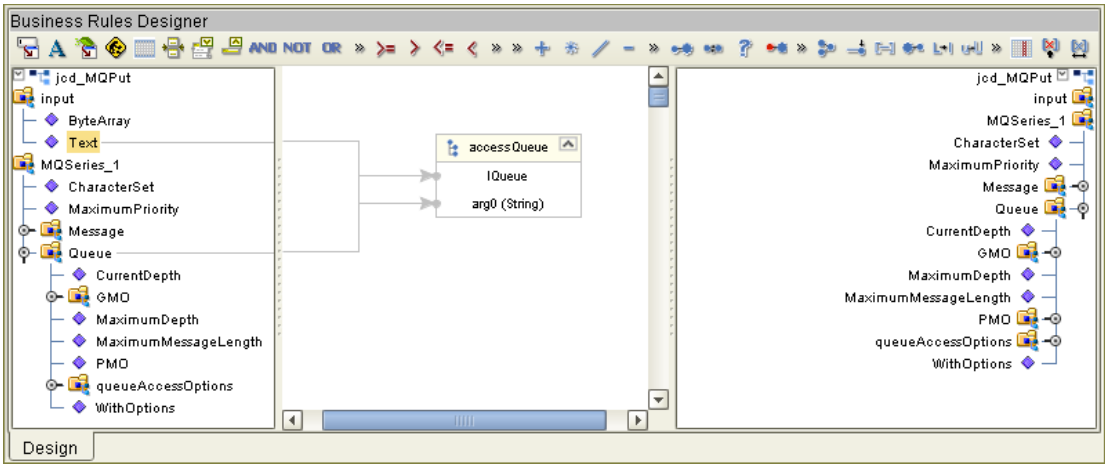
A From the Business Rules toolbar click the **rule** button to add a new rule.

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

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

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 37).

Figure 37 Queue.accessQueue(Text) Business Rule



4 To create the **MQSeries_1.Message.writeString("This is a message for queue".concat(input.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**.

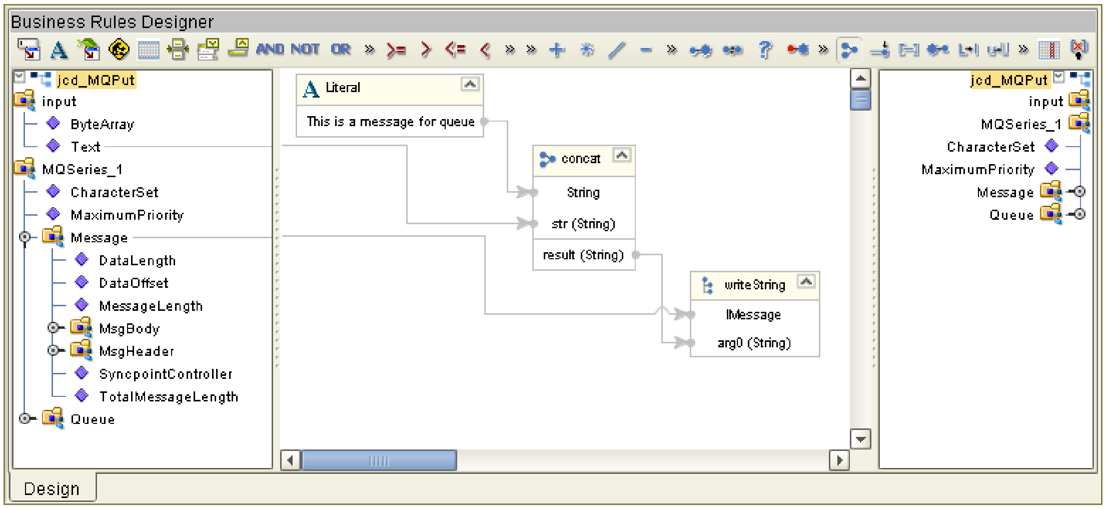
E Select **writeString(String arg0)** from the method selection menu. The **writeString** method box appears.

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 38).

Figure 38 Message.writeString("This is a message for queue ".concat(input.Text)) Rule



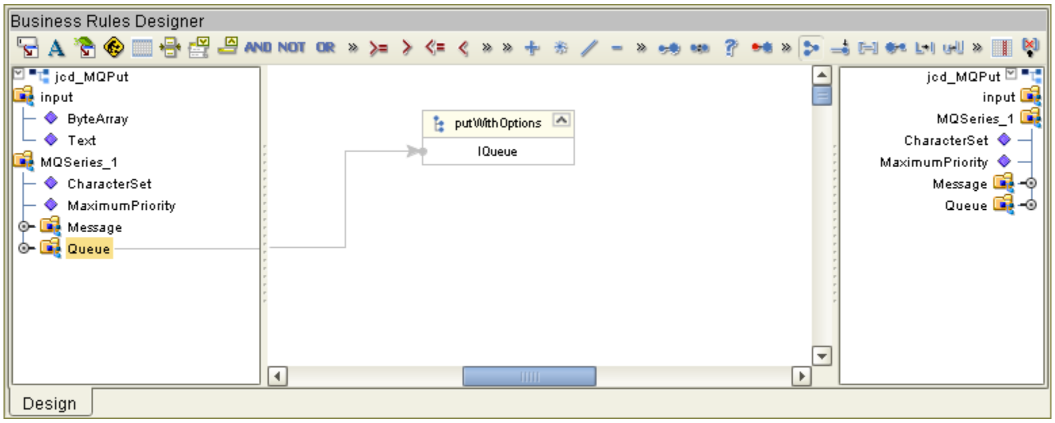
5 To create the **MQSeries_1.Queue.putWithOptions** Business Rule do the following:

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**.

C Select **putWithOptions()** from the method selection menu. The **putWithOptions** method box appears on the Business Rules Designer canvas (see **Figure 39 on page 75**).

Figure 39 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**.

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 Collaboration Editor see the *eGate Integrator User's Guide*.

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

5.8.5. 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.

5.8.6. 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 WebSphere MQ External System**. Name the External System **MQExtSysOut** and select **Outbound WebSphere MQ 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.

5.8.7. Configuring the eWay Properties

The MQ_JCE_Put_New_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 WebSphere MQ 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 Editor opens to the Inbound File eWay properties. Modify the properties for your system.

Configuring the WebSphere MQ eWay Properties

The WebSphere MQ eWay properties must be set in both the Project Explorer and Environment Explorer. For more information on the WebSphere MQ eWay properties and the Properties Editor, see **Creating and Configuring the WebSphere MQ eWay** on page 18 or see the *eGate Integrator User's Guide*.

Modifying the WebSphere MQ eWay Connectivity Map Properties

1 From the Connectivity Map, double-click the **WebSphere MQ** eWay. The Properties Editor opens to the WebSphere MQ eWay Connectivity Map properties.

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

Modifying the WebSphere MQ eWay Environment Explorer Properties

1 From the **Environment Explorer** tree, right-click the Inbound WebSphere MQ External System, and select **Properties**. The Properties Editor opens to the WebSphere MQ eWay environment properties.

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

5.8.8 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_New_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.

5.8.9. 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.

5.9 The MQ_JCE_Put_Sample Project

The **MQ_JCE_PUT_Sample** project performs the same functions as the **MQ_JCE_Put_New_Sample**, but reflects the MQSeries OTD from the previous version of the WebSphere MQ eWay. For this reason, the sample must be updated before it will run with the current WebSphere MQ eWay. For directions on updating the project, see **Updating Project Collaborations that use the “receive” Web Service Operation** on page 57.

Chapter 6

Java Classes and Methods for the WebSphere MQ eWay

6.1 WebSphere MQ eWay Classes and Methods

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

- IGMO Class
- IMessage Class (contains the IMessage.msgBody and Imessage.MsgHeader interfaces)
- IMgOtd Class
- IPMO Class
- IQueue Class
- ISyncpointControl

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.

Appendix A

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 Collaboration Editor (Java). 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 Collaboration Editor (Java) sets the header properties by calling **readProperty()** or **writeProperty()**.

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

Index