

*SeeBeyond ICAN Suite*

# MSMQ eWay Intelligent Adapter User's Guide

*Release 5.0*



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.

SeeBeyond, e\*Gate, e\*Way, and e\*Xchange are the registered trademarks of SeeBeyond Technology Corporation in the United States and/or select foreign countries. The SeeBeyond logo, SeeBeyond Integrated Composite Application Network Suite, eGate, eWay, eInsight, eVision, eXchange, eView, eIndex, eTL, ePortal, eBAM, and e\*Insight are trademarks of SeeBeyond Technology Corporation. The absence of a trademark from this list does not constitute a waiver of SeeBeyond Technology Corporation's intellectual property rights concerning that trademark. This document may contain references to other company, brand, and product names. These company, brand, and product names are used herein for identification purposes only and may be the trademarks of their respective owners.

© 2005 SeeBeyond Technology Corporation. All Rights Reserved.

**This work is confidential and proprietary information of SeeBeyond and must be maintained in strict confidence.**

Version 20050602164916.

# Contents

---

## Chapter 1

<b>Introduction</b>	<b>7</b>
About Microsoft™ Message Queuing (MSMQ)	7
About the MSMQ eWay Intelligent Adapter	8
What's New in This Release	8
About This Document	8
Organization of Information	8
MSMQ Javadoc	9
Scope of the Document	9
Intended Audience	9
Document Conventions	9
SeeBeyond Web Site	10
SeeBeyond Documentation Feedback	10
Related Documents	10

---

## Chapter 2

<b>Installing the MSMQ eWay</b>	<b>11</b>
Supported Operating Systems	11
System Requirements	11
Supported External Applications	12
Installing the MSMQ eWay	12
Installing the MSMQ eWay	12
Adding the eWay to an Existing ICAN Suite Installation	13
Downloading and Adding the MSMQ DLLs and JNI files	14
Uploading a JAR file to the Logical Host	14
Adding the DLL files to the Environment Path	15
After Installation	15

---

Chapter 3

<b>Configuring the MSMQ eWay</b>	<b>16</b>
<b>Creating and Configuring the MSMQ eWay</b>	<b>16</b>
Selecting MSMQ as the External Application	16
Modifying the MSMQ eWay Properties	17
Using the Properties Editor	18
<b>MSMQ eWay Properties</b>	<b>19</b>
<b>Inbound MSMQ eWay Connectivity Map Properties</b>	<b>20</b>
MSMQ Access Mode	20
<i>Identifying an MSMQ Queue</i>	20
MSMQ Format Name	21
MSMQ Queue Alias	21
MSMQ Queue Name	21
MSMQ Receive Action Code	22
MSMQ Receive Interval	22
MSMQ Share Mode	22
<b>Outbound MSMQ eWay Connectivity Map Properties</b>	<b>23</b>
Message Priority	23
MSMQ Access Mode	24
<i>Identifying an MSMQ Queue</i>	24
MSMQ Format Name	24
MSMQ Queue Alias	25
MSMQ Queue Name	25
MSMQ Receive Action Code	25
MSMQ Share Mode	26
MSMQ Transaction Type	26
<b>Inbound MSMQ eWay Environment Properties</b>	<b>27</b>
MSMQ Host Name	27
<b>Outbound MSMQ eWay Environment Properties</b>	<b>28</b>
MSMQ Connection Mode	28
MSMQ Host Name	29
<b>Enabling Rollback When an MSMQ Message Fails</b>	<b>30</b>
<b>Manually Configuring the eWay</b>	<b>31</b>
<b>Alerting and Logging</b>	<b>32</b>

---

Chapter 4

<b>Using the MSMQ eWay With eInsight</b>	<b>33</b>
eInsight Engine and Components	33
The MSMQ eWay With eInsight	34
MSMQ eWay Sample Project for eInsight	34

<b>Importing a Sample Project</b>	<b>35</b>
<b>Creating the MSMQ_BP_Sample Project</b>	<b>35</b>
Creating a Project	35
Creating OTDs Using the DTD Wizard	36
Creating the Business Processes	37
Creating the bpGet Business Process	37
<b>Creating the Business Rules Using the Business Rules Designer</b>	<b>39</b>
Creating the bpGet Business Process Business Rules	39
Creating the bpPut Business Process Business Rules	43
Creating the bpReceive Business Process Business Rules	46
<b>Creating the Connectivity Maps</b>	<b>47</b>
<b>Binding the Project Components</b>	<b>49</b>
Binding the components of the cmPut Connectivity Map	51
Binding the components of the cmPut Connectivity Map	51
<b>Creating an Environment</b>	<b>51</b>
<b>Configuring the eWays</b>	<b>52</b>
Configure the eWay properties for your system, including the settings for the input and output directories. For the most part, the eWay settings can be left with their default settings. See <a href="#">“Configuring the MSMQ eWay” on page 16</a> for more information on configuring the eWays for your system.	52
<b>Creating and Activating the Deployment Profile</b>	<b>53</b>
<b>Running a Project</b>	<b>55</b>

---

## Chapter 5

<b>Implementing an MSMQ eWay Project</b>	<b>56</b>
<b>MSMQ eWay Components</b>	<b>56</b>
<b>MSMQ eWay Sample Project</b>	<b>57</b>
<b>Importing a Sample Project</b>	<b>57</b>
<b>Creating the MSMQ_JCD_Sample Project</b>	<b>58</b>
Creating a New Project	58
Creating the Connectivity Maps	58
Creating OTDs Using the DTD Wizard	60
<b>Creating the Collaboration Definitions</b>	<b>62</b>
Creating the jcdGet Collaboration	62
Creating the jcdPut Collaboration	63
Creating the jcdReceive Collaboration	63
<b>Creating the Business Rules Using the Collaboration Editor</b>	<b>64</b>
Creating the jcdGet Collaboration Business Rules	64
Creating the jcdPut Collaboration Business Rules	70
Creating the jcdReceive Collaboration Business Rules	73
<b>Binding the eWay Components</b>	<b>77</b>
Binding the components of the cmGet Connectivity Map	77
Binding the components of the cmPut Connectivity Map	78
Binding the components of the cmPut Connectivity Map	78
<b>Creating an Environment</b>	<b>79</b>
<b>Configuring the eWays</b>	<b>80</b>
<b>Creating and Activating the Deployment Profile</b>	<b>80</b>

**Contents**

**Running a Project** 82

**Index** 83

# Introduction

Welcome to the MSMQ™ eWay Intelligent Adapter User's Guide. This document describes how to install, configure, and implement the MSMQ eWay Intelligent Adapter in a typical SeeBeyond® Integrated Composite Application Network Suite™ (ICAN) environment.

## What's in This Chapter

- [About Microsoft™ Message Queuing \(MSMQ\)](#) on page 7
- [About the MSMQ eWay Intelligent Adapter](#) on page 8
- [About This Document](#) on page 8
- [SeeBeyond Web Site](#) on page 10

---

## 1.1 About Microsoft™ Message Queuing (MSMQ)

Microsoft Message Queuing, or MSMQ, is message queuing communications “middleware” that enables applications on diverse systems to communicate with each other. MSMQ allows applications to communicate through messages, and can guarantee the delivery of messages even in the event of system or network failures.

By using messages, MSMQ can provide asynchronous communication between applications running on the same computer or on separate computers connected through a network. When an application receives a message, it processes the message by reading the message content and acting accordingly. If appropriate, the receiving application can send a response message back to the sending application. MSMQ protects these transactions by safely storing the messages in queues, which prevents message loss and make messages available to an application whenever it is prepared to receive them.

Applications using MSMQ are able to send and process messages, even when the receiving application is not available. Upon connection, the receiver can accept or pick up the messages. Applications using MSMQ's transactional delivery mode can be assured that messages are delivered exactly one time only, and that they are delivered in the order that they were sent.

---

## 1.2 About the MSMQ eWay Intelligent Adapter

The SeeBeyond MSMQ eWay Intelligent Adapter (referred to as the MSMQ eWay throughout this document) allows the ICAN system to exchange data with Microsoft's MSMQ. The MSMQ eWay transparently integrates existing systems with MSMQ.

The MSMQ eWay enables ICAN to use business logic within Collaborations or Business Processes to perform 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 ICAN, permitting applications to run independently of one another at different speeds and times. This document explains how to install and configure the MSMQ eWay.

---

## 1.3 What's New in This Release

Version 5.0 is a new release of the MSMQ eWay Intelligent Adapter for SeeBeyond's ICAN suite. MSMQ eWay 5.0 supports MSMQ 3.0.

---

## 1.4 About This Document

This section provides a brief outline of this user's guide.

### 1.4.1 Organization of Information

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

- **Chapter 1 "Introduction"** provides an overview of the MSMQ eWay.
- **Chapter 2 "Installing the MSMQ eWay"** describes how to install the MSMQ eWay and lists the supported operating systems and system requirements.
- **Chapter 3 "Configuring the MSMQ eWay"** describes the process of configuring the MSMQ eWay to run in your environment.
- **Chapter 4 "Using the MSMQ eWay With eInsight"** describes the features and functionality of the MSMQ eWay using the eInsight Business Process Manager and the eInsight Web Services interface.
- **Chapter 5 "Implementing an MSMQ eWay Project"** describes the features and functionality of the MSMQ eWay using the eGate Integrator and the Collaboration Editor (Java).



## MSMQ Javadoc

An MSMQ eWay Javadoc is also provided, which documents the available Java methods provided with the MSMQ eWay. The Javadoc is uploaded with the eWay's documentation file (MSMQeWayDocs.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.

### 1.4.2 Scope of the Document

This user's guide provides a description of the MSMQ eWay Intelligent Adapter. It includes directions for installing the eWay, configuring the eWay properties, and implementing the eWay's sample Projects. This document is also intended as a reference guide, listing available properties, functions, and considerations. For a reference of available MSMQ eWay Java methods, see the associated Javadoc.

### 1.4.3 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, and/or HP NonStop Server), and must be thoroughly familiar with Windows-style GUI operations.

### 1.4.4 Document Conventions

The following conventions are observed throughout this document.

**Table 1** Document Conventions

Text	Convention	Example
Button, file, icon, parameter, variable, method, menu, and object names.	<b>Bold</b> text	<ul style="list-style-type: none"> <li>▪ Click <b>OK</b> to save and close.</li> <li>▪ From the <b>File</b> menu, select <b>Exit</b>.</li> <li>▪ Select the <b>logicalhost.exe</b> file.</li> <li>▪ Enter the <b>timeout</b> value.</li> <li>▪ Use the <b>getClassname()</b> method.</li> <li>▪ Configure the <b>Inbound</b> File eWay.</li> </ul>
Command line arguments and code samples	Fixed font. Variables are shown in <b><i>bold italic</i></b> .	bootstrap -p <b><i>password</i></b>
Hypertext links	<b>Blue</b> text	<a href="http://www.seebeyond.com">http://www.seebeyond.com</a>

---

## 1.5 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>

---

## 1.6 SeeBeyond Documentation Feedback

We appreciate your feedback. Please send any comments or suggestions regarding this document to:

[docfeedback@seebeyond.com](mailto:docfeedback@seebeyond.com)

---

## 1.7 Related Documents

The following SeeBeyond documents provide additional information about the ICAN product suite:

- *eGate Integrator User's Guide*
- *SeeBeyond ICAN Suite Installation Guide*
- *eInsight Business Process Manager User's Guide*

# Installing the MSMQ eWay

This chapter describes the process of installing the MSMQ eWay and the accompanying sample Projects.

### What's in This Document

- [Supported Operating Systems](#) on page 11
- [System Requirements](#) on page 11
- [Installing the MSMQ eWay](#) on page 12
- [Uploading a JAR file to the Logical Host](#) on page 14
- [Adding the DLL files to the Environment Path](#) on page 15
- [After Installation](#) on page 15

---

## 2.1 Supported Operating Systems

The MSMQ eWay is available for the following operating systems:

- Windows XP and Windows Server 2003

---

## 2.2 System Requirements

The system requirements for the MSMQ eWay are the same as those for ICAN. For more information, refer to the *SeeBeyond ICAN Suite Installation Guide*.

Although the MSMQ 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.

For additional system requirements, see the **MSMQ eWay Readme**, accessed from the Enterprise Manager's **Documentation** tab.

---

## 2.3 Supported External Applications

The MSMQ eWay supports the following external systems:

- Microsoft MSMQ version 3.0
- MSMQ must be installed on the computer where the Logical Host is running. MSMQ data can be accessed from a remote computer, but the eWay requires several dependency library files from MSMQ.

---

## 2.4 Installing the MSMQ 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.

The MSMQ eWay also requires additional runtime DLL files and a JNI file that are included with the installation. For directions on how to install these additional files see [“Downloading and Adding the MSMQ DLLs and JNI files” on page 14](#)

### 2.4.1 Installing the MSMQ eWay

The ICAN Suite installation process includes the following operations:

- Install the eGate Repository
- Upload products to the Repository
- Download components (including the eGate Enterprise Designer and Logical Host)

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

- 1 From the Enterprise Manager’s **ADMIN** tab, browse to the **Add-ons** directory and select the **ProductsManifest.xml**, and click **Submit**. The available Add-on product list is now displayed.
- 2 Browse to and select the following files located in the **Add-ons** directory:
  - ♦ **FileeWay.sar** (to install the File eWay, used in the sample Projects)
  - ♦ **MSMQeWay.sar** (to install the MSMQ eWay)
- 3 Click **upload now** to upload the selected products.
- 4 Click the Manifest File field’s **Browse** option. Browse to the Add-on **Documentation** directory, select the **ProductsManifest.xml**, and click **Submit**. The available Add-on documentation list is now displayed.
- 5 Select and upload the following file:
  - ♦ **MSMQeWayDocs.sar** (to upload the MSMQ eWay User’s Guide, Javadoc, Readme, and sample Projects to the Enterprise Manager).

- 6 Click **upload now** to upload the selected product.
- 7 Continue installing the eGate Integrator as instructed in the *SeeBeyond ICAN Suite Installation Guide*.

## 2.4.2 Adding the eWay to an Existing ICAN Suite Installation

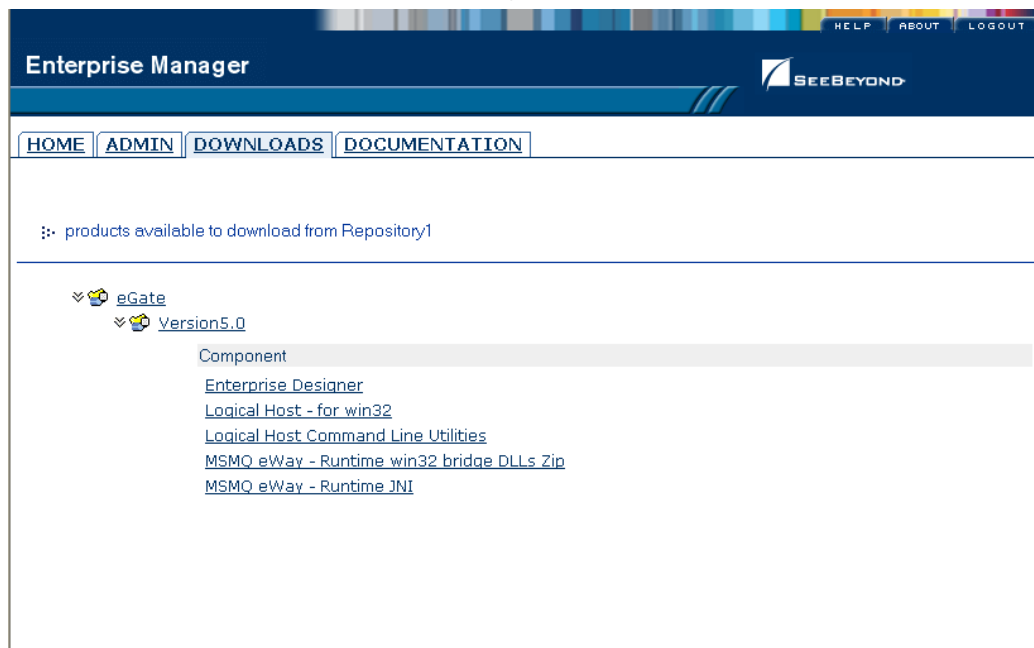
If you are installing the eWay to an existing ICAN installation, do the following:

- 1 Complete steps 1 through 7 above.
- 2 Open the Enterprise Designer and select **Update Center** from the Tools menu. The Update Center Wizard appears.
- 3 For Step 1 of the wizard, simply click **Next**.
- 4 For Step 2 of the wizard, click the **Add All** button to move all installable files to the **Include in Install** field. Click **Next**.
- 5 For Step 3 of the wizard, wait for the modules to download, then click **Next**.
- 6 The wizard's Step 4 window displays the installed modules. Click **Finish**.
- 7 When prompted, restart the IDE to complete the installation.

## 2.5 Downloading and Adding the MSMQ DLLs and JNI files

- 1 From the Enterprise Manager, click the **DOWNLOADS** tab. The Component list, as displayed in [Figure 1 on page 14](#), includes two MSMQ eWay components:
  - ♦ **MSMQ eWay - Runtime win32 bridge DLLs Zip** (msmqruntimejni.zip): This file contains a number of required DDL files that must be copied to the Windows Environment Variable Path.
  - ♦ **MSMQ eWay - Runtime JNI** (msmqjni.jar): This file must be uploaded using the Enterprise Designer.

**Figure 1** Enterprise Manager - DOWNLOADS



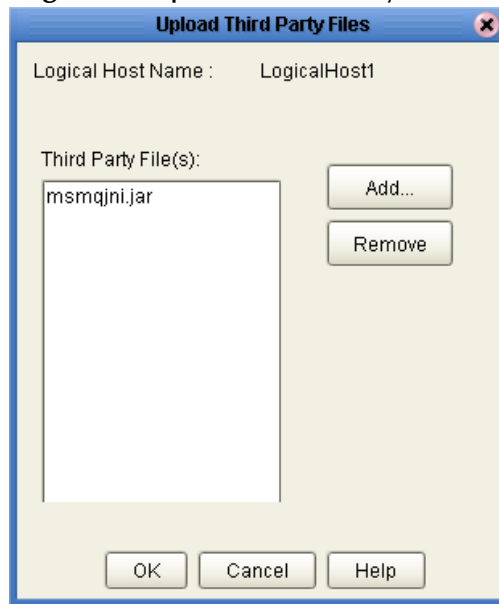
- 2 Download both files to a local directory. To add the msmqjni.jar file to the Windows Environment path, see [“Uploading a JAR file to the Logical Host” on page 14](#). To add the DLL file to the PATH for the Logical Host process, see [“Adding the DLL files to the Environment Path” on page 15](#).

### 2.5.1 Uploading a JAR file to the Logical Host

You must add the **Runtime JNI (msmqjni.jar)** to the path for the Logical Host process before running an MSMQ Project. After you have created an **Environment** and added a **Logical Host** for your Project, upload the JAR files to the Logical Host by doing the following:

- 1 From the Environment Explorer tree, right-click the **Logical Host** and select **Upload File** from the shortcut menu. The Upload Third Party Files dialog box appears.
- 2 Click **Add** and browse to the local directory to which you downloaded the JAR file. Add **msmqjni.jar** to the **Third Party Files** field (see [Figure 2 on page 15](#)). Click **OK**.

**Figure 2** Upload Third Party Files



For more information on creating an Environment, see [“Creating an Environment” on page 79](#).

## 2.5.2 Adding the DLL files to the Environment Path

The Runtime win32 bridge DLL files must be added to a directory that is included in the Windows Environment System Variable Path before running an MSMQ Project. To make sure that the correct file is added to the proper path location, do the following:

- 1 Locate the **msmqruntimejni.zip** file that you downloaded to a local folder.
- 2 Extract the ZIP file to a directory that is part of the Windows Environment path.

## 2.5.3 After Installation

Once the eWay is installed and configured, you must then incorporate it 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.

The MSMQ eWay’s User’s Guide, Javadoc, Readme, and sample Projects, can be accessed or downloaded from the Enterprise Manager’s Documentation tab.

# Configuring the MSMQ eWay

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

## What's in This Chapter

- [Using the Properties Editor](#) on page 18
- [Inbound MSMQ eWay Connectivity Map Properties](#) on page 20
- [Outbound MSMQ eWay Connectivity Map Properties](#) on page 23
- [Inbound MSMQ eWay Environment Properties](#) on page 27
- [Outbound MSMQ eWay Environment Properties](#) on page 28
- [Enabling Rollback When an MSMQ Message Fails](#) on page 30
- [Manually Configuring the eWay](#) on page 31
- [Alerting and Logging](#) on page 32

---

## 3.1 Creating and Configuring the MSMQ eWay

All eWays contain a set of properties that are unique to that eWay type. After the eWays are created and an MSMQ External System is added to the Project's Environment, the eWay parameters can be modified for your specific system. The MSMQ 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 MSMQ eWay's Connectivity Map properties vary depending on whether the eWay is an inbound or outbound eWay
- From the **Environment Explorer tree**. These properties are commonly global, and apply to all eWays (of the same type) in the Project. The saved properties are shared by all eWays in the MSMQ External System window.

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

### 3.1.1 Selecting MSMQ as the External Application

To create an MSMQ eWay, you must first create an MSMQ External Application in your Connectivity Map. MSMQ eWays are located between an MSMQ External Application

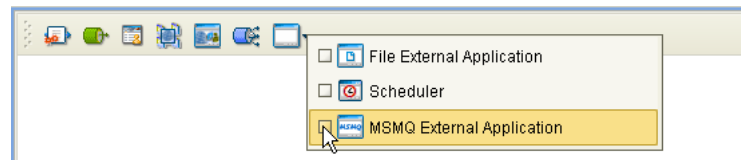


and a Service. Services are containers for Collaborations, Business Processes, eTL processes, and so forth.

### To create the MSMQ External Application

- 1 From the Connectivity Map toolbar, click the **External Applications** icon.
- 2 Select the **MSMQ External Application** from the menu (see Figure 3). The selected MSMQ External Application icon appears on the Connectivity Map toolbar.

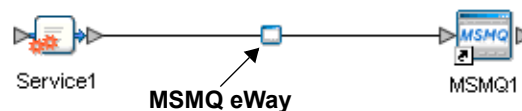
Figure 3 External Applications Selection Menu



- 3 Drag the new **MSMQ External Application** from the toolbar onto the Connectivity Map canvas. This represents an external MSMQ system.

From the Connectivity Map, you can associate (bind) the External Application to the Service to establish an eWay (see Figure 4).

Figure 4 eWay Location



When MSMQ is selected as the External Application, it automatically applies the default MSMQ eWay properties, provided by the OTD, to the eWay that connects it to the Service. You can then modify these properties for your specific system using the **Properties Editor**.

### 3.1.2 Modifying the MSMQ eWay Properties

A Project's eWay properties can be modified after the eWays have been created in the Connectivity Map and the External Applications have been added to the Project's Environment.

#### Modifying the MSMQ 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 MSMQ Connectivity Map properties. Make any necessary modifications and click **OK** to save the settings.

### Modifying the MSMQ eWay (Environment Explorer) Properties

- 1 From the Environment Explorer tree, right-click the Inbound or Outbound MSMQ external system. Select **Properties** from the shortcut menu. The **Properties Editor** appears.
- 2 Make any necessary modifications to the Environment parameters of the MSMQ eWays, and click **OK** to save the settings.

The eWay properties can be modified after the eWays have been created in the Connectivity Map and the Project's Environment has been created. MSMQ eWay properties are modified from two locations: from the Connectivity Map and from the Environment Explorer tree.

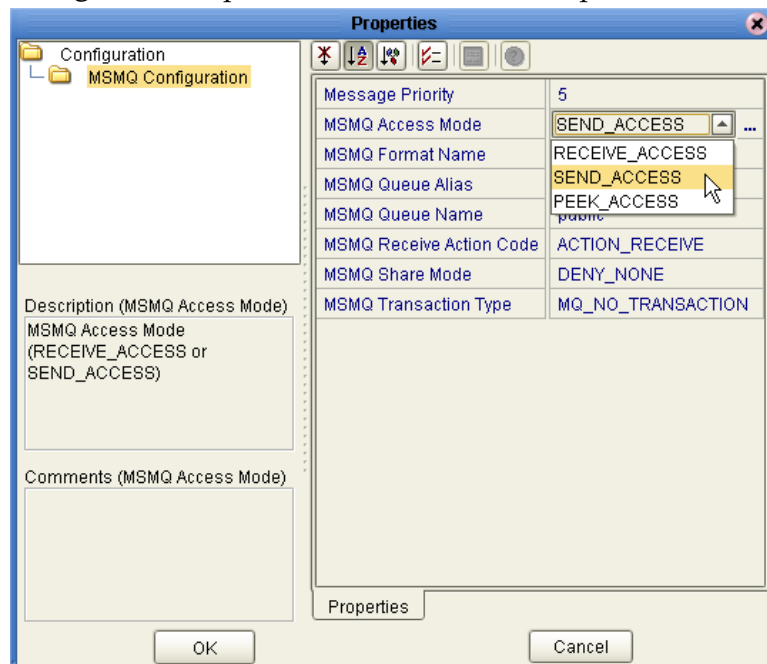
### 3.1.3 Using the Properties Editor

You can modify the default eWay configuration properties from the MSMQ eWay Properties Editor.

To modify the eWay properties do the following:

- 1 Open the MSMQ eWay Properties Editor for the eWay you want to edit. The MSMQ 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 from the Configuration tree. The parameters contained in that directory are now displayed in the right pane of the Properties Editor. For example, from the outbound eWay Connectivity Map Properties, click on **MSMQ Configuration** to display this section's editable parameters in the right pane, as shown in Figure 5.

Figure 5 Properties Editor -- MSMQ Properties



- 3 Click on any property field to make it editable. For example, click on the **MSMQ Access Mode** parameter to edit the MSMQ Access Mode 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 reference.
- 6 After modifying the configuration properties, click **OK** to close the Properties Editor and save the changes.

---

## 3.2 MSMQ eWay Properties

The MSMQ eWay's Properties are organized as follows:

[Inbound MSMQ eWay Connectivity Map Properties](#) on page 20

[Outbound MSMQ eWay Connectivity Map Properties](#) on page 23

[Inbound MSMQ eWay Environment Properties](#) on page 27

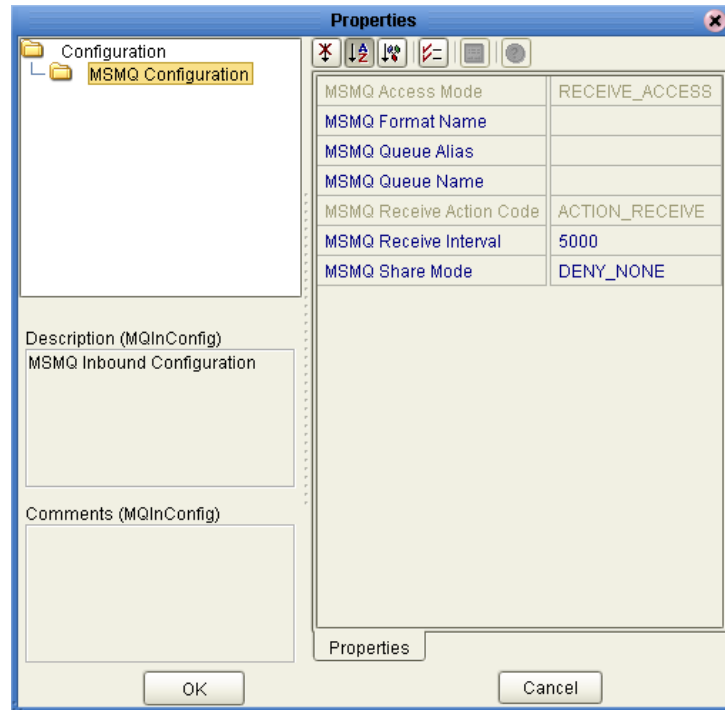
[Outbound MSMQ eWay Environment Properties](#) on page 28

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

## 3.3 Inbound MSMQ eWay Connectivity Map Properties

The **MSMQ Configuration** section of the inbound MSMQ eWay parameters contains the top-level properties displayed in Figure 6.

Figure 6 Inbound MSMQ eWay Connectivity Map Properties



### MSMQ Access Mode

#### Description

The MSMQ Access Mode inbound selection is not enabled. Only RECEIVE\_ACCESS is supported for inbound mode.

### Identifying an MSMQ Queue

The MSMQ eWay identifies an MSMQ Queue using the first available value of these three properties, in the following order:

- 1 MSMQ Queue Alias
- 2 MSMQ Format Name
- 3 MSMQ Queue Name

If you are using one of these properties, for example **Format Name**, to identify a queue, we recommend that you leave the value for the other two properties blank (in this case, you would leave the **MSMQ Queue Alias** and **MSMQ Queue Name** properties blank). This ensures that only the queue identification method you specify is used.

## MSMQ Format Name

### Description

Specifies the format name of the queue. The format name is a string that uniquely identifies a queue by providing some connection details and the queue's path, or by providing a special globally unique identifier (GUID). Different types of format names can be used to specify the way messages are routed, the type of destination, and the type of operation for which the queue is being opened.

To send or receive messages using a format name, **MSMQ** and the **Active Directory** service must be installed on the same computer as the Logical Host.

### Required Value

The public, private, or direct format name in the following manner

- Public = *QueueGUID*
- Private = *ComputerGuid\QueueNumber*
- Direct = *ComputerAddress\QueueName*

See [“Identifying an MSMQ Queue” on page 20](#) for more information.

## MSMQ Queue Alias

### Description

Specifies the queue alias. A queue alias associates an ADs path and a user-defined string (alias) with a public, private, or direct single-element format name.

To send or receive messages using a queue alias, **MSMQ** and the **Active Directory** service must be installed on the same computer as the Logical Host.

### Required Value

The queue alias.

See [“Identifying an MSMQ Queue” on page 20](#) for more information.

## MSMQ Queue Name

### Description

Specifies the Microsoft Message Queue Name.

To send or receive messages using a queue name, **MSMQ** and the **Active Directory** service must be installed on the same computer as the Logical Host.

### Required Value

The string name that identifies the queue. This is associated with the host name, so the value is entered as *host/queue*, where *host* is the host name and *queue* is the queue name.

See [“Identifying an MSMQ Queue” on page 20](#) for more information.

## MSMQ Receive Action Code

### Description

The MSMQ Receive Action Code inbound selection is not enabled. Only ACTION\_RECEIVE is supported for inbound mode.

## MSMQ Receive Interval

### Description

Specifies polling interval or frequency at which the eWay checks the queue for incoming messages in milliseconds.

### Required Value

A number indicating the polling interval in milliseconds. For example, 5000 milliseconds equals 5 seconds. The configured default is **5000**.

## MSMQ Share Mode

### Description

Specifies the MSMQ share mode (**DENY\_NONE** or **DENY\_RECEIVE\_SHARE**).

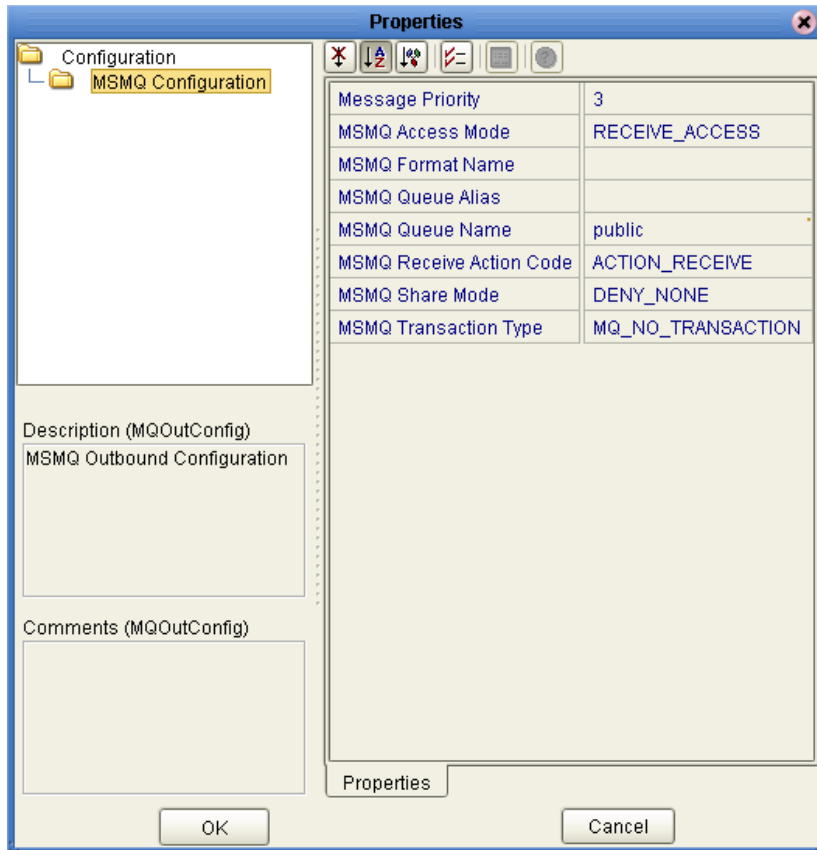
### Required Value

Select **DENY\_NONE** or **DENY\_RECEIVE\_SHARE**. **DENY\_NONE** is the configured default.

## 3.4 Outbound MSMQ eWay Connectivity Map Properties

The **MSMQ Configuration** section of the outbound MSMQ eWay Connectivity Map properties contains the top-level properties displayed in Figure 7.

**Figure 7** Outbound MSMQ eWay Connectivity Map Properties



### Message Priority

#### Description

Specifies the message priority level for all messages. Priorities range from level 0 (low priority) to 7 (high priority). Messages are stored in descending order in the queue, with higher priority messages first. Priority is only valid when **MSMQ Access Mode** is set to **SEND\_ACCESS** (see [“MSMQ Access Mode” on page 24](#))

#### Required Value

A number between 0 and 7 indicating the message priority. The configured default is 3.

## MSMQ Access Mode

### Description

Specifies whether Message Queuing opens the queue with peek, send, or receive access.

- **RECEIVE\_ACCESS**: Allows messages to be retrieved from a queue when they are read. This is used when the receiving application opens a queue to remove messages.
- **SEND\_ACCESS**: Allows messages to be sent to a queue. This is used when the sending application opens a queue to send messages.
- **PEEK\_ACCESS**: Allows messages to be read from a queue without removing them. This is used when the receiving application opens a queue to read message.

### Required Value

Select **RECEIVE\_ACCESS**, **SEND\_ACCESS**, or **PEEK\_ACCESS**. You must set Access Mode to **SEND ACCESS** to use Message Priority.

## Identifying an MSMQ Queue

The MSMQ eWay identifies an MSMQ Queue using the first available value of these three properties, in the following order:

- 1 MSMQ Queue Alias
- 2 MSMQ Format Name
- 3 MSMQ Queue Name

If you are using one of these properties, for example **Format Name**, to identify a queue, we recommend that you leave the value for the other two properties blank (in this case, you would leave the **MSMQ Queue Alias** and **MSMQ Queue Name** properties blank). This ensures that only the queue identification method you specify is used.

## MSMQ Format Name

### Description

Specifies the format name of the queue. The format name is a string that uniquely identifies a queue using connection details and the queue's path, or by providing a special globally unique identifier (GUID). Different types of format names can be used to specify how messages are routed, the type of destination, and the type of operation for which the queue is being opened.

To send or receive messages using a format name, **MSMQ** and the **Active Directory** service must be installed on the same computer as the Logical Host.

### Required Value

The format name. The public, private, or direct format name in the following manner

- Public = *QueueGUID*
- Private = *ComputerGuid\QueueNumber*



- Direct = *ComputerAddress\QueueName*

See [“Identifying an MSMQ Queue” on page 24](#) for more information.

## MSMQ Queue Alias

### Description

Specifies the queue alias. A queue alias associates an ADs path and a user-defined string (alias) with a public, private, or direct single-element format name.

To send or receive messages using a queue alias, **MSMQ** and the **Active Directory** service must be installed on the same computer as the Logical Host.

### Required Value

The queue alias.

See [“Identifying an MSMQ Queue” on page 24](#) for more information.

## MSMQ Queue Name

### Description

Specifies the Microsoft Message queue name.

To send or receive messages using a queue name, **MSMQ** and the **Active Directory** service must be installed on the same computer as the Logical Host.

### Required Value

The string name that identifies the queue. This is associated with the host name, so the value is entered as *host/queue*, where *host* is the host name and *queue* is the queue name.

See [“Identifying an MSMQ Queue” on page 24](#) for more information.

## MSMQ Receive Action Code

### Description

Specifies the MSMQ receive action code as one of the following:

- **ACTION\_RECEIVE**: Reads the message at the current cursor location and removes it from the queue.
- **ACTION\_PEEK\_CURRENT**: “Peeks” for last message in the queue. You can use this functionality when there are several messages coming in and going out of queue, and you want to peek (look) and get the latest message.

### Required Value

Select **ACTION\_RECEIVE** or **ACTION\_PEEK\_CURRENT**. **ACTION\_RECEIVE** is the configured default.

## MSMQ Share Mode

### Description

Specifies the MSMQ share mode as one of the following:

- **DENY\_NONE**: Grants everyone full access to send, peek at, or retrieve messages from the queue while it is open.
- **DENY\_RECEIVE\_SHARE**: You alone can peek at or retrieve messages until the queue is closed. Other applications are still able to send messages to the queue, but they may not delete messages while you have the queue open.

### Required Value

Select **DENY\_NONE** or **DENY\_RECEIVE\_SHARE**. **DENY\_NONE** is the configured default.

### Error Conditions

- An error message occurs if you or anyone tries to open the queue with **RECEIVE\_ACCESS** or **PEEK\_ACCESS** after Message Queuing opens the queue.
- If you attempt to open a queue with **DENY\_RECEIVE\_SHARE** when the queue is already open with **RECEIVE\_ACCESS** or **PEEK\_ACCESS**, the call will fail.

## MSMQ Transaction Type

### Description

Specifies the outbound transaction type. The options are:

- **MQ\_NO\_TRANSACTION**: Select this value when sending messages to a non-transactional queue.
- **MQ\_SINGLE\_MESSAGE**: Select this value when sending messages to a transactional queue.

This value does not apply to or affect inbound messages from a transactional queue. Use the default value, **MQ\_NO\_TRANSACTION**. For more information refer to your MSMQ user documentation.

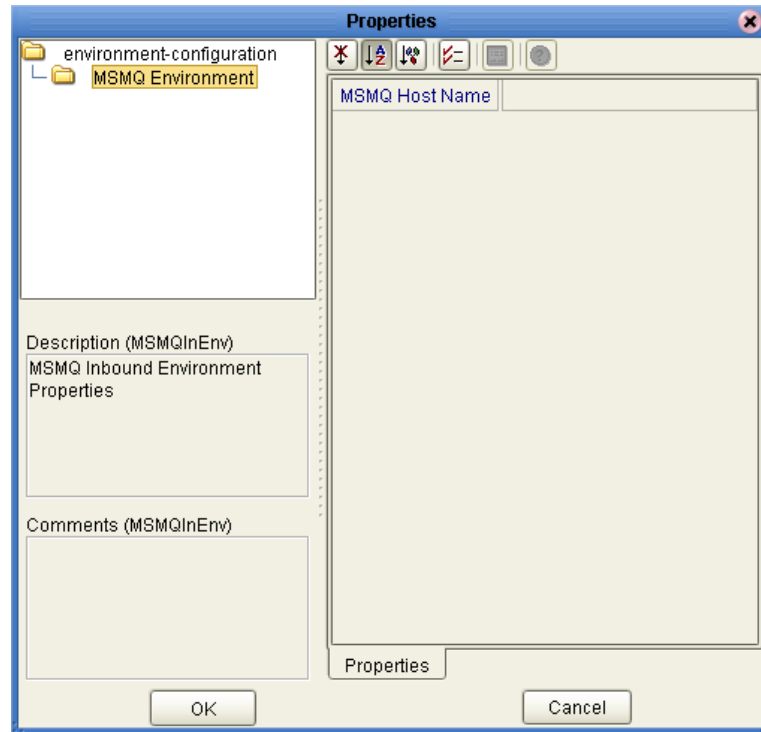
### Required Value

Select **MQ\_NO\_TRANSACTION** or **MQ\_SINGLE\_MESSAGE** as the transaction type. **MQ\_NO\_TRANSACTION** is the configured default.

## 3.5 Inbound MSMQ eWay Environment Properties

The **MSMQ Environment** section of the inbound MSMQ eWay Environment properties contains the top-level properties displayed in Figure 8.

Figure 8 Inbound MSMQ eWay Environment Properties



### MSMQ Host Name

#### Description

Specifies the Microsoft Message Queue host name.

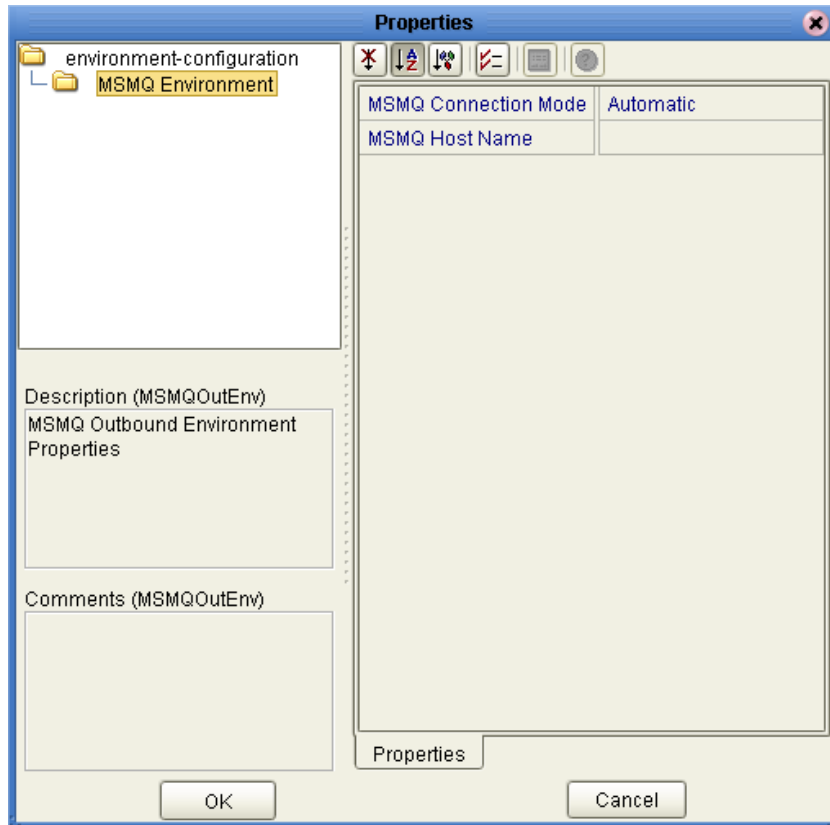
#### Required Value

The Microsoft Message Queue host name. Avoid using an IP address or “localhost” (for local server). These may not work in your MSMQ Environment.

## 3.6 Outbound MSMQ eWay Environment Properties

The **MSMQ Environment** section of the outbound MSMQ eWay Environment properties contains the top-level properties displayed in Figure 9.

**Figure 9** Outbound MSMQ eWay Environment Properties



### MSMQ Connection Mode

#### Description

Specifies the MSMQ connection mode. The options are:

- **Automatic:** The eWay automatically connects to the MSMQ server during the operation.
- **Manual (Dynamic):** Allows you to change the eWay configuration exposed in the Java Collaboration Definition. For more information see [“Manually Configuring the eWay” on page 31](#).

When using Business Processes (eInsight Projects), only Automatic mode is supported.

#### Required Value

Select **Automatic** or **Dynamic**. The configured default is **Automatic**.

## MSMQ Host Name

### Description

Specifies the Microsoft Message Queue host name.

### Required Value

The Microsoft Message Queue host name. Avoid using an IP address or “localhost” (for local server). These may not work in your MSMQ Environment.

## 3.7 Enabling Rollback When an MSMQ Message Fails

In order to roll back an outbound MSMQ message when a failure occurs in the Java Collaboration Definition (for example, failure to insert a duplicate row into a database table that is defined to have unique keys), do the following:

- 1 Select **Manual** as the MSMQ Connection Mode in the outbound MSMQ eWay Environment properties (see [“MSMQ Connection Mode” on page 28](#)).
- 2 Use the following JCD code:

```
try {
    MSMQClient_1.getEWayConfiguration().setOutMSMQName( "public" );
    MSMQClient_1.getEWayConfiguration().setOutMSMQShareMode(
"DENY_RECEIVE_SHARE" );
    MSMQClient_1.getEWayConfiguration().setOutMSMQReceiveActionCode(
"ACTION_PEEK_CURRENT" );
    MSMQClient_1.connect();
    MSMQClient_1.getMSMQMessage();
    TestDB_1.getTEST1().insert();
    TestDB_1.getTEST1().setTESTSTRING( "From JCD" );
    TestDB_1.getTEST1().insertRow();
    MSMQClient_1.getEWayConfiguration().setOutMSMQReceiveActionCode(
"ACTION_RECEIVE" );
    MSMQClient_1.getMSMQMessage();
    MSMQClient_1.disconnect();
} catch ( Exception Catch ) {
    MSMQClient_1.disconnect();
}
```

The key code items are highlighted in **bold**. You must also use a **try/catch block** to catch the exception (for this example, the error is a failed insert to database).

- 3 Open the queue with the following settings:
  - ♦ MSMQ Share Mode – **DENY\_RECEIVE\_SHARE**: This locks the queue so that other applications cannot open the queue in Receive mode. Until your application closes the queue, no other application can open the queue to receive the message.
  - ♦ MSMQ Receive Action Code – **ACTION\_PEEK\_CURRENT**: This opens the queue so you can “peek” (view) the message.
- 4 Next, try a database update on the message you received from the queue. If this fails, move on to step 7 (close the queue instance). If that succeeds, change the eWay configuration properties to receive messages programmatically in this manner (see [“MSMQ Receive Action Code” on page 22](#)).
- 5 Next, receive the message. This removes the message from queue.
- 6 Disconnect from the queue.
- 7 Finally, close the Queue using **MSMQClient\_1.disconnect()**. This closes the current queue instance and unlocks the queue.

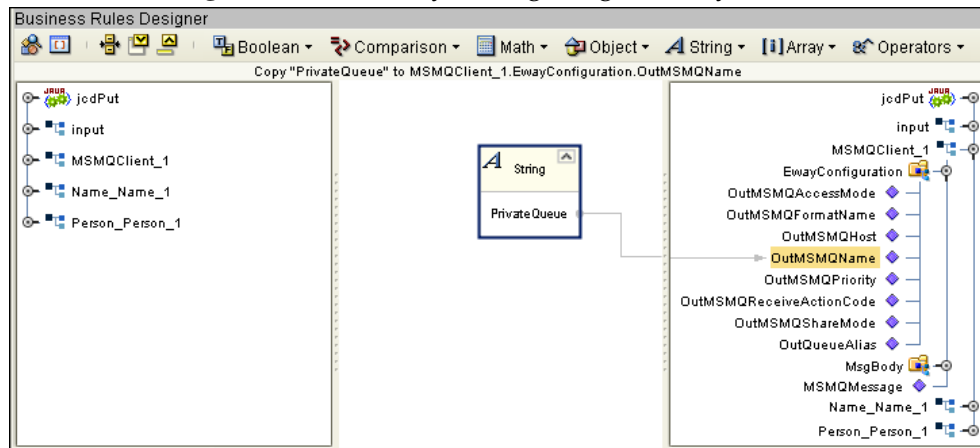
## 3.8 Manually Configuring the eWay

Manual mode allows you to override eWay configurations in the Java Collaboration Definition. To enable this, set the MSMQ Connection Mode (in the outbound Environment properties) to “Manual”. The MSMQ Connection Mode is only available for outbound mode (see “MSMQ Connection Mode” on page 28).

Collaborations cannot include both **Manual** mode and **Automatic** mode. Collaborations using **Manual** mode may not participate in connection pooling.

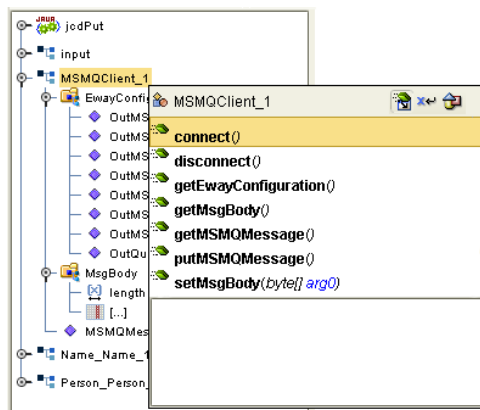
In Manual mode, all available properties and methods can be changed through the Collaboration’s OTDs. For example, to set the **OutMSMQName**, you assign a **literal string** to the **OutMSMQName** parameter on the right hand side of the Collaboration in the Collaboration Editor (see Figure 10). You can also change the property value using other sources, such as the content in an input file or the value from another OTD.

Figure 10 Manually Configuring an eWay



Once all of the new properties are assigned, you must use the **Connect()** method to establish a connection to the queue (see Figure 11).

Figure 11 Manually Configuring an eWay



After you use the **getMSMQMessage()** or **putMSMQMessage()** method to either retrieve or send a message, you must use the **disconnect()** method to disconnect from the queue.

---

## 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*.



# Using the MSMQ eWay With eInsight

This chapter describes how to use the MSMQ 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.

## What's in This Chapter

- [eInsight Engine and Components](#) on page 33
- [The MSMQ eWay With eInsight](#) on page 34
- [MSMQ eWay Sample Project for eInsight](#) on page 34
- [Importing a Sample Project](#) on page 35
- [Creating the MSMQ\\_BP\\_Sample Project](#) on page 35

---

## 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 MSMQ eWay With eInsight

An eInsight Business Process Activity can be associated with the MSMQ 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 MSMQ eWay has the following operators available:

- **receiveMSMQMessage**
- **putWSMSMQMessage**
- **getWSMSMQMessage**

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 MSMQ eWay.

---

## 4.3 MSMQ eWay Sample Project for eInsight

The MSMQ eWay includes one sample Project that uses eInsight, named **MSMQ\_BP\_Sample**. This sample Project performs three separate operations that demonstrate the following:

- **Get Operation:** The eWay subscribes to an MSMQ queue. The File eWay is used as a trigger. When a "trigger" message is received by the File eWay, the MSMQ eWay picks up any available XML data (messages) from the queue and publishes them to an eInsight Business Process. The Business Process unmarshals the message, modifies the data by converting a name from one value (first, last) to two values (first and last), and marshals the data. The message is then published to a outbound File eWay. The File eWay publishes the message an external directory.
- **Put Operation:** A File eWay subscribes to an external directory. XML data (message) is picked up from the input directory and published to an eInsight Business Process. The Business Process unmarshals the message, modifies the data converting a name from two values (first and last) to one value (first, last), and marshals the data. The message is then published to an MSMQ queue.
- **Receive Operation:** The eWay subscribes to an MSMQ queue. XML data (message) is picked up from the queue and published to an eInsight Business Process. The Business Process unmarshals the message, modifies the data converting a name from one value (first, last) to two values (first and last), and marshals the data. The message is then published to a outbound File eWay. The File eWay publishes the message to a directory.

Both the **Get** and **Receive** operations read the message from the queue. The **Get** operation requires a trigger, such as the presence of an input file. **Receive**, on the other hand, routinely polls the queue for messages. The polling rate is specified by the inbound property, MSMQ Receive Interval. No trigger is necessary.

This chapter provides directions for importing and completing the MSMQ\_BP\_Sample. It also explains how to create the sample Project manually.

---

## 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 You must upload sample files with the eWay's documentation SAR file and download them from the Enterprise Manager's Documentation tab. Extract the MSMQ eWay sample Projects 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 your local directory that contains the sample Project zip files. Select the sample file (for this sample, **MSMQ\_BP\_Sample.zip**) and click **Import**. After the sample Project is successfully imported, click **Close**.
- 4 Before you can run an imported sample Project, you must do the following:
  - ♦ Create an **Environment** (see ["Creating an Environment" on page 51](#)).
  - ♦ Configure the eWays properties for your specific system (see ["Configuring the MSMQ eWay" on page 16](#)).
  - ♦ Create a **Deployment Profile** (see ["Creating and Activating the Deployment Profile" on page 53](#)).
- 5 After you create the Deployment Profile, run the bootstrap (see ["Running a Project" on page 55](#)) and run the Project.

---

## 4.5 Creating the MSMQ\_BP\_Sample Project

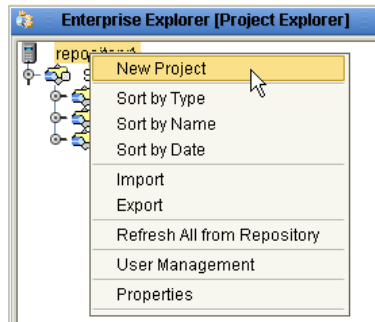
The following section explains how the sample Project's components are created. The first step is to create and name a new Project in eGate Enterprise Designer.

### 4.5.1 Creating a Project

First, create a new Project in the SeeBeyond Enterprise Designer.

- 1 From the Enterprise Explorer's Project Explorer tab, right-click the Repository and select **New Project** (see [Figure 12 on page 36](#)).

Figure 12 Enterprise Explorer - New Project



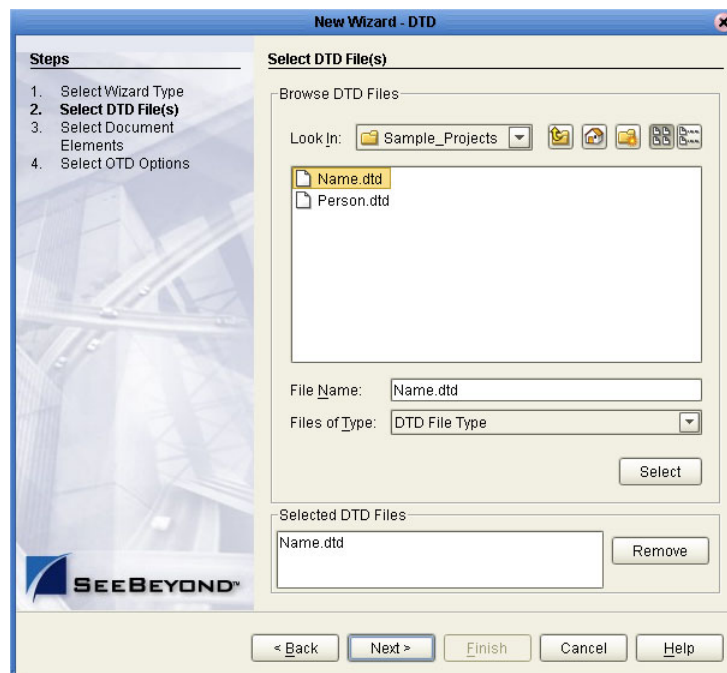
- 2 A new Project (Project1) appears on the Project Explorer tree. Rename the Project to MSMQ\_BP\_Sample.

### 4.5.2 Creating OTDs Using the DTD Wizard

Two sample XML Document Type Definitions (DTDs) are included with the sample Projects: **Name.dtd** and **Person.dtd**. Convert these to Object Type Definitions (OTDs) using the **Object Type Definition Wizard**.

- 1 From the Project Explorer tree, right-click the **MSMQ\_BP\_Sample** Project and select **New > Object Type Definition** from the shortcut menu. The **Object Type Definition Wizard** appears.
- 2 From the Select Wizard Type box, select **DTD**. The **DTD Wizard** appears.
- 3 For Step 2 of the wizard, "Select DTD File(s)," browse to the location where you saved the sample Project data, and select the **Name.dtd** file (see Figure 13) and click **Select**. Click **Next**.

Figure 13 DTD Wizard - Select DTD File(s)



- 4 For step 3 of the wizard, “Select Document Elements,” click **Next**.
- 5 For step 4 of the wizard, “Select OTD Options,” click **Finish**. The OTD Editor appears displaying the new **Name\_Name.otd**.
- 6 Repeat steps 1 - 4 using the Person.dtd to create the Person\_Person.otd.
- 7 After you have created both OTDs, save your work to the Repository and close the OTD Editor.

### 4.5.3 Creating the Business Processes

The next step to creating the sample is to create three Business Processes using the **eInsight Business Process Designer**. These are:

- **bpGet**: for the Get operation
- **bpPut**: for the Put operation
- **bpReceive**: for the Receive operation

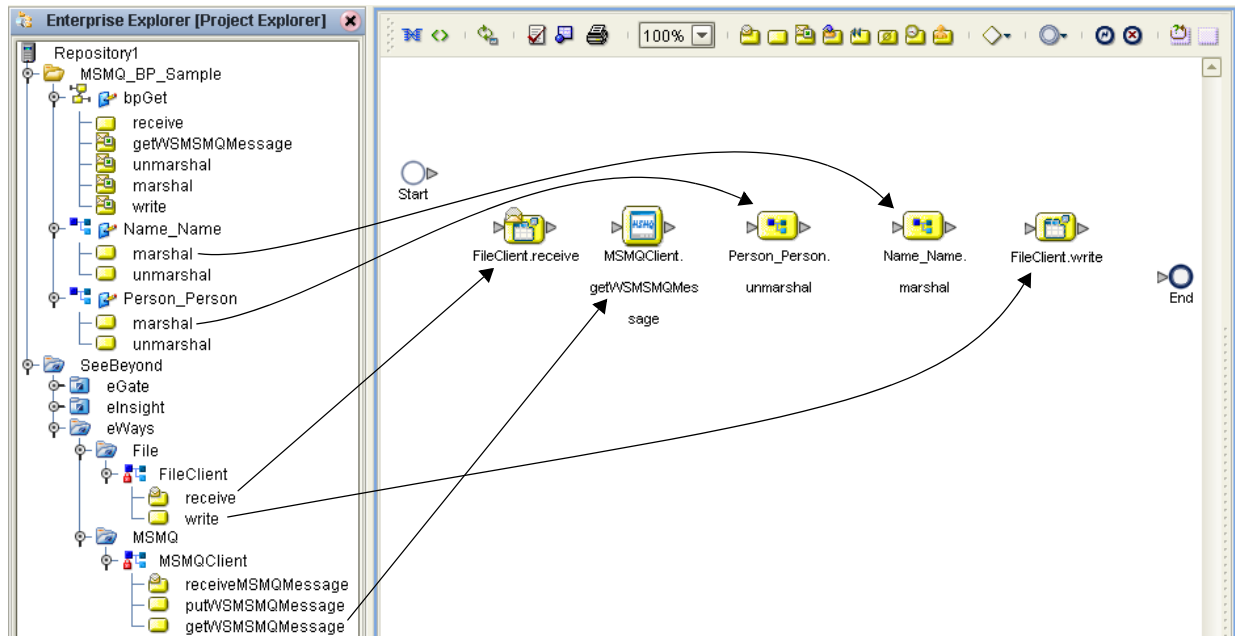
Once you have created the Business Process, the Business Rules of the Business Process are written using the Business Rules Designer.

## Creating the bpGet Business Process

### Creating the Business Process Flow

- 1 From the Project Explorer tree, right-click **MSMQ\_BP\_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 **bpGet**.
- 2 From the Project Explorer tree, expand **SeeBeyond > eWays > MSMQ > MSMQClient** nodes. In the same way, expand **SeeBeyond > eWays > File > FileClient** nodes to expose the available Business Process elements.
- 3 Populate the eInsight Business Process Designer **bpGet** canvas with the following elements from the Project Explorer tree, as displayed in [Figure 14 on page 38](#):
  - ♦ **receive**, under SeeBeyond > eWays > File > FileClient
  - ♦ **getWMSMQMessage** under SeeBeyond > eWays > MSMQ > MSMQClient
  - ♦ **unmarshal** under Person\_Person
  - ♦ **marshal** under Name\_Name
  - ♦ **write**, under SeeBeyond > eWays > File > FileClient

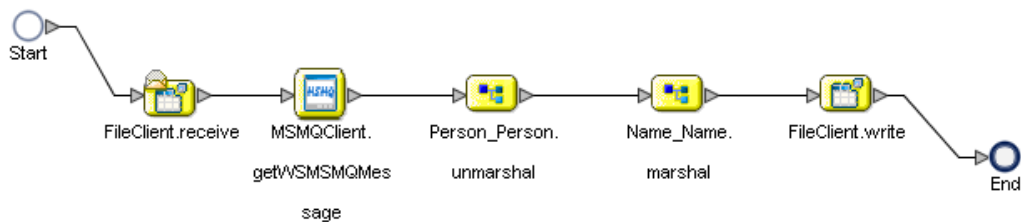
**Figure 14** eInsight Business Process Designer - Populate the Canvas



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 15. Link the following:

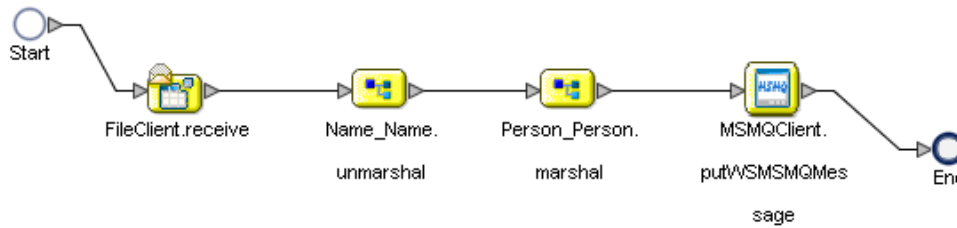
- ◆ Start to FileClient.receive
- ◆ FileClient.receive to MSMQClient.getWSMSMQMessage
- ◆ MSMQClient.getWSMSMQMessage to Person\_Person.unmarshal
- ◆ Person\_Person.unmarshal to Name\_Name.marshal
- ◆ Name\_Name.marshal to FileClient.write
- ◆ FileClient.write to End

**Figure 15** eInsight Business Process Designer - bpGet Business Process

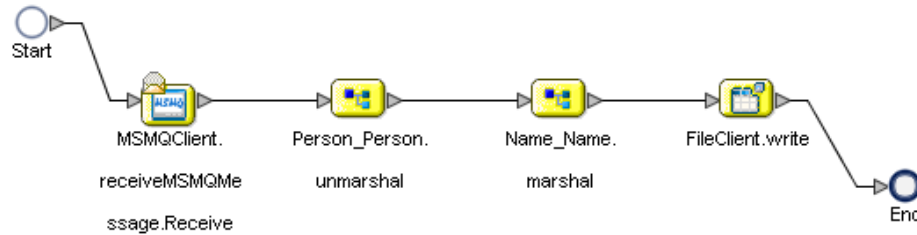


5 Create the **bpPut** and **bpReceive** Business Processes in the same manner, adding components and names to match those in [Figure 16 on page 39](#) for the **bpPut** Business Process, and in [Figure 17 on page 39](#) for the **bpReceive** Business Process.

**Figure 16** eInsight Business Process Designer - bpPut Business Process



**Figure 17** eInsight Business Process Designer - bpReceive Business Process



6 Save your current changes to the Repository.

#### 4.5.4 Creating the Business Rules Using the Business Rules Designer

Business Rules that you create between the Business Process elements define the relationship between the input and output Attributes of the elements.

#### Creating the bpGet Business Process Business Rules

- 1 From the Project Explorer tree, double-click the **bpGet** Business Process. The Business Process Designer opens, displaying the **bpGet** Business Process.
- 2 To create the **MSMQClient.Receive** to **Person\_Person.unmarshal** Business Rule, do the following:
  - A Right-click the link between the **MSMQClient.Receive** and **Person\_Person.unmarshal** activities, and select **Add Business Rule** from the shortcut menu as displayed in Figure 18.

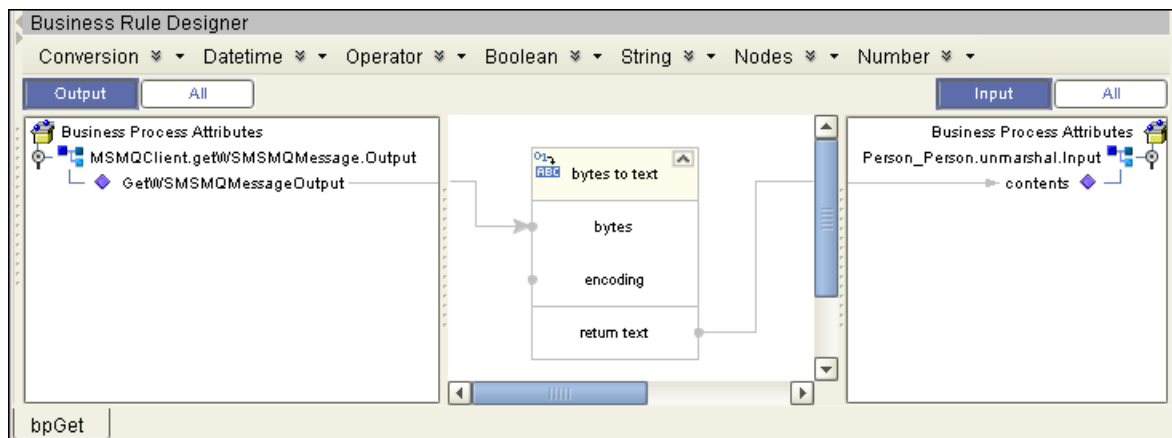
**Figure 18** eInsight Business Process Designer - Adding Business Rules



- B Double-click the Business Rule icon. The Business Rule Designer appears at the bottom of the eInsight Business Process Designer.

- C From the Business Rule Designer's String menu, select **Settings**. The Method Palette appears. From the **String** tab of the Method Palette, select **bytes to text** (click the check box for this option) and click **Close**. The **bytes to text** option is added to the **String** menu.
- D From the Business Rule Designer's String menu, select **bytes to text**. The **bytes to text** method box appears.
- E From the Output pane of the Business Rules Designer, map **GetWSMSMQMessageOutput**, under **MSMQClient.getWSMSMQMessage.Output**, to the **bytes** input node of the **bytes to text** method box. To do this, click on **ByteArrayData** and drag your cursor to the **bytes** input node of the **bytes to text** method box.
- F Map the **return text** output node of the **bytes to text** method box to **contents**, under **Person\_Person.unmarshal.input** in the Input pane of the Business Rule Designer (see Figure 19).

**Figure 19** eInsight Business Rule Designer

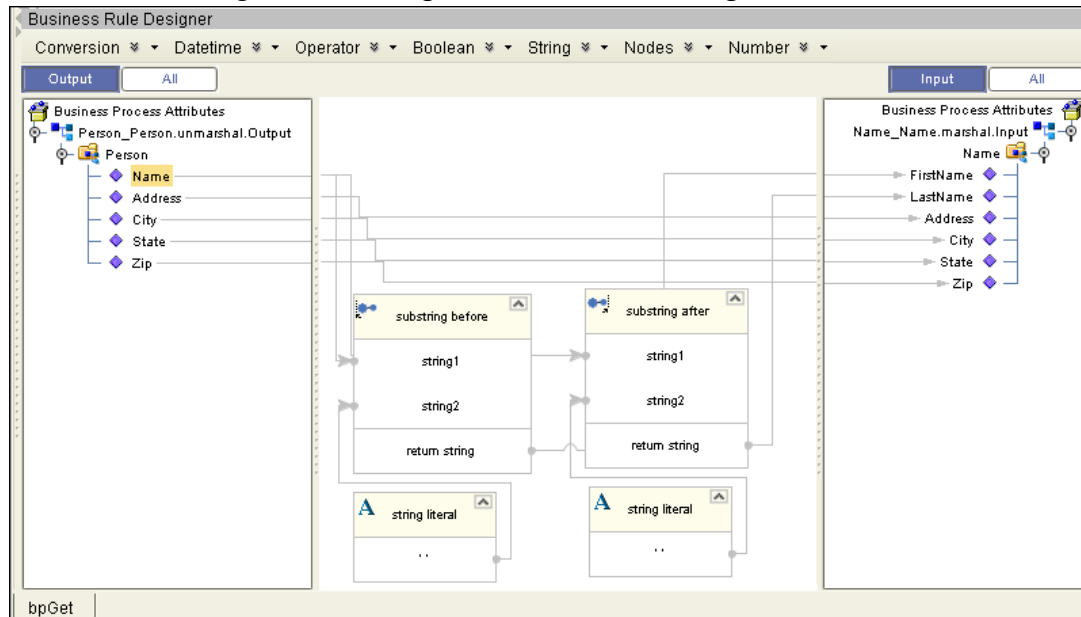


- 3 To create the **Person\_Person.unmarshal** to **Name\_Name.marshall** Business Rule, do the following:
  - A Right-click the link between the **Person\_Person.unmarshal** and **Name\_Name.marshall** activities and select **Add Business Rule** from the shortcut menu.
  - B From the Business Rule Designer's String menu, select **Settings**. The Method Palette appears. From the **String** tab of the Method Palette, select **substring before** and **substring after**, and click **Close**. The **substring before** and **substring after** options are added to the **String** menu.
  - C From the Business Rule Designer's String menu, select **substring before**. The **substring before** method box appears.
  - D Map **Name** under **Person\_Person.unmarshal.Output > Person**, in the Output pane of the Business Rule Designer, to the **string1** input node of the **substring before** method box.



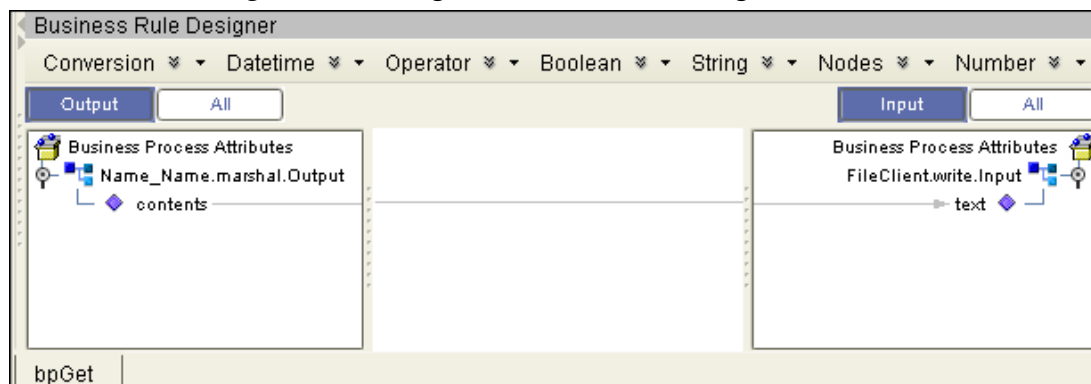
- E From the Business Rule Designer's String menu, select **string literal**. Enter one space (press the space bar one time) as the literal value and click **OK**. The **string literal** box appears.
- F Map the *space* output node of the first string literal box to the **string2** input node of the **substring before** method box.
- G Map the **return string** node of the **substring before** method box, to **FirstName** under Name\_Name.marshal.input > Name, in the Input pane of the Business Rule Designer.
- H From the Business Rule Designer's String menu, select **substring after**. The **substring after** method box appears.
- I Map **Name** under Person\_Person.unmarshal.Output > Person in the Output pane of the Business Rule Designer, to the **string1** input node of the **substring after** method box.
- J Repeat step E to create a second **string literal** box.
- K Map the *space* output node of the second string literal box to the **string2** input node of the **substring after** method box.
- L Map the **return string** node of the **substring after** method box, to **LastName** under Name\_Name.marshal.input > Name, in the Input pane of the Business Rule Designer.
- M Map **Address** under Person\_Person.unmarshal.Output > Person, in the Output pane of the Business Rule Designer, to **Address** under Name\_Name.marshal.input > Name, in the Input pane of the Business Rule Designer.
- N Map **City** under Person\_Person.unmarshal.Output > Person, in the Output pane, to **City** under Name\_Name.marshal.input > Name, in the Input pane.
- O Map **State** under Person\_Person.unmarshal.Output > Person, in the Output pane, to **State** under Name\_Name.marshal.input > Name, in the Input pane.
- P Map **Zip** under Person\_Person.unmarshal.Output > Person, in the Output pane, to **Zip** under Name\_Name.marshal.input > Name, in the Input pane (see [Figure 20 on page 42](#)).

Figure 20 eInsight Business Rule Designer



- 4 To create the **Name\_Name.marshal** to **FileClient.write** Business Rule, do the following:
  - A Right-click the link between the **Name\_Name.marshal** and **FileClient.write** activities and select **Add Business Rule** from the shortcut menu.
  - B From the Output pane of the Business Rules Designer, map **contents** under **Name\_Name.marshal.output**, to **text** under **FileClient.write.Input**, in the Input pane of the Business Rule Designer (see Figure 21).

Figure 21 eInsight Business Rule Designer



- 5 When all the Business Rules in the Business Process are complete, click the **Synchronize Graphical Model and Business Process Code** icon, on the Business Process Designer toolbar, to synchronize the graphical interface to the Business Process code.
- 6 Click the Enterprise Designer's **Save All** icon to save your current changes.

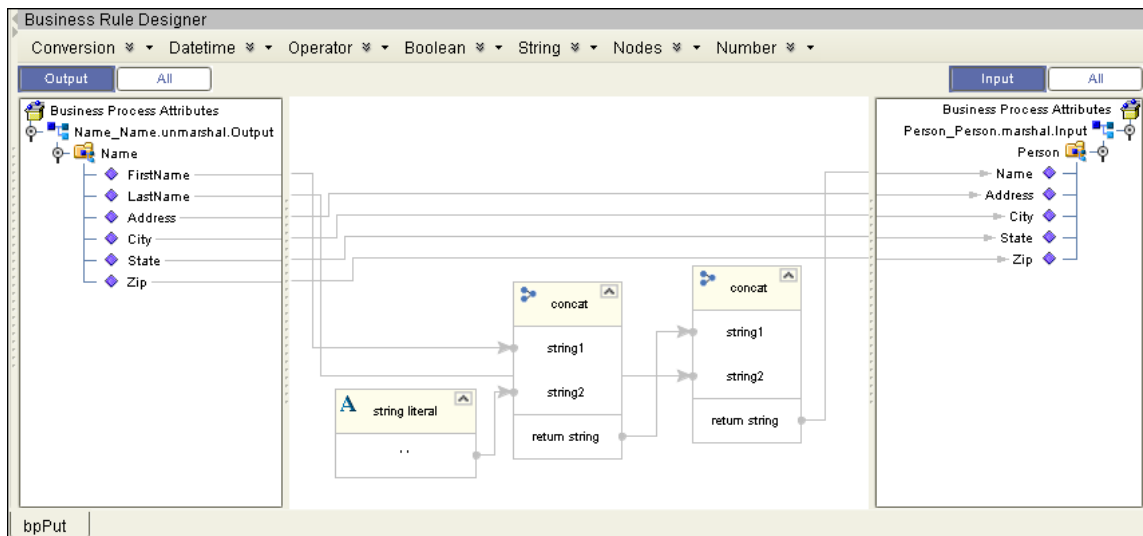
## Creating the bpPut Business Process Business Rules

- 1 Open the **bpPut** Business Process in the Business Process Designer.

- 2 To create the **FileClient.receive** to **Name\_Name.unmarshal** Business Rule, do the following:
  - A Right-click the link between the **FileClient.receive** and **Name\_Name.unmarshal** activities and select **Add Business Rule** from the shortcut menu.
  - B From the Output pane of the Business Rules Designer, map **text** under **FileClient.write.output** in the Output pane of the Business Rule Designer, to **contents** under **Name\_Name.unmarshal.Input**, in the Input pane of the Business Rule Designer.
- 3 To create the **Name\_Name.unmarshal** to **Person\_Person.marshall** Business Rule, do the following:
  - A Right-click the link between the **Name\_Name.unmarshal** and **Person\_Person.marshall** activities and select **Add Business Rule** from the shortcut menu.
  - B From the Business Rule Designer's String menu, select **concat**. The **concat** method box appears.
  - C From the Output pane of the Business Rules Designer, map **FirstName** under **Name\_Name.unmarshal.Output > Name**, to the **string1** input node of the **concat** method box.
  - D From the Business Rule Designer's String menu, select **string literal**. Enter one space (press the space bar one time) as the literal value and click **OK**. The **string literal** box appears.
  - E Map the *space* output node of the first **string literal** box to the **string2** input node of the **concat** method box.
  - F From the Business Rule Designer's String menu, select **concat**. A second **concat** method box appears.
  - G Map the **return string** node of the first **concat** method box to the **string1** input node of the second **concat** method box.
  - H From the Output pane of the Business Rules Designer, map **LastName**, under **Name\_Name.unmarshal.Output > Name**, in the Output pane of the Business Rule Designer, to the **string2** input node of the second **concat** method box.
  - I Map the **return string** node of the second **concat** method box to the **string1** input node of the second **concat** method box, to **Name**, under **Person\_Person.marshall.input > Person**, in the Input pane of the Business Rule Designer.
  - J From the Output pane of the Business Rules Designer, map **Address** under **Name\_Name.unmarshal.Output > Name**, to **Address**, under **Person\_Person.marshall.input > Person**, in the Input pane of the Business Rule Designer.
  - K From the Output pane of the Business Rules Designer, map **City**, under **Name\_Name.unmarshal.Output > Name**, to **City** under **Person\_Person.marshall.input > Person**, in the Input pane.

- L From the Output pane of the Business Rules Designer, map **State**, under Name\_Name.unmarshal.Output > Name, to **State** under Person\_Person.marshal.input > Person, in the Input pane.
- M From the Output pane of the Business Rules Designer, map **Zip**, under Name\_Name.unmarshal.Output > Name, to **Zip**, under Person\_Person.marshal.input > Person, in the Input pane.

**Figure 22** eInsight Business Rule Designer



- 4 To create the **Person\_Person.marshal** to **MSMQClient.putWSMQMessage** Business Rule, do the following:
  - A Right-click the link between the **Person\_Person.marshal** and **MSMQClient.putWSMQMessage** activities, and select **Add Business Rule** from the shortcut menu.
  - B From the Business Rule Designer’s String menu, select **Settings**. The Method Palette appears. From the **String** tab of the Method Palette, select **text to bytes** (Click the check box for this option) and click **Close**. The **text to bytes** option is added to the **String** menu.
  - C From the Business Rule Designer’s String menu, select **text to bytes**. The **text to bytes** method box appears.
  - D From the Output pane of the Business Rules Designer, map **contents** under Person\_Person.marshal.Output to the **text** input node of the **text to bytes** method box.
  - E Map the **return text** output node of the text to bytes method box to **payload** under MSMQClient.putWSMQMessage.Input, in the Input pane of the Business Rule Designer.
- 5 When all the Business Rules in the Business Process are complete, 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.

- 6 Save your current changes to the repository.

## Creating the bpReceive Business Process Business Rules

- 1 Open **bpReceive** in the Business Process Designer.
- 2 To create the **MSMQClient.receiveMSMQMessage.Receive to Person\_Person.unmarshal** Business Rule, do the following:
  - A Right-click the link between the **MSMQClient.receiveMSMQMessage.Receive** and **Person\_Person.unmarshal** activities and select **Add Business Rule** from the shortcut menu.
  - B From the Business Rule Designer's String menu, select **bytes to text**. The **bytes to text** method box appears.
  - C From the Output pane of the Business Rules Designer, map **payload**, under **MSMQClient.receiveMSMQMessage.Output**, to the **bytes** input node of the **bytes to text** method box.
  - D Map the **return text** output node of the **bytes to text** method box to **contents** under **Person\_Person.unmarshal.Input** in the Input pane of the Business Rules Designer.
- 3 To create the **Person\_Person.unmarshal to Name\_Name.marshal** Business Rule, do the following:
  - A Right-click the link between the **Person\_Person.unmarshal** and **Name\_Name.marshal** activities and select **Add Business Rule** from the shortcut menu.
  - B From the Business Rule Designer's String menu, select **substring before**. The **substring before** method box appears.
  - C From the Output pane of the Business Rules Designer, map **Name** under **Person\_Person.unmarshal.Output > Person**, to the **string1** input node of the **substring before** method box.
  - D From the Business Rule Designer's String menu, select **string literal**. Enter one space (press the space bar one time) as the literal value and click **OK**. The **string literal** box appears.
  - E Map the *space* output node of the first string literal box to the **string2** input node of the **substring before** method box.
  - F Map the **return string** node of the substring before method box to **FirstName** under **Name\_Name.marshal.input > Name**, in the Input pane of the Business Rule Designer.
  - G From the Business Rule Designer's String menu, select **substring after**. The **substring after** method box appears.
  - H From the Output pane of the Business Rules Designer, map **Name**, under **Person\_Person.unmarshal.Output > Person** to the **string1** input node of the **substring after** method box.
  - I Repeat step E to create a second **string literal** box.

- J Map the *space* output node of the second string literal box to the **string2** input node of the **substring after** method box.
  - K Map the **return string** node of the substring after method box to **LastName** under **Name\_Name.marshall.input > Name** in the Input pane of the Business Rule Designer.
  - L Map **Address**, under **Person\_Person.unmarshal.Output > Person** in the Output pane of the Business Rule Designer, to **Address**, under **Name\_Name.marshall.input > Name** in the Input pane of the Business Rule Designer.
  - M From the Output pane of the Business Rules Designer, map **City**, under **Person\_Person.unmarshal.Output > Person** to **City**, under **Name\_Name.marshall.input > Name** in the Input pane.
  - N Map **State**, under **Person\_Person.unmarshal.Output > Person** in the Output pane to **State**, under **Name\_Name.marshall.input > Name** in the Input pane.
  - O Map **Zip**, under **Person\_Person.unmarshal.Output > Person** in the Output pane to **Zip**, under **Name\_Name.marshall.input > Name** in the Input pane (see [Figure 20 on page 42](#)).
- 4 To create the **Name\_Name.marshall** to **FileClient.write** Business Rule, do the following:
    - A Right-click the link between the **Name\_Name.marshall** and **FileClient.write** activities and select **Add Business Rule** from the shortcut menu.
    - B From the Output pane of the Business Rules Designer, map **contents**, under **Name\_Name.marshall.output** to **text**, under **FileClient.write.Input** in the Input pane of the Business Rule Designer (see [Figure 21 on page 42](#)).
  - 5 When you have completed all the Business Rules in the Business Process, click the **Synchronize Graphical Model and Business Process Code** icon on the Business Process Designer toolbar to synchronize the graphical interface to the Business Process code.
  - 6 Save your current changes to the Repository.

### 4.5.5 Creating the Connectivity Maps

Connectivity Maps provide a canvas for assembling and configuring a Project's components. The MSMQ\_BP\_Sample uses three Connectivity Maps:

- **cmGet**: for the Get operation
- **cmPut**: for the Put operation
- **cmReceive**: for the Receive operation

To create the **cmGet** Connectivity Map, do the following:

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

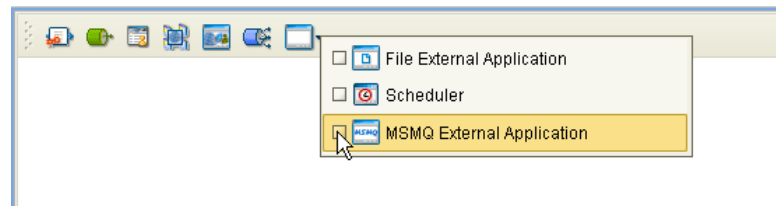
- 2 The New Connectivity Map appears and a node named **CMap1** is added under the Project on the Project Explorer tree. From the Project Explorer tree, rename the Connectivity Map to **cmGet**.

### Select the External Applications

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

eWays in the Connectivity Map are associated with External Application. For example, to establish a connection to an MSMQ queue, you must first select MSMQ as an External Application to use in your Connectivity Map (see Figure 23).

**Figure 23** Connectivity Map - External Applications



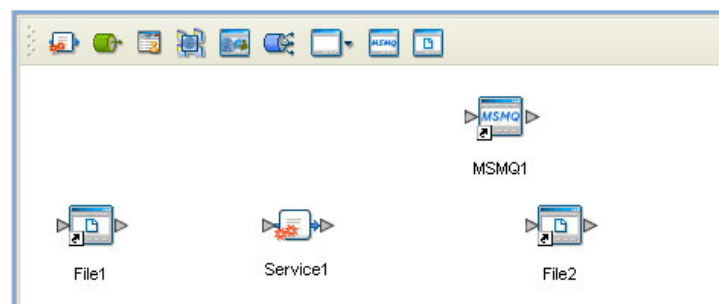
- 1 Click the **External Application** icon on the Connectivity Map toolbar.
- 2 Select the external systems needed for your Project (for this sample, the **MSMQ**, and **File** External Applications). Icons representing the selected external systems are added to the Connectivity Map toolbar.

### 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:
  - ♦ **1 MSMQ External System**
  - ♦ **1 Service** (a service is a container for Collaborations, Business Processes, eTL processes, and so forth)
  - ♦ **2 File External Systems**

**Figure 24** cmGet Connectivity Map with Components

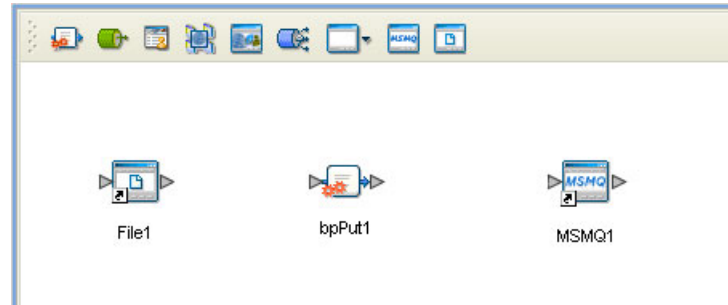


- 2 Rename the Service (Service1) by right-clicking the object and selecting Rename from the shortcut menu. Change the name to **bpGet1**.



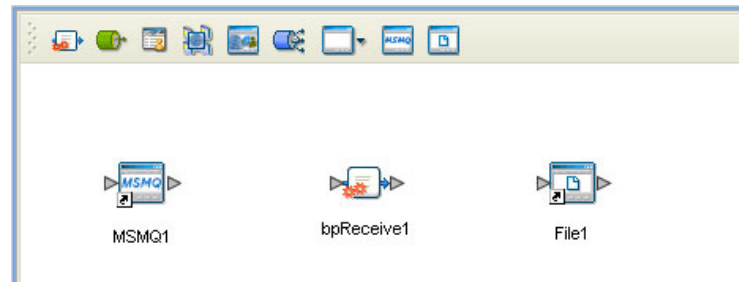
- 3 Create the **cmPut** and **cmReceive** Connectivity Maps in the same manner, adding components and names to match those in Figure 25 for the **cmPut** Connectivity Map, and in Figure 26 for the **cmReceive** Connectivity Map.

**Figure 25** cmPut Connectivity Map with Components



- 4 Rename the service in the **cmPut** Connectivity Map to **bpPut1**.

**Figure 26** cmReceive Connectivity Map with Components



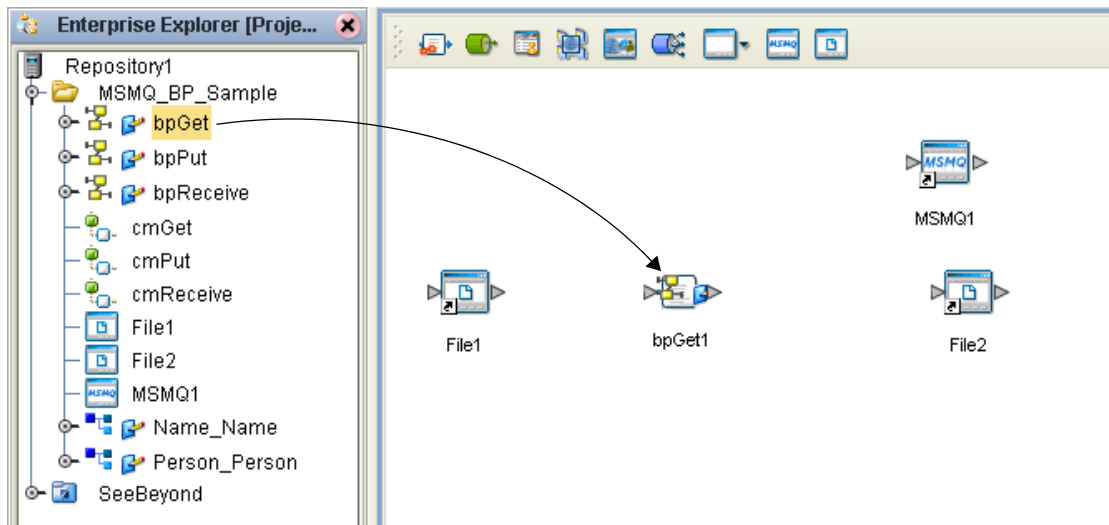
- 5 Rename the service in the **cmReceive** Connectivity Map to **bpReceive1**.
- 6 Save your current changes to the Repository.

#### 4.5.6 Binding the Project Components

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

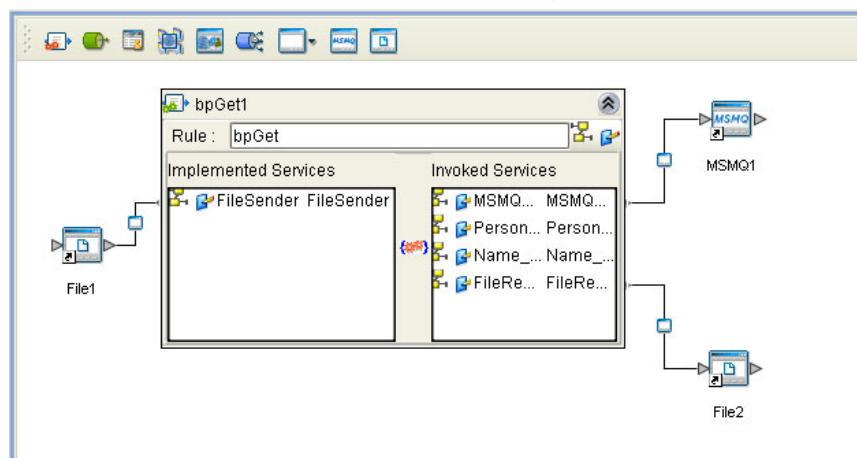
- 1 Open the **cmGet** Connectivity Map.
- 2 Drag and drop the **bpGet** Business Process from the Project Explorer tree to the **bpGet1** service (see [Figure 27 on page 50](#)).

Figure 27 Connectivity Map - Binding the Components



- 3 Double-click the **bpGet1** service. The **bpGet1** binding box appears with the **bpGet** Rule.
- 4 From the **bpGet1** binding box, map **FileSender** (under Implemented Services) to the **File1** application.
- 5 From the **bpGet1** binding box, map **MSMQReceiver** (under Invoked Services) to the **MSMQ1** External Application (see Figure 28).
- 6 From the **bpGet1** binding box, map **FileReceiver** (under Invoked Services) to the **File2** External Application (see Figure 28).

Figure 28 Connectivity Map - Binding the Components



- 7 Minimize the **bpGet1** binding box by clicking the chevrons in the upper-right corner and save your current changes to the Repository.

## Binding the components of the cmPut Connectivity Map

- 1 From the Project Explorer, double-click **cmPut**. The **cmPut Connectivity Map** appears.
- 2 Drag and drop the **bpPut** Collaboration from the Project Explorer tree onto the **bpPut1** service.
- 3 Double-click the **bpPut1** service. The **bpPut1** Binding dialog box appears using **bpPut** as the rule.
- 4 From the **bpPut1** Binding dialog box, drag **FileSender** (Implemented Services) to the **File1** External Application.
- 5 From the **bpPut1** Binding dialog box, drag **MSMQReceiver** (under Invoked Services) to the **MSMQ1** External Application.
- 6 Save your current changes to the Repository.

## Binding the components of the cmReceive Connectivity Map

- 1 From the Project Explorer, double-click **cmReceive**. The **cmReceive Connectivity Map** appears.
- 2 Drag and drop the **bpReceive** Collaboration from the Project Explorer tree onto the **bpReceive1** service.
- 3 Double-click the **bpReceive1** service. The **bpReceive1** Binding dialog box appears using **bpReceive** as the rule.
- 4 From the **bpReceive1** Binding dialog box, drag **MSMQSender** (Implemented Services) to the **MSMQ1** External Application.
- 5 From the **bpReceive1** Binding dialog box, drag **FileReceiver** (under Invoked Services) to the **File1** External Application.
- 6 Save your current changes to the Repository.

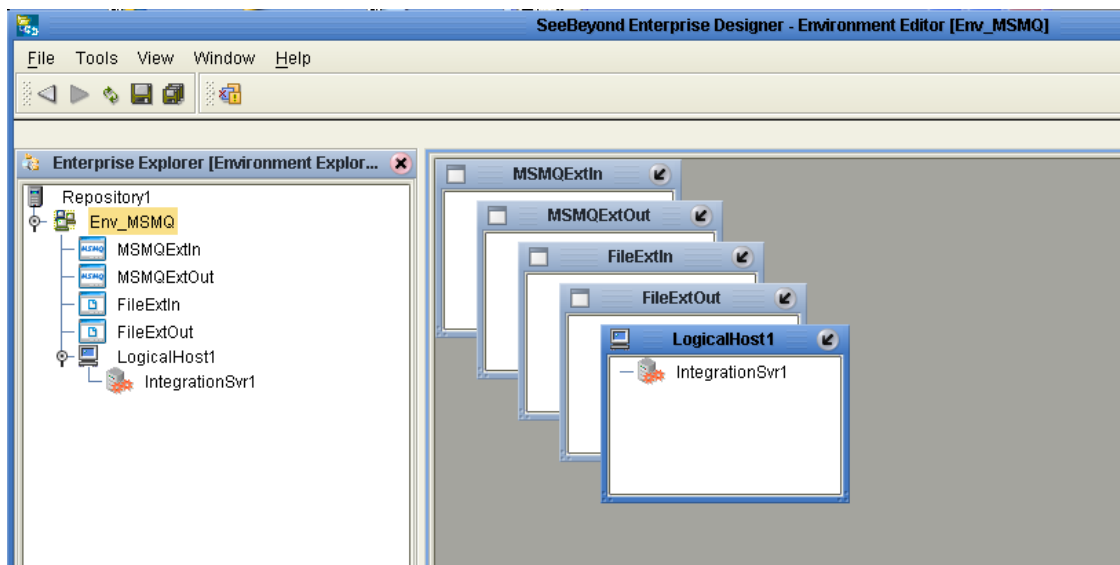
### 4.5.7 Creating an Environment

Environments include the external systems, Logical Hosts, integration servers and message servers used by a Project, and contain the configuration information for these components. You create Environments using the Enterprise Designer's Environment Editor. To create the MSMQ sample Environment do the following:

- 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 tree. Rename the new Environment to **Env\_MSMQ**.
- 3 Right-click **Env\_MSMQ** and select **New MSMQ External System**. Name the External System **MSMQExtIn** and select **Inbound MSMQ eWay** as the type. Click **OK**.

- 4 Right-click **Env\_MSMQ** and select **New MSMQ External System**. Name the External System **MSMQExtOut** and select **Outbound MSMQ eWay** as the type. Click **OK**.
- 5 Right-click **Env\_MSMQ** and select **New File External System**. From the Create an External System dialog box, name the External System **FileExtIn**, and select **Inbound File eWay** as the type. Click **OK**.
- 6 Right-click **Env\_MSMQ** and select **New File External System**. From the Create an External System dialog box, name the External System **FileExtOut**, and select **Outbound File eWay** as the type. Click **OK**.
- 7 Right-click **Env\_MSMQ** and select **New Logical Host**. The **LogicalHost1** box is added to the Environment and **LogicalHost1** is added to the Environment tree.
- 8 From the Environment tree, right-click the **LogicalHost1**, and select **New SeeBeyond Integration Server**. A new Integration Server, **IntegrationSvr1**, is added to the **LogicalHost1** box (see Figure 29).

**Figure 29** Environment Editor - Env\_MSMQ Environment



- 9 Save your current changes to the Repository.

## 4.5.8 Configuring the eWays

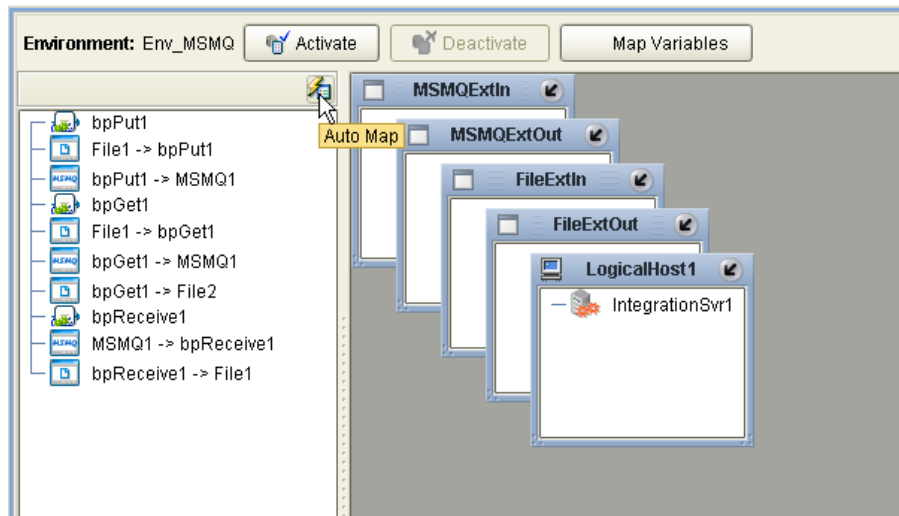
Configure the eWay properties for your system, including the settings for the input and output directories. For the most part, the eWay settings can be left with their default settings. See [“Configuring the MSMQ eWay” on page 16](#) for more information on configuring the eWays for your system.

## 4.5.9 Creating and Activating the Deployment Profile

Deployment Profiles assign Collaborations and Message Destinations to the Integration Server and Message Server. You create a Deployment Profiles using the Deployment Editor. To create the sample Deployment Profile, do the following.

- 1 From the Enterprise Explorer's Project Explorer, right-click the Project (**MSMQ\_BP\_Sample**) and select **New > Deployment Profile**.
- 2 Enter a name for the Deployment Profile (for this sample, **DP\_MSMQBP**). Make sure that the selected Environment is **Env\_MSMQ**. Click **OK**.
- 3 From the Deployment Editor toolbar, click the **Auto Map** icon (see Figure 30). The Project's components are automatically mapped to their system windows. If any of the Project components are not mapped automatically after Auto Map is used, those components can be mapped manually by following the appropriate steps below. Once all components are mapped, proceed to step 14.

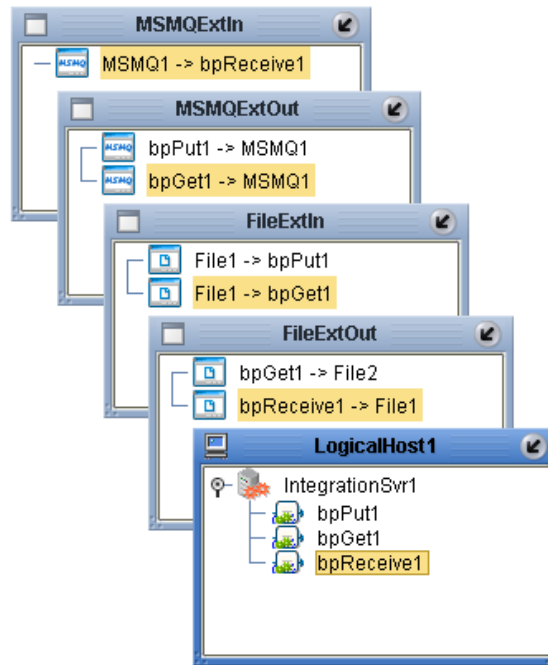
**Figure 30** Deployment Profile - Auto Map



- 4 From the left pane of the Deployment Editor, drag **MSMQ1 -> bpReceive1** (External Application) to the **MSMQExtIn** box.
- 5 Drag **bpPut1 -> MSMQ1** (External Application) to the **MSMQExtOut** box.
- 6 Drag **bpGet1 -> MSMQ1** (External Application) to the **MSMQExtOut** box.
- 7 Drag **File1 -> bpPut1** (External Application) to the **FileExtIn** box.
- 8 Drag **File1 -> bpGet1** (External Application) to the **FileExtIn** box.
- 9 Drag the **bpGet1 -> File2** (External Application) to the **FileExtOut** box.
- 10 Drag the **bpReceive1 -> File1** (External Application) to the **FileExtOut** box.
- 11 Drag the **bpPut1** service to **IntegrationSvr1** in the Logical Host (**LogicalHost1**) Environment box.
- 12 Drag the **bpGet1** service to **IntegrationSvr1** in the Logical Host (**LogicalHost1**) Environment box.

- 13 Drag the **bpReceive1** service to **IntegrationSvr1** in the Logical Host (**LogicalHost1**) Environment box (see [Figure 31 on page 54](#)).

**Figure 31** Deployment Profile



- 14 Click **Activate**. When activation succeeds, save your changes to the Repository.

---

## 4.6 Running a Project

The following procedure assume that the Enterprise Designer is located at **C:\ican50**. If this is not the case, replace **C:\ican50** in the following procedure with the appropriate location.

- 1 From the Enterprise Manager Downloads tab, download **Logical Host - for win32**.
- 2 Extract the file to the **ican50** directory.
- 3 Navigate to **C:\ican50\logicalhost\bootstrap\config** directory and open the **logical-host.properties** file using a text editor.
- 4 Enter the following information in the appropriate fields:
  - ♦ Logical Host root directory: **C:\ican50\logicalhost**
  - ♦ Repository URL: **http://localhost:port number/repository name**
  - ♦ Repository user name and password: *Your user name and password*
  - ♦ Logical Host Environment name: For example, **Env\_MSMQ**
  - ♦ Logical Host name: **LogicalHost1**

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

- 5 Run the **bootstrap.bat** file in the **ican50\logicalhost\bootstrap\bin** directory.
- 6 To run the **Get** operation, copy the sample **Trigger** file to the input folder with the appropriate extension.
- 7 To run the **Put** operation, copy the sample **Name** file to the input director with the appropriate extension.

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*.

# Implementing an MSMQ eWay Project

This chapter provides an introduction to the MSMQ 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*.

## What's in This Chapter

- [MSMQ eWay Components](#) on page 56
- [MSMQ eWay Sample Project](#) on page 57
- [Importing a Sample Project](#) on page 57

---

## 5.1 MSMQ eWay Components

This chapter presents a sample MSMQ 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 MSMQ eWay include the following:

- **MSMQ eWay Properties file:** The Properties file for the MSMQ eWay contains the parameters necessary to connect with a specific external system. These parameters are set with the Properties Editor. For more information about the MSMQ eWay Properties file and the Properties Editor, see [“Configuring the MSMQ eWay” on page 16](#).
- **MSMQClient OTD:** The **MSMQClient** OTD is provided with the eWay and contains methods and attributes used to create the Business Rules that invoke the **MSMQ** program.



---

## 5.2 MSMQ eWay Sample Project

The MSMQ eWay includes one sample Project that uses eGate with Java Collaboration Definitions, named **MSMQ\_JCD\_Sample**. This sample Project performs three separate operations that demonstrate the following:

- **Get Operation:** The eWay subscribes to an MSMQ queue. The File eWay is used as a trigger. When a “trigger” message is received by the File eWay, the MSMQ eWay picks up any available XML data (messages) from the queue and publishes them to a Java Collaboration. The Collaboration unmarshals the message, modifies the data by converting a name from one value (first, last) to two values (first and last), and marshals the data. The message is then published to a outbound File eWay. The File eWay publishes the message an external directory.
- **Put Operation:** A File eWay subscribes to an external directory. XML data (message) is picked up from the input directory and published to a Java Collaboration. The Collaboration unmarshals the message, modifies the data converting a name from two values (first and last) to one value (first, last), and marshals the data. The message is then published to an MSMQ queue.
- **Receive Operation:** The eWay subscribes to an MSMQ queue. XML data (message) is picked up from the queue and published to a Java Collaboration. The Collaboration unmarshals the message, modifies the data converting a name from one value (first, last) to two values (first and last), and marshals the data. The message is then published to a outbound File eWay. The File eWay publishes the message to a directory.

Both the **Get** and **Receive** operations read the message from the queue. The **Get** operation requires a trigger, such as the presence of an input file. **Receive**, on the other hand, routinely polls the queue for messages. The polling rate is specified by the inbound property, MSMQ Receive Interval. No trigger is necessary.

This chapter provides directions for importing and completing the **MSMQ\_JCD\_Sample**. It also explains how to create the sample Project manually.

---

## 5.3 Importing a Sample Project

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

- 1 You must Upload sample files with the eWay’s documentation SAR file and download them from the Enterprise Manager’s Documentation tab. Extract the MSMQ eWay sample Projects 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 your local directory that contains the sample Project zip files. Select the sample file (for this sample, **MSMQ\_JCD\_Sample.zip**) and click **Import**. After the sample Project is successfully imported, click **Close**.

- 4 Before you can run an imported sample Project, you must do the following:
  - ♦ Create an **Environment** (see “[Creating an Environment](#)” on page 79).
  - ♦ Configure the eWays properties for your specific system (see “[Configuring the MSMQ eWay](#)” on page 16).
  - ♦ Create a **Deployment Profile** (see “[Creating and Activating the Deployment Profile](#)” on page 80).
- 5 After you create the Deployment Profile, run the bootstrap (see “[Running a Project](#)” on page 82) and run the Project.

---

## 5.4 Creating the MSMQ\_JCD\_Sample Project

The following section explains how the sample Project’s components are created. The first step is to create and name a new Project in eGate Enterprise Designer.

### 5.4.1 Creating a New Project

- 1 From the Enterprise Designer’s 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 **MSMQ\_JCD\_Sample**.

### 5.4.2 Creating the Connectivity Maps

Connectivity Maps provide a canvas for assembling and configuring a Project’s components. The MSMQ\_JCD\_Sample uses three Connectivity Maps:

- **cmGet**: for the Get operation
- **cmPut**: for the Put operation
- **cmReceive**: for the Receive operation

To create the **cmGet** Connectivity Map, do the following:

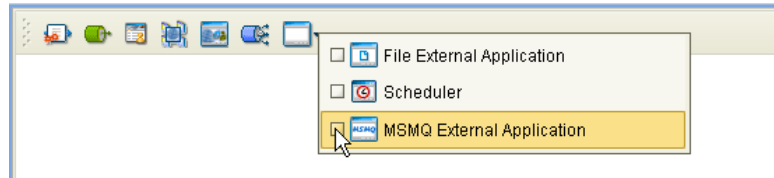
- 1 In Enterprise Explorer’s Project Explorer, right-click the **MSMQ\_JCD\_Sample** Project and select **New > Connectivity Map** from the shortcut menu.
- 2 The New Connectivity Map appears and a node named **CMap1** is added under the Project on the Project Explorer tree. From the Project Explorer tree, rename the Connectivity Map to **cmGet**.

#### Select the External Applications

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

eWays in the Connectivity Map are associated with External Application. For example, to establish a connection to an MSMQ queue, you must first select MSMQ as an External Application to use in your Connectivity Map (see Figure 32).

**Figure 32** Connectivity Map - External Applications



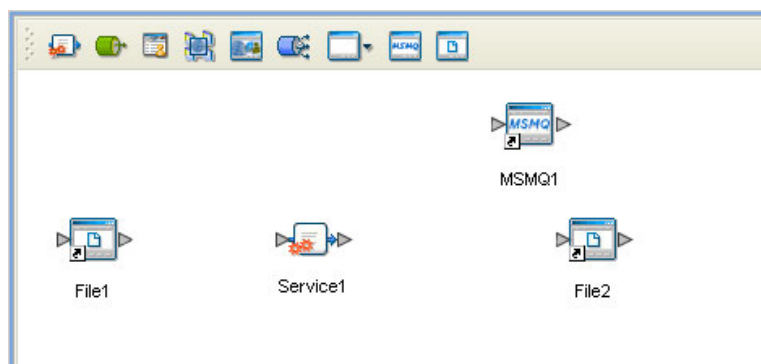
- 1 Click the **External Application** icon on the Connectivity Map toolbar.
- 2 Select the external systems needed for your Project (for this sample, the **MSMQ**, and **File** External Applications). Icons representing the selected external systems are added to the Connectivity Map toolbar.

### Populate the Connectivity Map

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

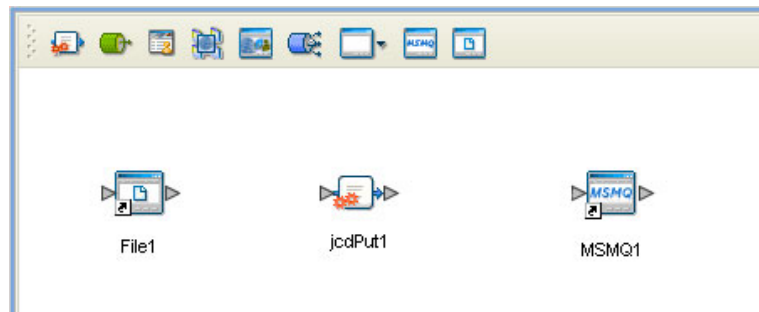
- 1 For this sample, drag and drop the following components onto the canvas as displayed in Figure 33:
  - ♦ 1 **MSMQ External System**
  - ♦ 1 **Service** (A service is a container for Collaborations, Business Processes, eTL processes, and so forth)
  - ♦ 2 **File External Systems**

**Figure 33** cmGet Connectivity Map with Components



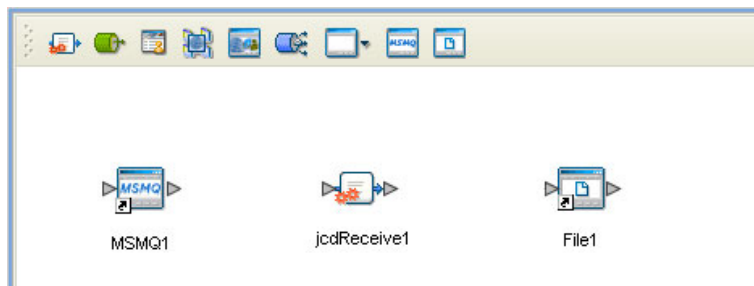
- 2 Rename the Service (Service1) by right-clicking the object and selecting Rename from the shortcut menu. Change the names to **jcdGet1**.
- 3 Create the **cmPut** and **cmReceive** Connectivity Maps in the same manner, adding components and names to match those in [Figure 34 on page 60](#) for the **cmPut** Connectivity Map, and in [Figure 35 on page 60](#) for the **cmReceive** Connectivity Map.

**Figure 34** cmPut Connectivity Map with Components



- 4 Rename the service in the **cmPut** Connectivity Map to **jcdPut1**.

**Figure 35** cmReceive Connectivity Map with Components



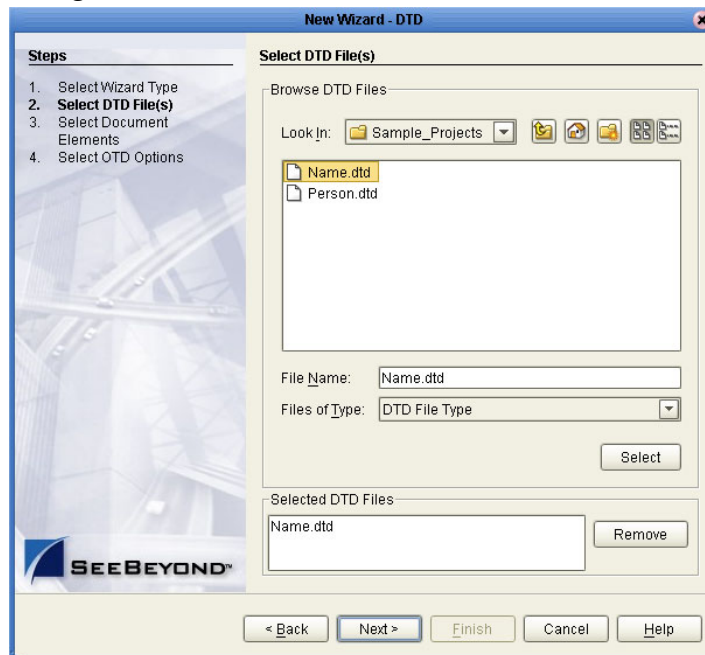
- 5 Rename the service in the **cmReceive** Connectivity Map to **jcdReceive1**.
- 6 Save your current changes to the Repository.

### 5.4.3 Creating OTDs Using the DTD Wizard

Two sample XML Document Type Definitions (DTDs) are included with the sample Projects: **Name.dtd** and **Person.dtd**. Convert these to Object Type Definitions (OTDs) using the **Object Type Definition Wizard**.

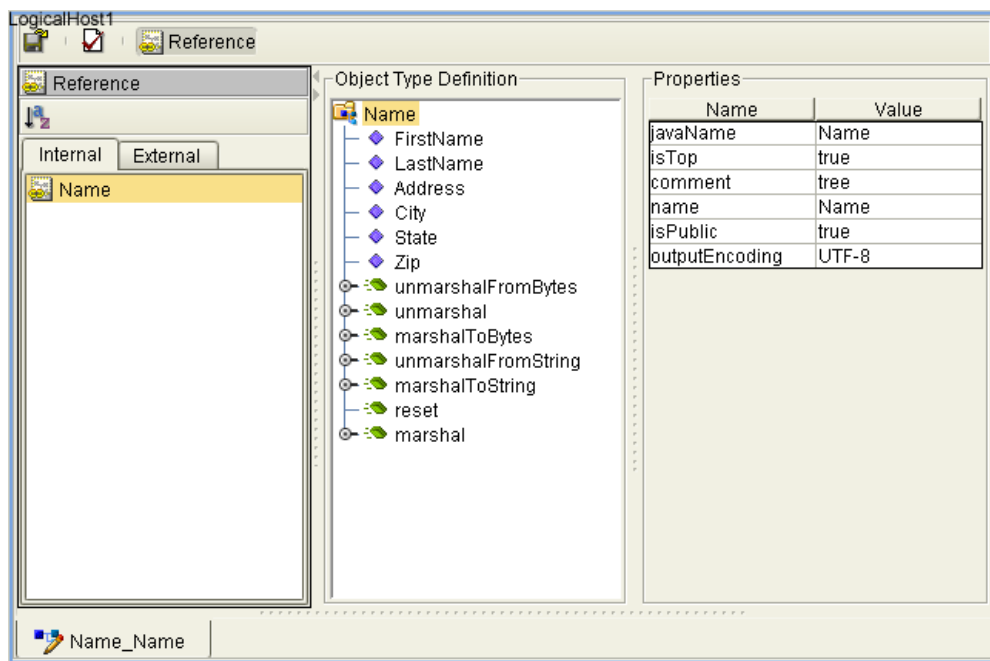
- 1 From the Project Explorer tree, right-click the **MSMQ\_JCD\_Sample** Project and select **New > Object Type Definition** from the shortcut menu. The **Object Type Definition Wizard** appears.
- 2 From the Select Wizard Type box, select **DTD**. The **DTD Wizard** appears.
- 3 For Step 2 of the wizard, “Select DTD File(s),” browse to the location where you saved the sample Project data, and select the **Name.dtd** file (see [Figure 36 on page 61](#)) and click **Select**. Click **Next**.

**Figure 36** DTD Wizard - Select DTD File(s)



- 4 For step 3 of the wizard, “Select Document Elements,” click **Next**.
- 5 For step 4 of the wizard, “Select OTD Options,” click **Finish**. The OTD Editor appears displaying the new **Name\_Name.otd** (see Figure 37).

**Figure 37** DTD Wizard - Select DTD File(s)



- 6 Repeat steps 1 - 4 using the **Person.dtd** to create the **Person\_Person.otd**.
- 7 After you have created both OTDs, save your work to the Repository and close the OTD Editor.

## 5.4.4 Creating the Collaboration Definitions

The next step to creating the sample is to create three Collaborations using the **Collaboration Definition Wizard (Java)**. These are:

- **jcdGet**: for the Get operation
- **jcdPut**: for the Put operation
- **jcdReceive**: for the Receive operation

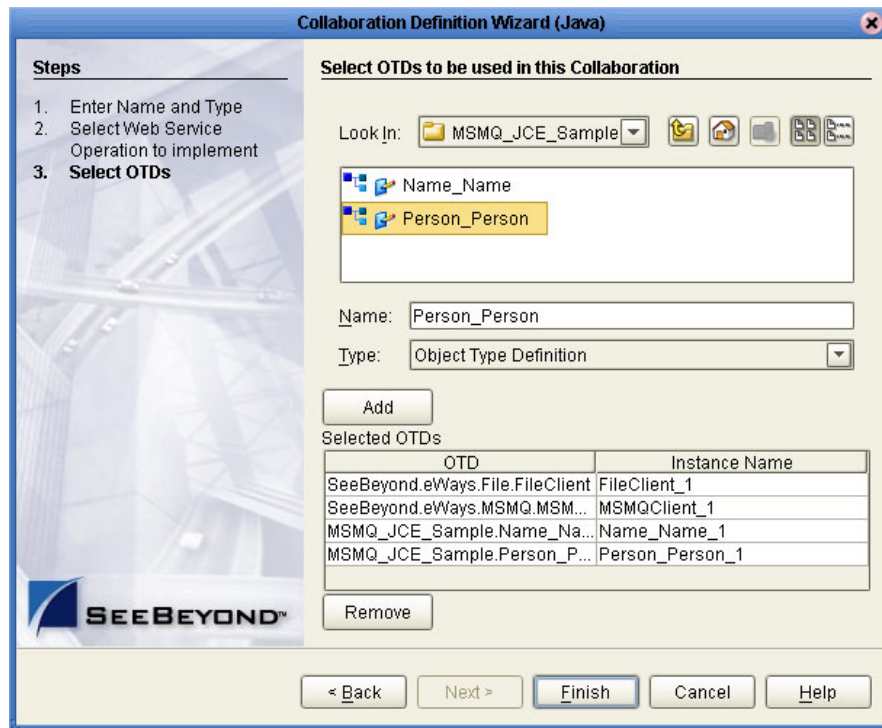
Once you have created the Collaboration Definitions, the Business Rules of the Collaborations are written using the Collaboration Editor.

### Creating the jcdGet Collaboration

The **jcdGet** Collaboration defines transactions from the MSMQ queue, using a File eWay as a trigger, to an outbound File eWay, and in process, modifies the data converting a name from one value (first, last) to two values (first and last). To create the **jcdGet** Collaboration, do the following:

- 1 From the Project Explorer tree, right-click the **MSMQ\_JCD\_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 **jcdGet**) and click **Next**.
- 3 In Step 2 of the wizard, “Select Web Service Operation to Implement,” from the Web Services Interfaces selection window, select **SeeBeyond > eWays > MSMQClient > getWSMSMQMessage**, and click **Next**.
- 4 In Step 3, “Select OTDs,” click **SeeBeyond > eWays > File** and double-click **FileClient**. The **FileClient\_1** OTD is added to the Selected OTDs field.
- 5 Click the **Up One Level** button to return to the Repository level, and click **SeeBeyond > eWays > MSMQ** and double-click **MSMQClient**. The **MSMQClient\_1** OTD is added to the Selected OTDs field.
- 6 Click the **Up One Level** button to return to the Repository level, and click **MSMQ\_JCD\_Sample** and double-click **Name\_Name**, and double-click **Person\_Person**. The **Name\_Name\_1** OTD and **Person\_Person\_1** OTD are added to the Selected OTDs field (see [Figure 38 on page 63](#)).

**Figure 38** Collaboration Definition Wizard (Java) - Select OTDs Interface



- 7 Click **Finish**. The Collaboration Editor (Java) appears with the new **jcd\_MQGet** Collaboration in the right pane of the Enterprise Designer.

## Creating the jcdPut Collaboration

The **jcdPut** Collaboration defines transactions from a File eWay to an MSMQ queue, and in process, modifies the data converting a name from one value (first, last) to two values (first and last). To create the **jcdPut** Collaboration, do the following:

- 1 Follow the steps for creating the **jcdGet** Collaboration, with the following changes:
  - A Enter **jcdPut** as the name of the new Collaboration.
  - B Select **SeeBeyond > eWays > MSMQClient > putWSMSMQMessage** as the Web Service Operation.
  - C Apply only steps 5 and 6 when selecting OTDs in step 3 of the wizard.
- 2 Click **Finish**. The Collaboration Editor (Java) appears with the new **jcdPut** Collaboration in the right pane of the Enterprise Designer.

## Creating the jcdReceive Collaboration

The **jcdReceive** Collaboration defines transactions from an MSMQ queue to a File eWay, and in process, modifies the data converting a name from two values (first and last) to one value (first, last). To create the **jcdReceive** Collaboration, do the following:

- 1 Follow the steps for creating the **jcdGet** Collaboration, with the following changes:
  - A Enter **jcdReceive** as the name of the new Collaboration.

- B Select SeeBeyond > eWays > MSMQClient > receiveMSMQMessage as the Web Service Operation.
  - C Apply only steps 4 and 6 when selecting OTDs in step 3 of the wizard.
- 2 Click **Finish**. The Collaboration Editor (Java) appears with the new **jcdReceive** Collaboration in the right pane of the Enterprise Designer.

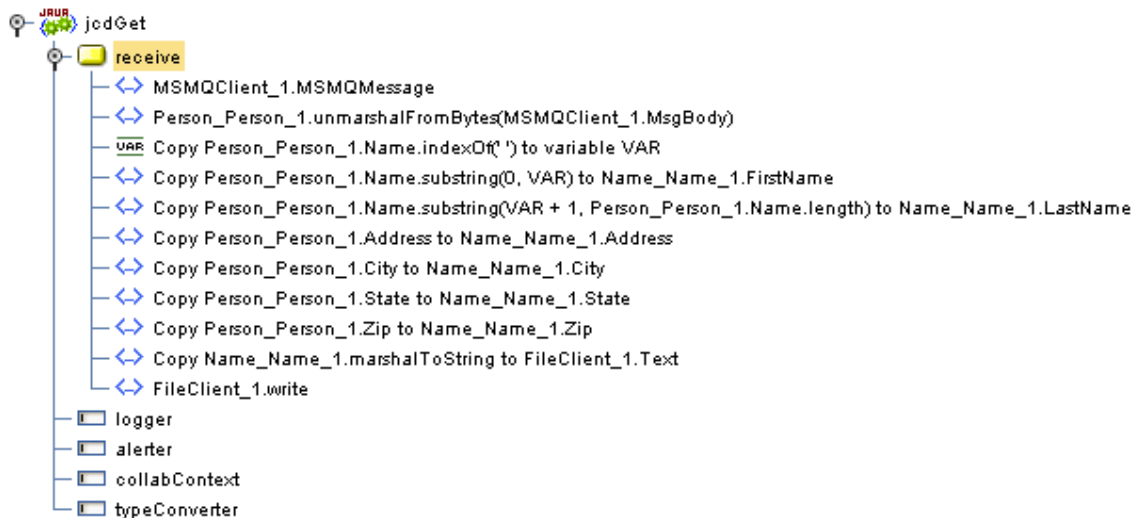
## 5.4.5 Creating the Business Rules Using the Collaboration Editor

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

### Creating the jcdGet Collaboration Business Rules

The **jcdGet** Collaboration contains the Business Rules displayed in Figure 39.

**Figure 39** jcdGet Collaboration Business Rules



To create the **jcdGet** Collaboration Business Rules, do the following:

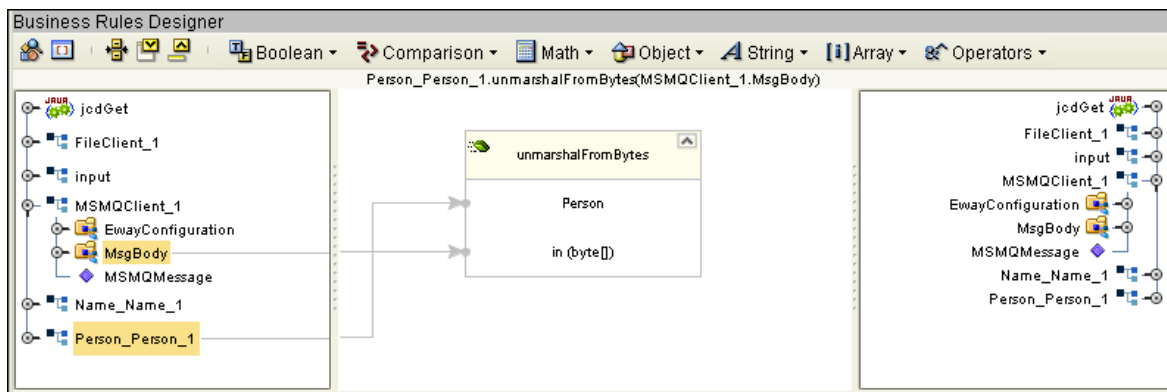
- 1 Create the **MSMQClient\_1.MSMQMessage** Business Rule:
  - A Right-click the **MSMQClient\_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
  - B Select **getMSMQMessage()** from the method selection menu. The **getMSMQMessage** method box appears with an arrow connecting **MSMQMessage\_1**, in the left pane of the Business Rule Designer, to the **MSMQClientApplication** input node.
  - C This creates the following Java code shown in the Java Source Editor:
 

```
MSMQClient_1.getMSMQMessage();
```



- 2 Create the **Person\_Person\_1.unmarshalFromBytes(MSMQClient\_1.MsgBody)** Business Rule:
  - A From the Business Rules toolbar click the **rule** button to add a new rule.
  - B Right-click the **Person\_Person\_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
  - C Select **unmarshalFromBytes(byte[] in)** from the method selection menu. The **unmarshalFromBytes** method box appears with an arrow connecting **Person\_Person\_1**, in the left pane of the Business Rule Designer, to the **Person** input node.
  - D From the left pane of the Business Rules Designer, map **MsgBody**, under **MSMQClient\_1**, to the in(byte[]) input node of the **unmarshalFromBytes** method box. To do this, click on **MsgBody** in the left pane of the Business Rules Designer and drag the cursor to the **in (byte[])** input node of the **unmarshalFromBytes** method box (see Figure 40).

**Figure 40** Collaboration Editor - Business Rules Designer

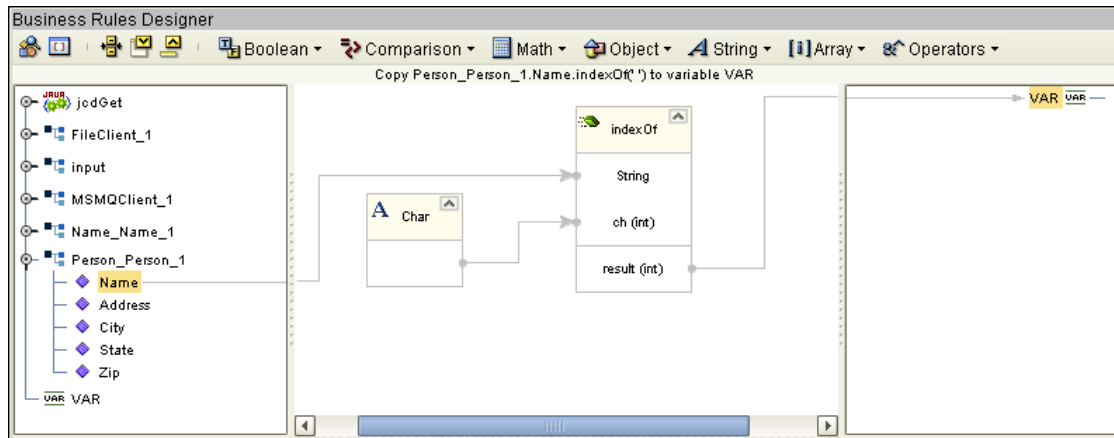


- E This creates the following Java code:
 

```
Person_Person_1.unmarshalFromBytes( MSMQClient_1.getMsgBody() );
```
- 3 Create the **Copy Person\_Person\_1.Name.indexOf(' ') to variable VAR** Business Rule:
  - A From the Business Rules toolbar, click **Local Variable**. The **Create Variable** dialog box appears.
  - B Enter **VAR** as the variable name, select **Primitive > int**, as the Type, and click **OK**.
  - C Right-click **Name**, under **Person\_Person\_1** in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
  - D Select **indexOf(int ch)** from the method selection menu. The **indexOf** method box appears with an arrow connecting **Name**, in the left pane of the Business Rule Designer, to the **String** input node.
  - E From the Business Rules Designer toolbar, select **Literal Character** from the String menu. The **Char Literal** box appears. Enter one space (press the space bar one time) as the Literal Character value.

- F Map the *space* output node of the **Char** literal box to the **ch (int)** input node of the **indexOf** method box.
- G Map the **result (int)** output node of the **indexOf** method box to **VAR** in the right pane of the Business Rules Designer (see Figure 41).

**Figure 41** Collaboration Editor - Business Rules Designer

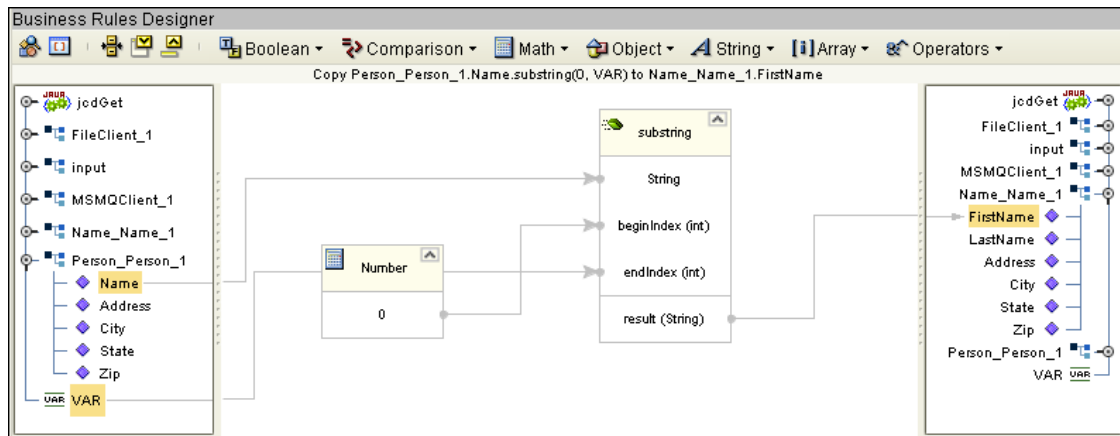


- H This creates the following Java code:

```
int VAR = Person_Person_1.getName().indexOf( ' ' );
```

- 4 Create the **Copy Person\_Person\_1.Name.substring(0, VAR) to Name\_Name\_1.FirstName** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B Right-click **Name** under **Person\_Person\_1** in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
  - C Select **substring(int beginIndex, int endIndex)** from the method selection menu. The **substring** method box appears with an arrow connecting **Name**, in the left pane of the Business Rule Designer, to the **String** input node.
  - D From the Business Rules Designer toolbar, select **Literal Number** from the Math menu. The **Number** Literal box appears with a value of **0**.
  - E Map the **0** output node of the **Number** literal box to the **beginIndex (int)** input node of the **substring** method box.
  - F From the left pane of the Business Rules Designer, map **VAR** to the **endIndex (int)** input node of the **substring** method box.
  - G Map the **result (String)** output node of the **substring** method box to **FirstName**, under **Name\_Name\_1** in the right pane of the Business Rules Designer (see [Figure 42 on page 67](#)).

**Figure 42** Collaboration Editor - Business Rules Designer



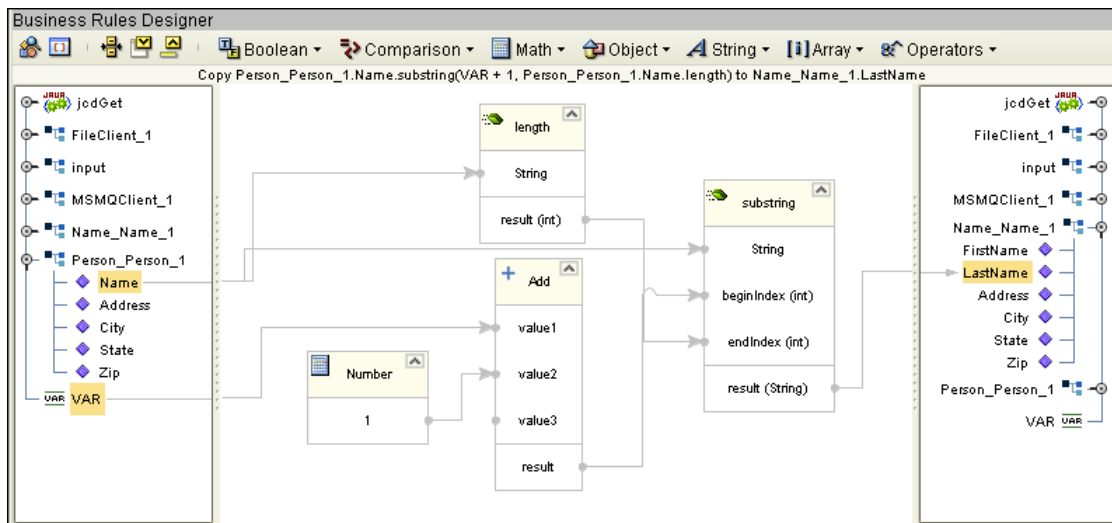
H This creates the following Java code:

```
Name_Name_1.setFirstName(
    Person_Person_1.getName().substring( 0, VAR ) );
```

5 Create the **Copy Person\_Person\_1.Name.substring(VAR + 1, Person\_Person\_1.Name.length) to Name\_Name\_1.LastName** Business Rule:

- A From the Business Rules toolbar, click the **rule** button to add a new rule.
- B Right-click **Name** under **Person\_Person\_1** in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
- C Select **substring(int beginIndex, int endIndex)** from the method selection menu. The **substring** method box appears with an arrow connecting **Name**, in the left pane of the Business Rule Designer, to the **String** input node.
- D Again, right-click **Name** under **Person\_Person\_1** in the left pane of the Business Rule Designer, click **Select method to call**, and select **length()** as the method. The **length** method box appears.
- E Right-click **VAR** in the left pane of the Business Rule Designer, click **Select method to call**, and select **Add** as the method. The **Add** method box appears.
- F From the Business Rules Designer toolbar, select **Literal Number** from the Math menu. The **Number** Literal box appears with a value of 0. Double-click the number and change the value to 1.
- G Map the 1 output node of the Number literal box to the **value2** input node of the Add method box.
- H Map the **result (int)** output node of the length box to the **endIndex (int)** input node of the substring method box.
- I Map the **result** output node of the Add method box to the **beginIndex (int)** input node of the substring method box.
- J Map the **result (String)** output node of the substring method box to **LastName** under **Name\_Name\_1** in the right pane of the Business Rules Designer (see [Figure 49 on page 77](#)).

**Figure 43** Collaboration Editor - Business Rules Designer



**6** Create the **Copy Person\_Person\_1.Address to Name\_Name\_1.Address** Business Rule:

- A** From the Business Rules toolbar, click the **rule** button to add a new rule.
- B** From the left pane of the Business Rules Designer, map **Address**, under **Person\_Person\_1**, to **Address**, under **Name\_Name\_1**, in the right pane of the Business Rules Designer (see Figure 44).

**Figure 44** Collaboration Editor - Business Rules Designer

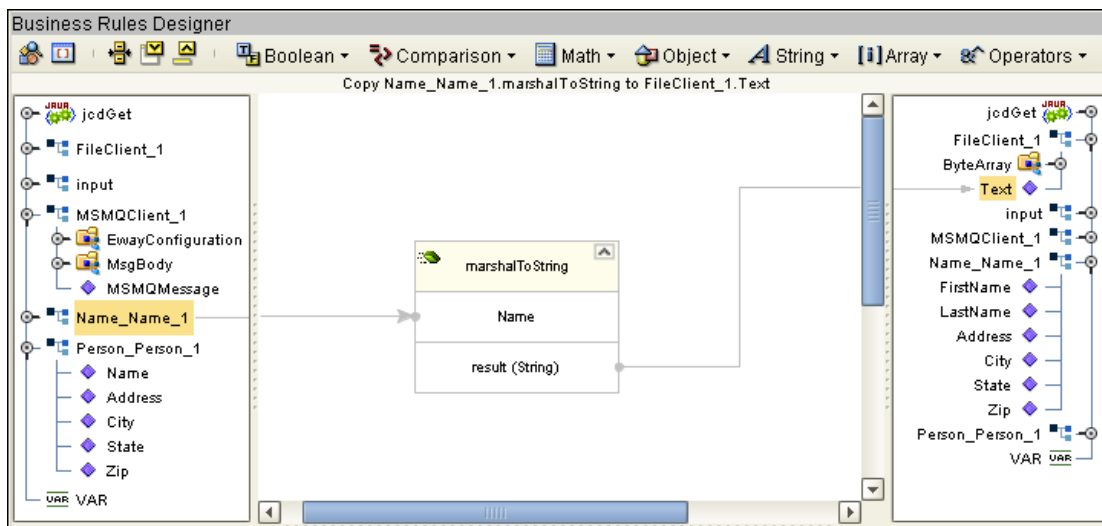


**7** Create the **Copy Person\_Person\_1.City to Name\_Name\_1.City** Business Rule:

- A** From the Business Rules toolbar, click the **rule** button to add a new rule.
- B** From the left pane of the Business Rules Designer, map **City**, under **Person\_Person\_1**, to **City**, under **Name\_Name\_1**, in the right pane of the Business Rules Designer.

- 8 Create the **Copy Person\_Person\_1.State to Name\_Name\_1.State** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the left pane of the Business Rules Designer, map **State**, under **Person\_Person\_1**, to **State** under **Name\_Name\_1** in the right pane of the Business Rules Designer.
- 9 Create the **Copy Person\_Person\_1.Zip to Name\_Name\_1.Zip** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the left pane of the Business Rules Designer, map **Zip**, under **Person\_Person\_1**, to **Zip** under **Name\_Name\_1** in the right pane of the Business Rules Designer.
- 10 Create the **Copy Name\_Name\_1.marshallToString to FileClient\_1.Text** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B Right-click **Name\_Name\_1** in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
  - C Select **marshallToString()** from the method selection menu. The **marshallToString** method box appears with an arrow connecting **Name\_Name\_1**, in the left pane of the Business Rule Designer, to the **Name** input node.
  - D Map the **result (String)** output node of the **marshallToString** method box to **Text** under **FileClient\_1** in the right pane of the Business Rules Designer (see Figure 45).

**Figure 45** Collaboration Editor - Business Rules Designer



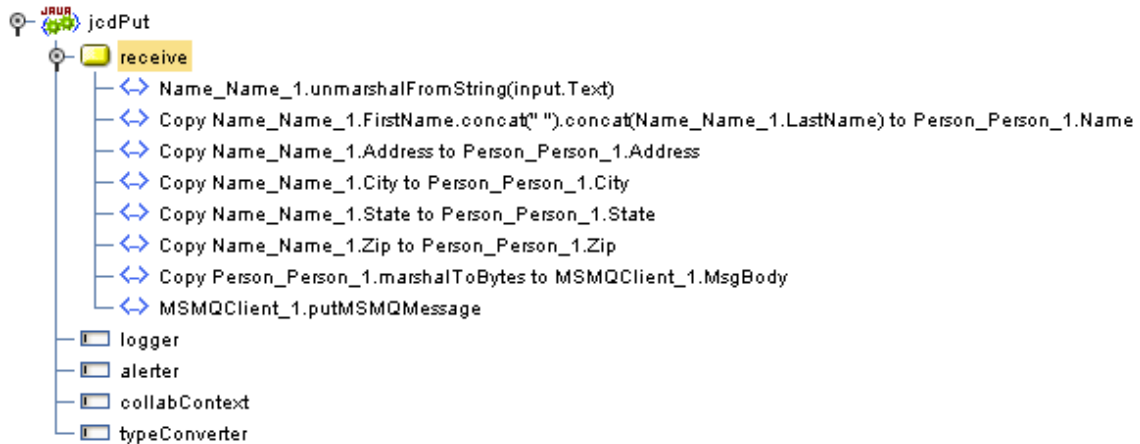
- 11 Create the **FileClient\_1.write** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B Right-click **FileClient\_1** in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.

- 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.
- 12 Save your current changes to the Repository.

## Creating the jcdPut Collaboration Business Rules

The jcdPut Collaboration contains the Business Rules displayed in Figure 46.

**Figure 46** jcdPut Collaboration Business Rules

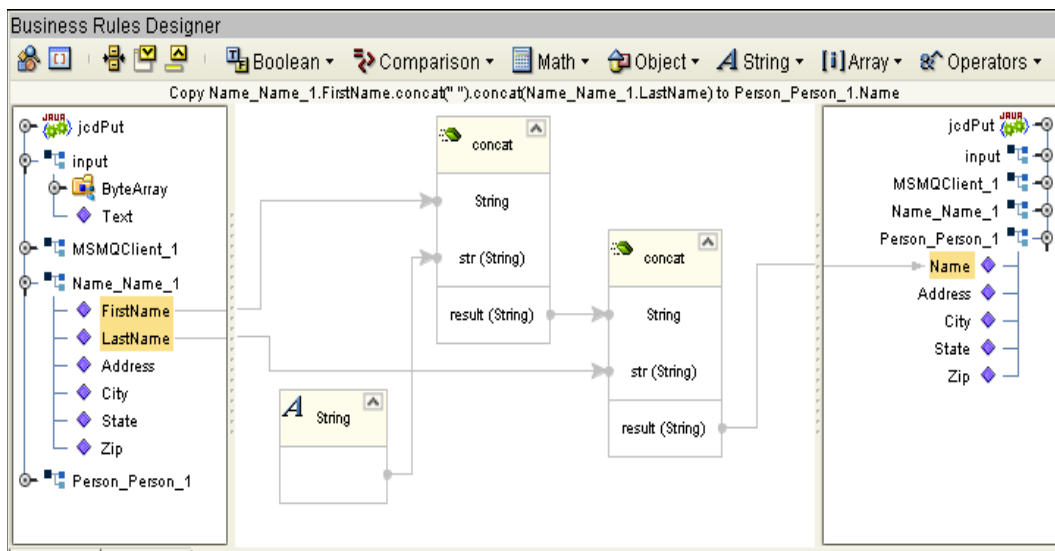


To create the jcdPut Collaboration Business Rules do the following:

- 1 Create the **Name\_Name\_1.unmarshalFromString(input.Text)** Business Rule:
  - A Right-click the **Name\_Name\_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
  - B Select **unmarshalFromString(String in)** from the method selection menu. The **unmarshalFromString** method box appears with an arrow connecting
  - C From the left pane of the Business Rules Designer, map **Text**, under the **input** node, to the **in (String)** input node of the **unmarshalFromString** method box.
  - D Map the **result (String)** of the **String** method box to **Text**, under **FileClient\_1** in the right pane of the Business Rule Designer.
- 2 Create the **Copy Name\_Name\_1.FirstName.concat(" ").concat (Name\_Name\_1.LastName) to Person\_Person\_1.Name** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the Business Rules Designer's String menu, select **concat**. The **concat** method box appears. Repeat this step to create two **concat** method boxes on the Business Rules Designer canvas.
  - C From the Business Rules Designer's String menu, select **Literal String**. The **String** literal box appears. Enter one space (press space bar one time) as the literal String value.

- D From the left pane of the Business Rules Designer, map **FirstName**, under **Name\_Name\_1**, to the **String** input node of the first **concat** method box.
- E Map the space output node of the **String** Literal box to the **str (String)** input node of the first **concat** method box.
- F Map the **result (String)** output node of the first **concat** method box to the **String** input node of the second **concat** method box.
- G From the left pane of the Business Rules Designer, map **LastName**, under **Name\_Name\_1**, to the **str (String)** input node of the second **concat** method box.
- H Map the **result (String)** output node of the second **concat** method box to **Name**, under the **Person\_Person\_1** node in the right pane of the Business Rules Designer (see Figure 47).

**Figure 47** Collaboration Editor - Business Rules Designer



- 3 Create the **Copy Name\_Name\_1.Address to Person\_Person\_1.Address** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the left pane of the Business Rules Designer, map **Address**, under **Name\_Name\_1**, to **Address**, under **Person\_Person\_1** in the right pane of the Business Rules Designer.
- 4 Create the **Copy Name\_Name\_1.City to Person\_Person\_1.City** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the left pane of the Business Rules Designer, map **City**, under **Name\_Name\_1** to **City**, under **Person\_Person\_1** in the right pane of the Business Rules Designer.
- 5 Create the **Copy Name\_Name\_1.State to Person\_Person\_1.State** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.





## Creating the jcdReceive Collaboration Business Rules

The jcdReceive Collaboration contains the Business Rules displayed in Figure 48.

**Figure 48** jcdReceive Collaboration Business Rules



To create the jcdReceive Collaboration Business Rules do the following:

- 1 Create the **If condition: input has Payload** if/then statement:
  - A From the Business Rules tree, right-click the **receiveMSMQMessage** node, and select **if/then** from the shortcut menu.
  - B Right-click the **input** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
  - C Select **hasPayload()** from the method selection menu. The **hasPayload** method box appears with an arrow connecting the **input** node in the left pane of the Business Rule Designer to the **payloadInput** input node.
  - D Map the **result (boolean)** output node of the **String** method box to the **condition** node in the right pane of the Business Rule Designer.
- 2 Create the **Person\_Person\_1.unmarshalFromBytes(input.Payload)** Business Rule under the **then** statement:
  - A From the Business Rules pane of the Collaboration Editor, select the **then** statement in the Business Rules tree, and from the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the Business Rules Designer's String menu, select **concat**. The **concat** method box appears. Repeat this step to create two **concat** method boxes on the Business Rules Designer canvas.
  - C From the left pane of the Business Rules Designer, right-click the **Person\_Person\_1**, under the **input** node, and click **Select method to call**. The method selection menu appears.
  - D Select **unmarshalFromBytes(byte[] in)** from the method selection menu. The **unmarshalFromBytes** method box appears with an arrow connecting the

- Person\_Person\_1** node in the left pane of the Business Rule Designer to the **Person** input node.
- E From the left pane of the Business Rules Designer, map **Payload** to the **in (byte[]) input** node of the **unmarshalFromBytes** method box.
- 3 Create the **Copy Person\_Person\_1.Name.indexOf(' ') to variable VAR** Business Rule:
- A From the Business Rules toolbar, click **Local Variable**. The **Create Variable** dialog box appears.
  - B Enter **VAR** as the variable name, select **Primitive > int**, as the Type, and click **OK**.
  - C From the left pane of the Business Rules Designer, right-click **Name**, under **Person\_Person\_1**, and click **Select method to call**. The method selection menu appears.
  - D Select **indexOf(int ch)** from the method selection menu. The **indexOf** method box appears with an arrow connecting **Name**, in the left pane of the Business Rule Designer, to the **String** input node.
  - E From the Business Rules Designer toolbar, select **Literal Character** from the String menu. The **Char** Literal box appears. Enter one space (press the space bar one time) as the Literal Character value.
  - F Map the *space* output node of the **Char** literal box to the **ch (int)** input node of the **indexOf** method box.
  - G Map the **result (int)** output node of the **indexOf** method box to **VAR** in the right pane of the Business Rules Designer.
- 4 Create the **Copy Person\_Person\_1.Name.substring(0, VAR) to Name\_Name\_1.FirstName** Business Rule:
- A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the left pane of the Business Rules Designer, right-click **Name** under **Person\_Person\_1**, and click **Select method to call**. The method selection menu appears.
  - C Select **substring(int beginIndex, int endIndex)** from the method selection menu. The **substring** method box appears with an arrow connecting **Name**, in the left pane of the Business Rule Designer, to the **String** input node.
  - D From the Business Rules Designer toolbar, select **Literal Number** from the Math menu. The **Number** Literal box appears with a value of **0**.
  - E Map the **0** output node of the **Number** literal box to the **beginIndex (int)** input node of the **substring** method box.
  - F From the left pane of the Business Rules Designer, map **VAR**, to the **endIndex (int)** input node of the **substring** method box.
  - G Map the **result (String)** output node of the substring method box to **FirstName**, under **Name\_Name\_1** in the right pane of the Business Rules Designer.

- 5 Create the **Copy Person\_Person\_1.Name.substring(VAR + 1, Person\_Person\_1.Name.length) to Name\_Name\_1.LastName** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the left pane of the Business Rules Designer, right-click **Name** under **Person\_Person\_1**, and click **Select method to call**. The method selection menu appears.
  - C Select **substring(int beginIndex, int endIndex)** from the method selection menu. The **substring** method box appears with an arrow connecting **Name**, in the left pane of the Business Rule Designer, to the **String** input node.
  - D Again, right-click **Name** under **Person\_Person\_1**, click **Select method to call**, and select **length()** as the method. The **length** method box appears.
  - E From the left pane of the Business Rules Designer, right-click **VAR**, click **Select method to call**, and select **Add** as the method. The **Add** method box appears.
  - F From the Business Rules Designer toolbar, select **Literal Number** from the Math menu. The **Number** Literal box appears with a value of **0**. Double-click the number and change the value to **1**.
  - G Map the **1** output node of the **Number** literal box to the **value2** input node of the **Add** method box.
  - H Map the **result (int)** output node of the **length** box to the **endIndex (int)** input node of the **substring** method box.
  - I Map the **result** output node of the **Add** method box to the **beginIndex (int)** input node of the **substring** method box.
  - J Map the **result (String)** output node of the **substring** method box to **LastName**, under **Name\_Name\_1** in the right pane of the Business Rules Designer
- 6 Create the **Copy Person\_Person\_1.Address to Name\_Name\_1.Address** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the left pane of the Business Rules Designer, map **Address** under **Person\_Person\_1**, to **Address**, under **Name\_Name\_1** in the right pane of the Business Rules Designer.
- 7 Create the **Copy Person\_Person\_1.City to Name\_Name\_1.City** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the left pane of the Business Rules Designer, map **City** under **Person\_Person\_1**, to **City**, under **Name\_Name\_1** in the right pane of the Business Rules Designer.
- 8 Create the **Copy Person\_Person\_1.State to Name\_Name\_1.State** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the left pane of the Business Rules Designer, map **State**, under **Person\_Person\_1**, to **State**, under **Name\_Name\_1** in the right pane of the Business Rules Designer.

- 9 Create the **Copy Person\_Person\_1.Zip to Name\_Name\_1.Zip** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the left pane of the Business Rules Designer, map **Zip** under **Person\_Person\_1**, to **Zip**, under **Name\_Name\_1** in the right pane of the Business Rules Designer.
- 10 Create the **Copy Name\_Name\_1.marshallToString to FileClient\_1.Text** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the left pane of the Business Rules Designer, right-click **Name\_Name\_1**, and click **Select method to call**. The method selection menu appears.
  - C Select **marshallToString()** from the method selection menu. The **marshallToString** method box appears with an arrow connecting **Name\_Name\_1**, in the left pane of the Business Rule Designer, to the **Name** input node.
  - D Map the **result (String)** output node of the **marshallToString** method box to **Text** under **FileClient\_1** in the right pane of the Business Rules Designer.
- 11 Create the **FileClient\_1.write** Business Rule:
  - A From the Business Rules toolbar, click the **rule** button to add a new rule.
  - B From the left pane of the Business Rules Designer, right-click **FileClient\_1**, and click **Select 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.
- 12 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*.

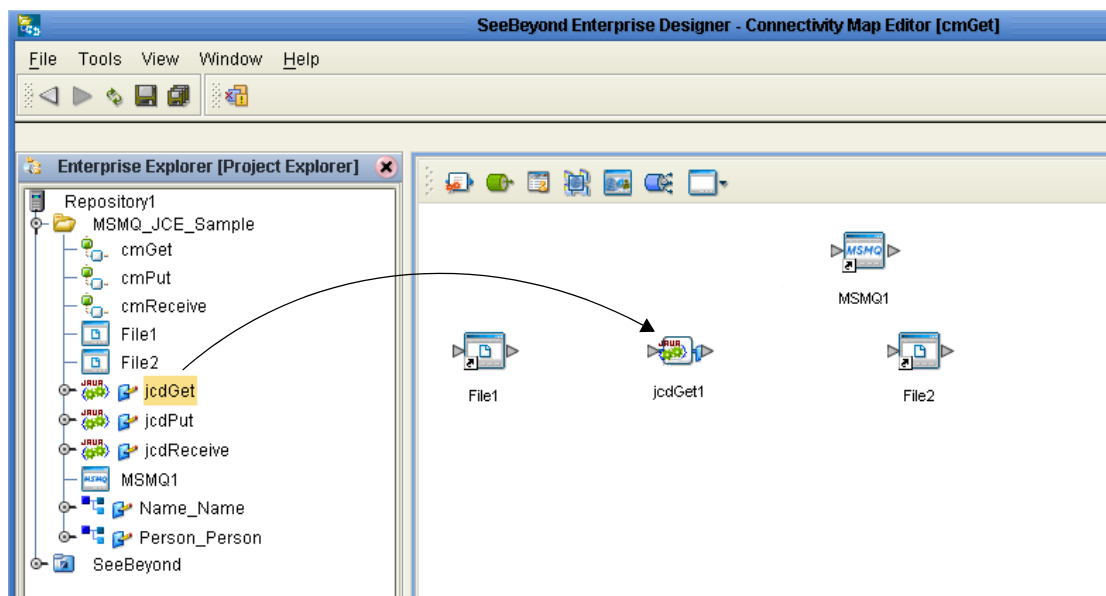
## 5.4.6 Binding the eWay Components

After you have created the Collaborations, associate the components in the Connectivity Map.

### Binding the components of the cmGet Connectivity Map

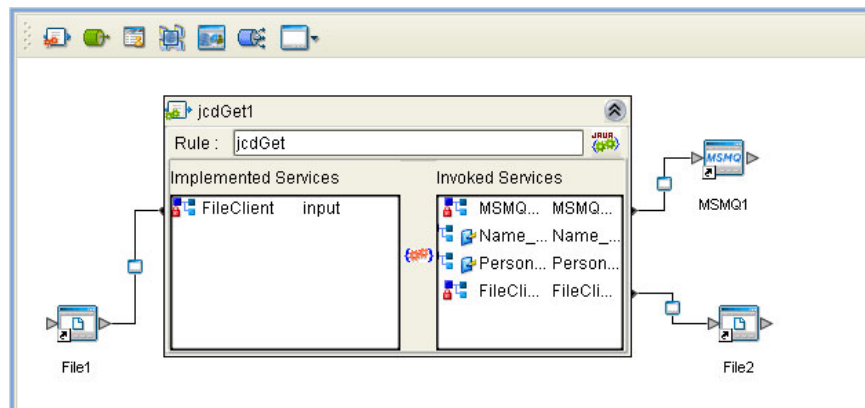
- 1 From the Project Explorer, double-click **cmGet**. The **cmGet Connectivity Map** appears.
- 2 Drag and drop the **jcdGet** Collaboration from the Project Explorer tree onto the **jcdGet1** service. If the Collaboration was successfully associated, the Service's "gears" icon changes from red to green (see Figure 49).

**Figure 49** Connectivity Map - Binding the Collaborations



- 3 Double-click the **jcdGet1** service. The **jcdGet1 Binding** dialog box appears, using **jcdGet** as the rule.
- 4 From the **jcdGet1** Binding dialog box, drag **FileClient input** (Implemented Services) to the **File1** External Application.
- 5 From the **jcdGet1** Binding dialog box, drag **MSMQClient\_1, MSMQClient** (under Invoked Services) to the **MSMQ1** External Application.
- 6 From the **jcdGet1** Binding dialog box, drag **FileClient\_1, FileClient** (under Invoked Services) to the **File2** External Application (see [Figure 50 on page 78](#)).

**Figure 50** Connectivity Map - Binding the Components



- 7 Click the chevrons in the upper-right corner of the **jcdGet** Binding dialog box to minimize the **jcdGet**.
- 8 Save your current changes to the Repository.

### Binding the components of the cmPut Connectivity Map

- 1 From the Project Explorer, double-click **cmPut**. The **cmPut Connectivity Map** appears.
- 2 Drag and drop the **jcdPut** Collaboration from the Project Explorer tree onto the **jcdPut1** service.
- 3 Double-click the **jcdPut1** service. The **jcdPut1** Binding dialog box appears, using **jcdPut** as the rule.
- 4 From the **jcdPut1** Binding dialog box, drag **FileClient input** (Implemented Services) to the **File1** External Application.
- 5 From the **jcdPut1** Binding dialog box, drag **MSMQClient\_1, MSMQClient** (under Invoked Services) to the **MSMQ1** External Application.
- 6 Save your current changes to the Repository.

### Binding the components of the cmPut Connectivity Map

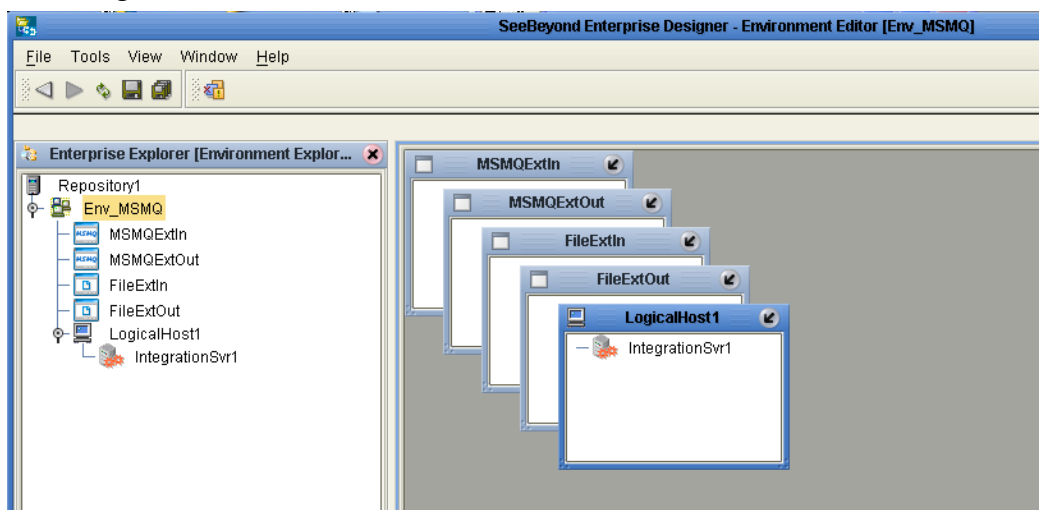
- 1 From the Project Explorer, double-click **cmReceive**. The **cmReceive Connectivity Map** appears.
- 2 Drag and drop the **jcdReceive** Collaboration from the Project Explorer tree onto the **jcdReceive1** service.
- 3 Double-click the **jcdReceive1** service. The **jcdReceive1** Binding dialog box appears using **jcdReceive** as the rule.
- 4 From the **jcdReceive1** Binding dialog box, drag **MSMQClient input** (Implemented Services) to the **MSMQ1** External Application.
- 5 From the **jcdReceive1** Binding dialog box, drag **FileClient\_1, FileClient** (under Invoked Services) to the **File1** External Application.
- 6 Save your current changes to the Repository.

## 5.4.7 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. You create Environments using the Enterprise Designer's Environment Editor. To create the MSMQ sample Environment, do the following:

- 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. Rename the new Environment to **Env\_MSMQ**.
- 3 Right-click **Env\_MSMQ** and select **New MSMQ External System**. Name the External System **MSMQExtIn** and select **Inbound MSMQ eWay** as the type. Click **OK**.
- 4 Right-click **Env\_MSMQ** and select **New MSMQ External System**. Name the External System **MSMQExtOut** and select **Outbound MSMQ eWay** as the type. Click **OK**.
- 5 Right-click **Env\_MSMQ** and select **New File External System**. In the Create an External System dialog box, name the External System **FileExtIn**, and select **Inbound File eWay** as the type. Click **OK**.
- 6 Right-click **Env\_MSMQ** and select **New File External System**. In the Create an External System dialog box, name the External System **FileExtOut**, and select **Outbound File eWay** as the type. Click **OK**.
- 7 Right-click **Env\_MSMQ** and select **New Logical Host**. The **LogicalHost1** box is added to the Environment and **LogicalHost1** is added to the Environment Editor tree.
- 8 From the Environment Explorer tree, right-click the **LogicalHost1**, and select **New SeeBeyond Integration Server**. A new Integration Server, **IntegrationSvr1**, is added to the **LogicalHost1** box (see Figure 51).

**Figure 51** Environment Editor - EnvMSMQ Environment



## 5.4.8 Configuring the eWays

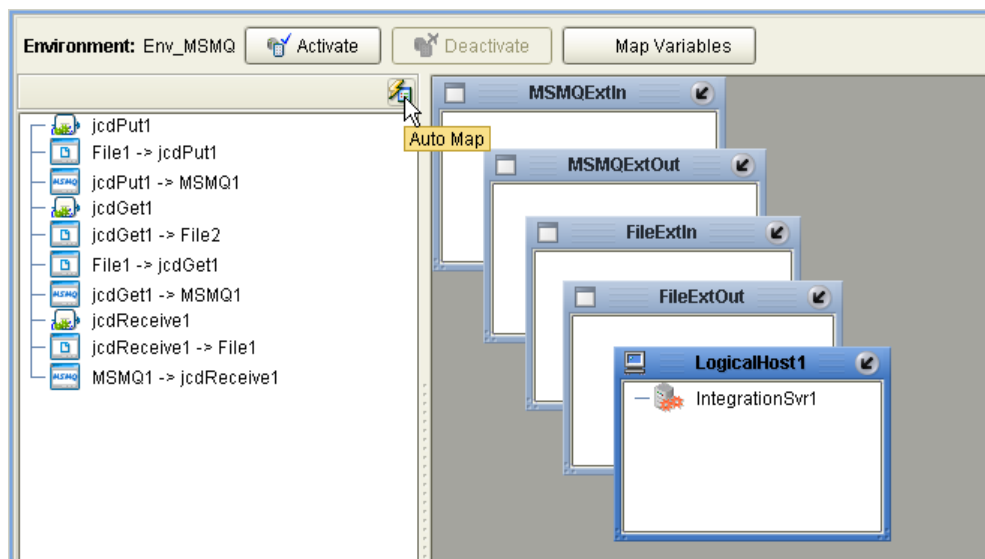
Configure the eWay properties for your system, including the settings for the input and output directories. For the most part, the eWay settings can be left with their default settings. See [“Configuring the MSMQ eWay” on page 16](#) for more information on configuring the eWays for your system.

## 5.4.9 Creating and Activating the Deployment Profile

Deployment Profiles assign Collaborations and Message Destinations to the Integration Server and Message Server. You create a Deployment Profiles using the Deployment Editor. To create the sample Deployment Profile, do the following.

- 1 From the Enterprise Explorer’s Project Explorer, right-click the Project (**MSMQ\_JCD\_Sample**) and select **New > Deployment Profile**.
- 2 Enter a name for the Deployment Profile (for this sample, enter **DP\_MSMQ**). Make sure that the selected Environment is **Env\_MSMQ**. Click **OK**.
- 3 From the Deployment Editor toolbar, click the **Auto Map** icon. The Project’s components are automatically mapped to their system windows. If any of the Project components are not mapped automatically after Auto Map is used, those components can be mapped manually by following the appropriate steps below. Once all components are mapped, proceed to step 14.

**Figure 52** Deployment Profile - Auto Map

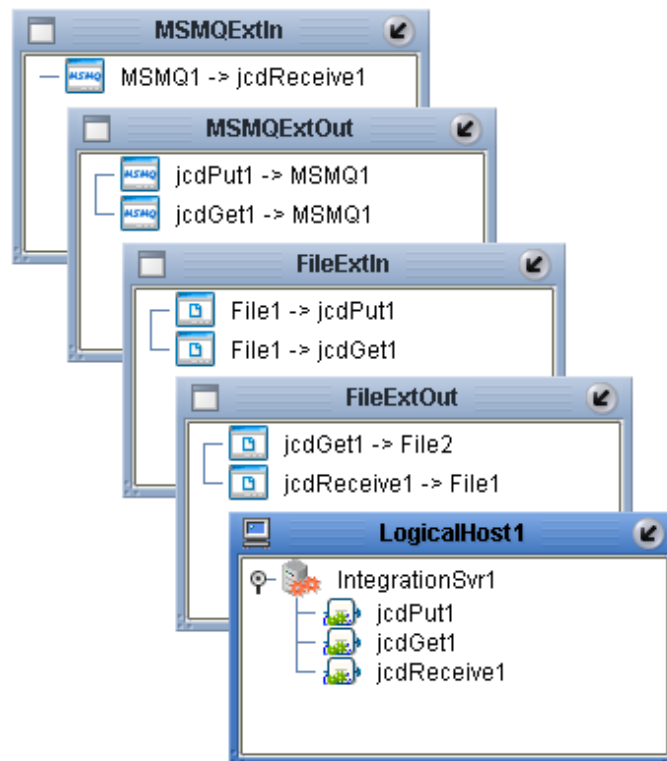


- 4 From the left pane of the Deployment Editor, drag **MSMQ1 -> jcdReceive1** (External Application) to the **MSMQExtIn** box.
- 5 Drag **jcdPut1 -> MSMQ1** (External Application) to the **MSMQExtOut** box.
- 6 Drag **jcdGet1 -> MSMQ1** (External Application) to the **MSMQExtOut** box.
- 7 Drag **File1 -> jcdPut1** (External Application) to the **FileExtIn** box.
- 8 Drag **File1 -> jcdGet1** (External Application) to the **FileExtIn** box.



- 9 Drag **jcdGet1** -> **File2** (External Application) to the **FileExtOut** box.
- 10 Drag **jcdReceive1** -> **File1** (External Application) to the **FileExtOut** box.
- 11 Drag **jcdPut1** service to **IntegrationSvr1** in the Logical Host (**LogicalHost1**) Environment box.
- 12 Drag **jcdGet1** service to **IntegrationSvr1** in the Logical Host (**LogicalHost1**) Environment box.
- 13 Drag **jcdReceive1** service to **IntegrationSvr1** in the Logical Host (**LogicalHost1**) Environment box (see Figure 53).

**Figure 53** Deployment Profile



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

---

## 5.5 Running a Project

The following procedure assumes that you downloaded the Enterprise Designer to **C:\ican50**. If this is not the case, replace **C:\ican50** in the following procedure with the appropriate location.

- 1 From the Enterprise Manager Downloads tab, download **Logical Host - for win32**.
- 2 Extract the file to the **ican50** directory.
- 3 Navigate to **C:\ican50\logicalhost\bootstrap\config** directory and open the **logical-host.properties** file using a text editor.
- 4 Enter the following information in the appropriate fields:
  - ♦ Logical Host root directory: **C:\ican50\logicalhost**
  - ♦ Repository URL: **http://localhost:port number/repository name**
  - ♦ Repository user name and password: *Your user name and password*
  - ♦ Logical Host Environment name: For example, **Env\_MSMQ**
  - ♦ Logical Host name: **LogicalHost1**

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

- 5 Run the **bootstrap.bat** file in the **ican50\logicalhost\bootstrap\bin** directory.
- 6 To run the **Get** operation, copy the sample **Trigger** file to the input folder with the appropriate extension.
- 7 To run the **Put** operation, copy the sample **Name** file to the input directory with the appropriate extension.

# Index

## A

About 7  
 ACTION\_PEEK\_CURRENT 25  
 ACTION\_RECEIVE 25  
 alerting 32  
 Auto Map 53, 80

## B

bootstrap 55, 82  
 building a project 56  
 Business Rules  
   Collaboration Editor (Java) 64

## C

Collaboration Definitions  
   creating 62  
 Collaboration Editor (Java) 64  
 configuring the eWay properties 16  
 configuring the MSMQ eWay 16  
 Connectivity Map 47, 58  
   populating 48, 59

## D

DENY\_NONE 22, 26  
 DENY\_RECEIVE\_SHARE 22, 26  
 Deployment Profile 80  
   activation 80  
   Auto Map 53, 80  
   creating 80  
 DLL files  
   adding 15  
 document  
   conventions 9  
 dynamic configuration 31

## E

eInsight  
   engine and components 33  
   using with the WebSphere MQ eWay 33  
 Environment 51, 79

  creating 51, 79  
 eWay  
   configuring the eWays 52, 80  
 eWay components 56  
 External Application  
   selecting 16, 17  
 External Applications  
   selecting 48, 58

## I

implementation 56  
 importing a project 57  
 installing  
   COM/DCOM eWay 12

## L

logging 32

## M

manual configuration 31

## O

operating systems  
   supported 11  
 organization of information 8  
 overview  
   MSMQ 7

## P

PEEK\_ACCESS 24  
 project  
   creating 35, 58  
   importing 57  
 properties  
   inbound Connectivity Map 20  
   inbound Environment 27  
   Message Priority  
     outbound 23  
   MSMQ Access Mode  
     inbound 20  
     outbound 24  
   MSMQ Connection Mode  
     outbound 28  
   MSMQ Format Name  
     inbound 21  
   MSMQ Host Name  
     inbound 27  
     outbound 29

## Index

- MSMQ Queue Alias
  - inbound 21
  - outbound 25
- MSMQ Queue Name
  - inbound 21
  - outbound 25
- MSMQ Receive Action Code
  - outbound 25
- MSMQ Receive Interval
  - inbound 22
- MSMQ Share Mode
  - inbound 22
  - outbound 26
- outbound Environment 28
- properties editor 18

## R

- RECEIVE\_ACCESS 24
- rollback
  - enabling for outbound messages 30
- running a project 55, 82
- Runtime Java API
  - adding 14
- Runtime win32 bridge DLLs
  - adding 15

## S

- sample eWay project
  - importing a project 57
- sample project
  - running 55, 82
- SEND\_ACCESS 24
- system requirements 11
  - external 12

## W

- WebSphere MQ Queue 57
  - creating 57
- writing conventions 9