

SeeBeyond ICAN Suite

TigerLogic eWay Intelligent Adapter User's Guide

Release 5.0.0



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 20050613145812.

Contents

Chapter 1

Introducing the TigerLogic eWay	6
About TigerLogic	6
About the TigerLogic eWay Intelligent Adapter	6
About This Document	7
Organization of Information in This Book	7
TigerLogic Javadoc	7
Scope of the Document	7
Intended Audience	7
Document Conventions	8
Screenshots	8
Related Documents	8
SeeBeyond Web Site	8
SeeBeyond Documentation Feedback	9

Chapter 2

Installing the TigerLogic eWay	10
Supported Operating Systems	10
System Requirements	10
External System Requirements	11
Before You Install	11
Installing the eWay Product Files	11
Installing the TigerLogic eWay	11
Copying JAR files to the eDesigner Library	12
Uploading JAR files to the Logical Host	12
Adding the eWay to an Existing ICAN Suite Installation	13
After You Install	14

Chapter 3

Configuring the TigerLogic eWay	15
Setting the TigerLogic eWay Properties	15
Outbound Properties in the Connectivity Map	15
TigerLogic Connection Mode	16
Setting the TigerLogic eWay Properties in the Environment	16
Outbound Properties in the Environment	17
Client Listener Port Number	17
Password	17
Server IP Address	17
Server PortNumber	18
UserName	18
Manually Configuring the eWay	18
Alerting and Logging	20

Chapter 4

Working with the Sample Project	21
Sample Project Overview	21
eInsight Engine and eGate Components	21
Importing the Sample Project in eGate	22
Before Running the Sample Project	23
Setting the eWay Properties	23
Creating the Environment Profile	23
Creating and Activating the Deployment Profile	24
Working with the Sample Project in eGate	25
Sample Project Inputs	25
Sample Project Insert Operation	26
Insert Operation Connectivity Map	26
jcdInsert Collaboration Definition (Java)	26
jcdInsert Collaboration Business Rules	26
Sample Project Replace Operation	30
Replace Operation Connectivity Map	30
jcdReplace Collaboration Definition (Java)	31
jcdReplace Collaboration Business Rules	31
Sample Project Delete Operation	34
Delete Operation Connectivity Map	34
jcdDelete Collaboration Definition (Java)	35
jcdDelete Collaboration Business Rules	35
Sample Project Query Operations	37
Query Operations Connectivity Maps	37
jcdQuery Collaboration Definitions (Java)	38
jcdQuery1 Collaboration Business Rules	38
jcdQuery2 Collaboration Business Rules	39

Contents

Running the Sample Project 41

Index 47

Chapter 1

Introducing the TigerLogic eWay

Welcome to the *TigerLogic eWay Intelligent Adapter User's Guide*. This document includes information about installing, configuring, and using the SeeBeyond® Integrated Composite Application Network Suite™ (ICAN) TigerLogic eWay Intelligent Adapter, referred to as the TigerLogic eWay throughout this guide.

What's in This Chapter

- [About TigerLogic](#) on page 6
- [About the TigerLogic eWay Intelligent Adapter](#) on page 6
- [About This Document](#) on page 7
- [Related Documents](#) on page 8
- [SeeBeyond Web Site](#) on page 8
- [SeeBeyond Documentation Feedback](#) on page 9

1.1 About TigerLogic

TigerLogic XML Data Management Server (TigerLogic XDMS) is an effective and highly powerful management and query tool of XML-format data which utilizes the Pick® Universal Data Model (Pick UDM). TigerLogic contains robust query, indexing, and profiling technologies across relational XML databases, repositories, or search engines. TigerLogic provides the distinct advantage of XML, Java, WSDL, and SOAP compatibility in a number of development scenarios. TigerLogic XDMS also offers increased flexibility in processing data schema and structures, and support for schema versioning.

1.2 About the TigerLogic eWay Intelligent Adapter

The TigerLogic eWay allows the ICAN system to exchange data with an a TigerLogic XML database. The TigerLogic eWay transparently integrates existing data files with XML databases.

The eWay enables ICAN to use business logic within Collaborations or Business Processes to perform data identification, manipulation, and transformation. Queues or

Topics provide non-volatile storage for data within ICAN, permitting applications to run independently of one another at different speeds and times. This user's guide describes how to install and configure the TigerLogic eWay.

1.3 About This Document

1.3.1 Organization of Information in This Book

This document includes the following chapters:

- **Chapter 1 "Introducing the TigerLogic eWay"**: Provides an overview description of the product as well as high-level information about this document.
- **Chapter 2 "Installing the TigerLogic eWay"**: Describes the system requirements and provides instructions for installing the TigerLogic eWay.
- **Chapter 3 "Configuring the TigerLogic eWay"**: Provides instructions for configuring the eWay to communicate with your TigerLogic database.
- **Chapter 4 "Working with the Sample Project"**: Provides instructions for installing and running the Sample Project.

TigerLogic Javadoc

A TigerLogic eWay Javadoc is also provided, which documents the available Java methods provided with the TigerLogic eWay. The Javadoc is uploaded with the eWay's documentation file (**TigerLogicWayDocs.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.

1.3.2 Scope of the Document

This user's guide provides a description of the TigerLogic eWay. It includes directions for installing the eWay, configuring the eWay properties, and implementing the eWay's sample projects.

This document does not include specific details about the installation and configuration of any prerequisite ICAN products. For detailed information on these products, refer to those products' user's guides and Javadocs.

1.3.3 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.

1.3.4 Document Conventions

The following conventions are observed throughout this document.

Table 1 Document Conventions

Text	Convention	Example
Names of buttons, files, icons, parameters, variables, methods, menus, and objects	Bold text	<ul style="list-style-type: none"> ▪ Click OK to save and close. ▪ From the File menu, select Exit. ▪ Select the logicalhost.exe file. ▪ Enter the timeout value. ▪ Use the getClassname() method. ▪ Configure the Inbound File eWay.
Command line arguments, code samples	Fixed font. Variables are shown in <i>bold italic</i> .	bootstrap -p <i>password</i>
Hypertext links	Blue text	See Document Conventions on page 8
Hypertext links for Web addresses (URLs) or email addresses	Blue underlined text	http://www.seebeyond.com docfeedback@seebeyond.com

1.3.5 Screenshots

Depending on what products you have installed, and how they are configured, the screenshots in this document may differ from what you see on your system.

1.4 Related Documents

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

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

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

Chapter 2

Installing the TigerLogic eWay

What's in This Chapter

- [Supported Operating Systems](#) on page 10
- [System Requirements](#) on page 10
- [External System Requirements](#) on page 11
- [Before You Install](#) on page 11
- [Installing the eWay Product Files](#) on page 11
- [Adding the eWay to an Existing ICAN Suite Installation](#) on page 13
- [After You Install](#) on page 14

2.1 Supported Operating Systems

The TigerLogic eWay is available on the following operating systems:

- Windows 2000, Windows XP, and Windows Server 2003
- HP-UX 11.0, 11i (PA-RISC), and 11i v2.0 (11.23)
- IBM AIX 5.1L and 5.2
- Red Hat Enterprise Linux AS 2.1 (Intel x86)
- Sun Solaris 8 and 9

2.2 System Requirements

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

Although the TigerLogic 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 TigerLogic eWay **Readme**, accessed from the Enterprise Manager's **Documentation** tab.

2.3 External System Requirements

The TigerLogic eWay supports the following external system:

- TigerLogic 2.1

For full information on the requirements for this product, see the product's documentation.

2.4 Before You Install

Open and review the **Readme.txt** file for the TigerLogic eWay for any additional information or requirements prior to installation. The **Readme.txt** file is located on the installation CD-ROM.

2.5 Installing the eWay Product Files

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 TigerLogic eWay also requires additional JAR files that are included with the installation. For directions on how to install these additional files, see [“Copying JAR files to the eDesigner Library” on page 12](#).

2.5.1 Installing the TigerLogic eWay

The installation process includes installing the following operations:

- Installing the ICAN Repository
- Uploading products to the Repository (including the TigerLogic eWay, documentation, sample files, and Javadocs)
- Downloading components (including the Enterprise Designer and Logical Host) from the Repository
- Updating products in the Enterprise Designer using the Update Center Wizard

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:

- ♦ **TigerLogicWay.sar** (to install the TigerLogic eWay)
- ♦ **FileeWay.sar** (to install the File eWay, used in the Sample Project)

Note: *These files may not be located on the same installation CD as the eGate.sar file.*

- 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:
 - ♦ **TigerLogicWayDocs.sar** (to upload the TigerLogic 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*.

2.5.2 Copying JAR files to the eDesigner Library

In order to create a TigerLogic Project, you must add six JAR files to the eDesigner library folder. After you have created an **Environment** and added a **Logical Host** for your Project (see "**Before Running the Sample Project**" on page 23), copy the JAR files to the eDesigner library folder:

- 1 Locate the following JAR files in TigerLogic's TLJDK library:
 - ♦ `connector.jar`
 - ♦ `jca-connector.jar`
 - ♦ `tlapi.jar`
 - ♦ `tlerror.jar`
 - ♦ `utility.jar`
 - ♦ `xmlparser.jar`
- 2 Copy the above listed JAR files to the eDesigner library folder at at `../eDesigner/lib`.

2.5.3 Uploading JAR files to the Logical Host

You must add six JAR files to the path for the Logical Host process. 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 TigerLogic's TLJDK library where the JAR files are located. They are:
 - ♦ `connector.jar`

- ♦ jca-connector.jar
- ♦ tlapi.jar
- ♦ tterror.jar
- ♦ utility.jar
- ♦ xmlparser.jar

- 3 Add the six JAR files to the **Third Party Files** field (see figure below). Click **OK**.

Figure 1 Upload JAR files



The JAR files may also be copied directly to the Logical Host library folder at **..logicalhost/stcis/lib**.

2.6 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, 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.

2.7 After You Install

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

The eWay's User's Guide, Javadoc, Readme, and Sample Project can be accessed from the Enterprise Manager's **Documentation** tab.

Chapter 3

Configuring the TigerLogic eWay

This chapter describes how to set the properties of the TigerLogic eWay.

What's in This Chapter

- [Setting the TigerLogic eWay Properties](#) on page 15
- [Setting the TigerLogic eWay Properties in the Environment](#) on page 16
- [Manually Configuring the eWay](#) on page 18
- [Alerting and Logging](#) on page 20

3.1 Setting the TigerLogic eWay Properties

All eWays contain a unique set of default configuration parameters. After the eWays are established and a TigerLogic External System is created in the Project's Environment, the eWay parameters are modified for your specific system. The TigerLogic eWay configuration parameters can be modified from two locations:

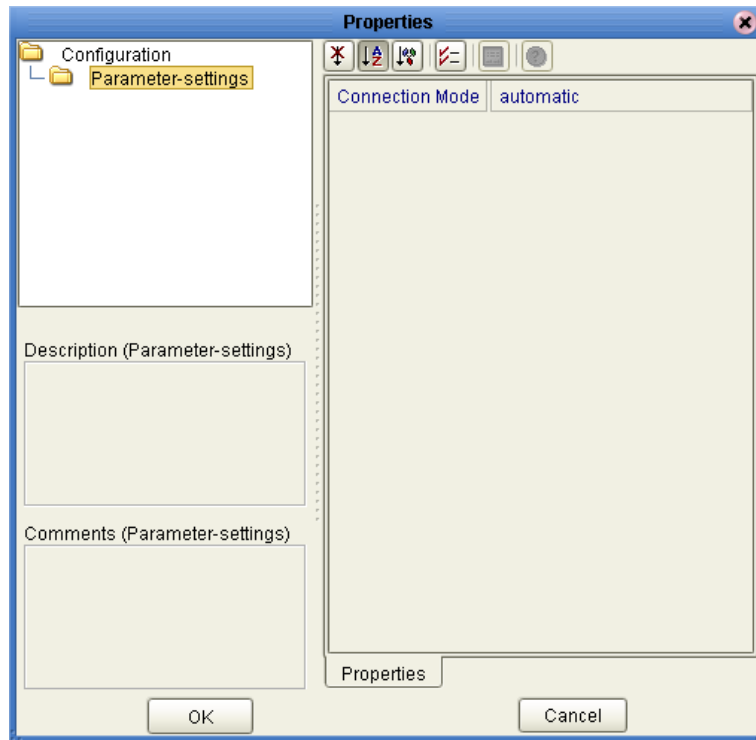
- From the **Connectivity Map**—which contains parameters specific to the TigerLogic eWay, and may vary from other eWays (of the same type) in the Project.
- From the **Environment Explorer** tree—which contains global parameters that commonly apply to all eWays (of the same type) in the Project. Saved parameters are shared by all eWays in the TigerLogic External System Properties window.

Note: You must set configuration parameters for the TigerLogic eWay in both locations.

3.1.1 Outbound Properties in the Connectivity Map

The outbound TigerLogic eWay properties in the Connectivity Map contain the top level settings displayed in Figure 2.

Figure 2 The eWay Properties



TigerLogic Connection Mode

Description

Specifies the TigerLogic connection mode. The options are:

- **Automatic:**
- **Manual (Dynamic):** Manual mode allows you to change the eWay configuration exposed in the Java Collaboration Definition. For more information see [“Manually Configuring the eWay” on page 18](#)

Required Value

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

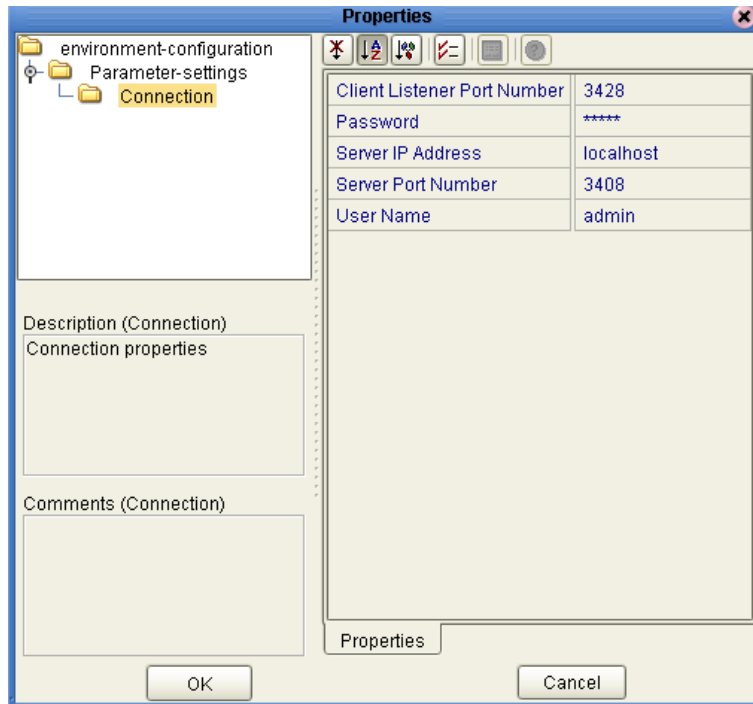
3.2 Setting the TigerLogic eWay Properties in the Environment

In the **Properties** window, using the descriptions below, enter the information necessary for the eWay to communicate with the Environment.

3.2.1 Outbound Properties in the Environment

Before deploying your eWay, you will need to set the properties of the eWay environment using the following descriptions.

Figure 3 eWay Environment Configuration



Client Listener Port Number

Description

Specify the I/O port number on which the server is listening for connection requests.

Required Values

A valid port number. The default is **3428**.

Password

Description

Specify the password used to access the TigerLogic database.

Required Values

Any valid string. The default is **admin**.

Server IP Address

Description

Specify the IP address where the TigerLogic XML Server resides.

Required Values

A valid port number. The default is **localhost**.

Server PortNumber

Description

Specify the I/O port number on which the TigerLogic XML server is listening for connection requests.

Required Values

A valid port number. The default is **3408**.

UserName

Description

Specify the user name the eWay uses to connect to the TigerLogic XML server.

Required Values

Any valid string. The default is **admin**.

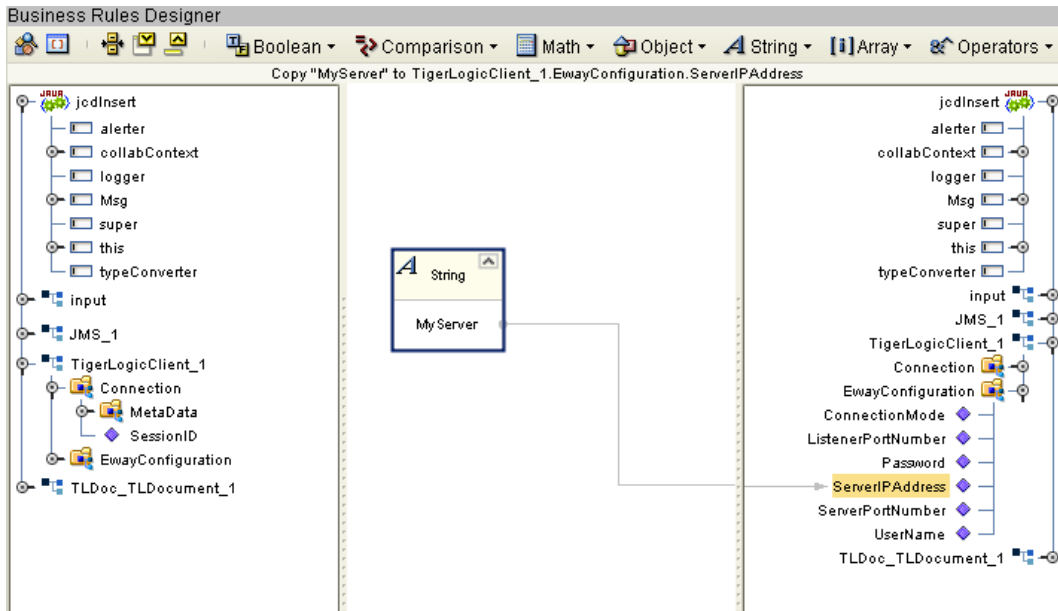
3.3 Manually Configuring the eWay

Manual mode allows you to override eWay configurations in the Java Collaboration Definition. To enable this, set the TigerLogic Connection Mode (in the outbound Connectivity Map properties) to “**Manual**”. The TigerLogic Connection Mode is only available for outbound mode (see “**TigerLogic Connection Mode**” on page 16).

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

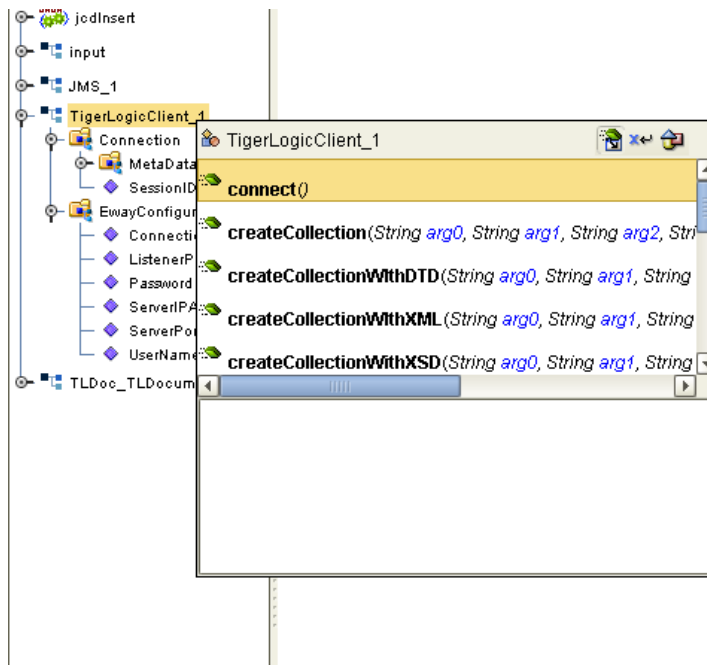
In Manual mode, all of the available properties and methods can be changed through the Collaboration’s OTDs. For example, to set the **ServerIPAddress**, you assign a **literal string** to the **ServerIPAddress** parameter on the right hand side of the Collaboration in the Collaboration Editor (see Figure 4). You can also change the property value using other sources such as the content in an input file.

Figure 4 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 5).

Figure 5 Manually Configuring an eWay



3.4 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*.

Chapter 4

Working with the Sample Project

This chapter describes how to build a TigerLogic eWay Project in a production environment.

What's in This Chapter

- [Sample Project Overview](#) on page 21
- [eInsight Engine and eGate Components](#) on page 21
- [Importing the Sample Project in eGate](#) on page 22
- [Before Running the Sample Project](#) on page 23
- [Working with the Sample Project in eGate](#) on page 25
- [Running the Sample Project](#) on page 41

4.1 Sample Project Overview

The Sample Project is designed to provide an overview of the basic functionality of the TigerLogic eWay by demonstrating how to pass information between eGate and the TigerLogic XML database. The Sample Project includes Insert, Replace, Delete, and Query operations. The data used for the Sample Project is contained within input files in XML format.

4.2 eInsight Engine and eGate Components

You can deploy an eGate component as an Activity in an eInsight Business Process. Once you have associated the desired component with an Activity, the eInsight engine can invoke it using a Web Services interface. Examples of eGate components that can interface with eInsight in this way are:

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

Using the eGate 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. When eInsight runs the Business Process, it automatically invokes that component via its Web Services interface.

4.3 Importing the Sample Project in eGate

This section describes how to use the **TigerLogic_Sample** Project with eGate Integrator. You will need to import the **TigerLogic_eWay_Sample.zip** file contained in the eWay sample folder on the installation CD-ROM.

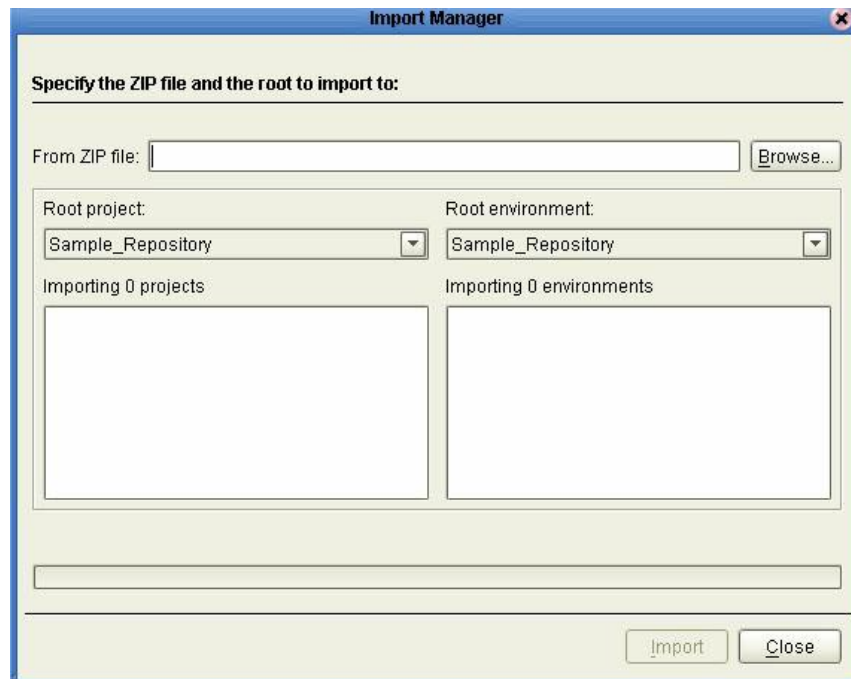
Importing the Sample Project

- 1 On the Enterprise Designer's Project Explorer, right-click the Repository and select **Import**.
- 2 In the **Import Manager** window, browse to the directory that contains the Sample Project zip file.

The Sample Project is contained within the zip file **TigerLogic_eWay_Sample.zip**, which is downloaded from the Repository to a folder of your choosing (see ["Installing the eWay Product Files" on page 11](#)).

- 3 Select the Sample Project zip file and click **Import**.

Figure 6 Select the Project file



- 4 Click the **Refresh All From Repository** icon located on the **Enterprise Explorer** toolbar.

4.4 Before Running the Sample Project

Before running the Sample Project, the following tasks are required:

- Setting the eWay Properties
- Creating the Environment Profile
- Creating and Activating the Deployment Profile

4.4.1 Setting the eWay Properties

Sample projects use both an inbound and an outbound File eWay, as well as an outbound TigerLogic eWay. Use the following information to configure the Sample Project eWays. For additional information on the eWay properties, see [“Setting the TigerLogic eWay Properties” on page 15](#).

To Configure the File eWays:

- 1 On the Connectivity Map, double-click the **Inbound File eWay**.
- 2 The **Properties** window for the Inbound File eWay opens. Modify any parameter settings necessary for your system, then change the Directory and Input file name to match the location and name of the sample data file.
- 3 Click **OK** to close the **Properties** window.
- 4 On the Connectivity Map, double-click the Outbound File eWay. The **Properties** window for the Outbound File eWay opens. Modify any parameter settings necessary for your system, including the target Directory and Output file name.
- 5 Click **OK** to close the Properties window.

To Configure the Outbound TigerLogic eWay:

- 1 On the Connectivity Map, double-click the **TigerLogic eWay**.
- 2 The **Properties** window for the TigerLogic eWay opens. Modify the parameter settings necessary for your system, then click **OK** to close the Properties window.

4.4.2 Creating the Environment Profile

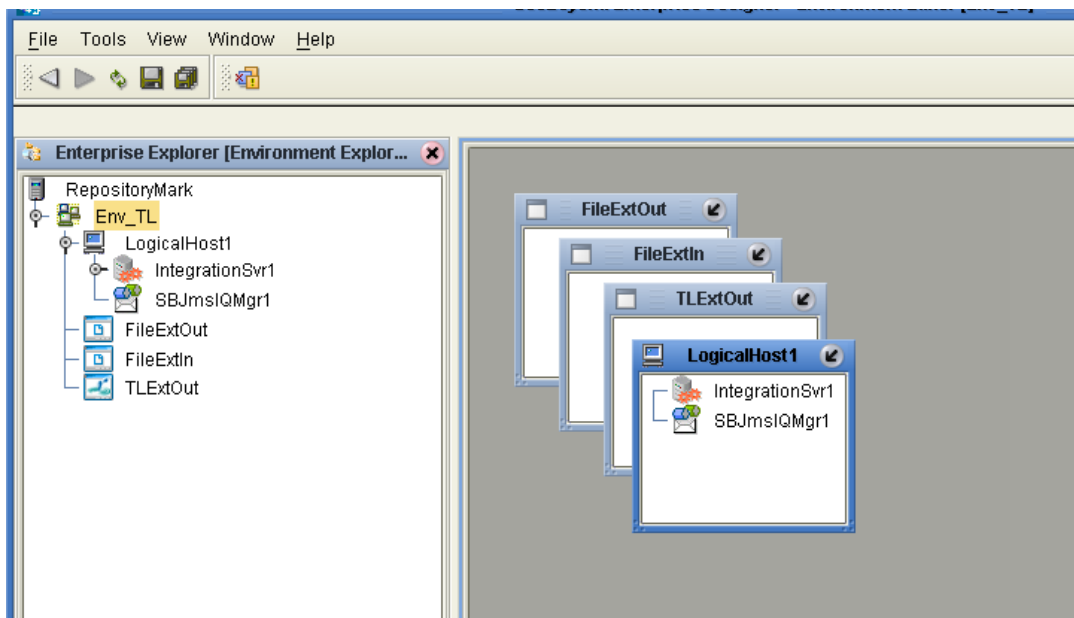
An eGate Environment represents the physical system required to implement a Project. A typical Environment contains several components, including Logical Hosts, Integration Servers, Message Servers, and External Systems. Environments are created using the Enterprise Designer’s Environment Explorer.

To create the TigerLogic 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_TL**.

- 3 Right-click **Env_TL** and select **New TigerLogic External System**. Name the External System **TLExtOut**. Click **OK**.
- 4 Right-click **Env_TL** 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**.
- 5 Right-click **Env_TL** 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**.
- 6 Right-click **Env_TL** and select **New Logical Host**. The **LogicalHost1** box is added to the Environment and **LogicalHost1** is added to the Environment Editor tree.
- 7 From the Environment Explorer tree, right-click the **LogicalHost1**, and select **New SeeBeyond Integration Server**. A new Integration Server, **IntegrationSvr1**, is added to the **LogicalHost1** box.
- 8 From the Environment Explorer tree, right-click the **LogicalHost1**, and select **New SeeBeyond JMS IQ Manager**. A new JMS IQ Manager, **SBJmsIQMgr1**, is added to the **LogicalHost1** box (see Figure 7).

Figure 7 Environment Editor - Env_TL Environment



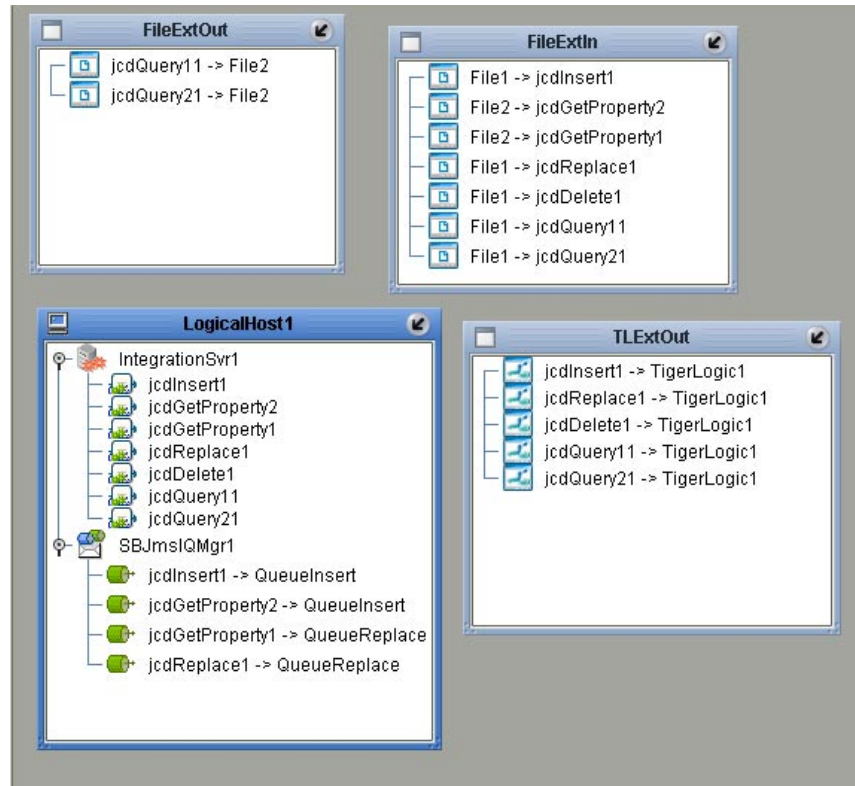
4.4.3 Creating and Activating the Deployment Profile

A Deployment Profile is used to assign Collaborations and Message Destinations to the Integration Server and Message Server. Deployment Profiles are created using the Deployment Editor. To create the sample Deployment Profile, do the following.

- 1 From the Enterprise Explorer's Project Explorer, right-click the Project (**TigerLogic_Sample**) and select **New > Deployment Profile**.
- 2 Enter a name for the Deployment Profile (for this sample **DP_TL**). Make sure that the selected Environment is **Env_TL**. Click **OK**.

- 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 component can be mapped manually. Refer to the figure below.

Figure 8 Deployment Profile - Auto Map



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

4.5 Working with the Sample Project in eGate

The eGate Sample Project **TigerLogic_Sample** includes four operations (Insert, Replace, Delete, and Query). The sections below describe each of these operations and their implementation in the Sample Project. Connectivity Maps, Collaboration Definitions, and Business Rules are detailed for each operation.

4.5.1 Sample Project Inputs

The eGate sample project includes a file and JMS input. The file input contains the actual data and is stored in XML format. The JMS input contains information pertaining to the database, such as the values for Database, Collection, and Document. It is not necessary to send this information via JMS. The information may be hard coded in the Java collaboration or another eWay (such as a Batch eWay) may be used to accomplish

the task. It is also possible to retrieve the same information directly from the XML file where the actual data is stored, provided that the database information is located there. In the following Sample Project where JMS is used, the collaboration to insert or replace data in the database is triggered by two files. The first file, whose filename begins with **TLDoc**, triggers data to be stored in JMS. The second file triggers data to be stored in the database. The two files must be triggered in that order.

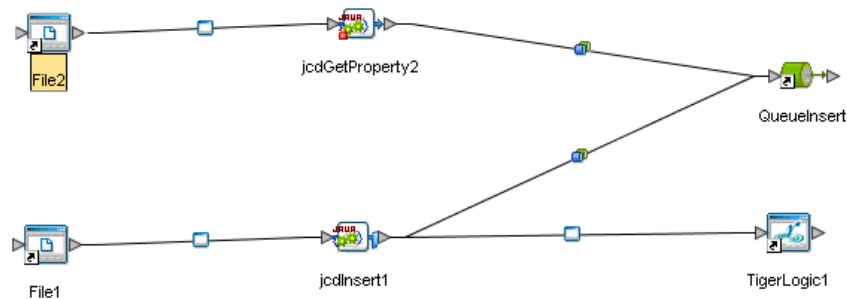
4.5.2 Sample Project Insert Operation

In the Sample Project Insert operation, the eWay subscribes to an XML input file. Using a File eWay as a trigger, XML data is picked up and published to a Java Collaboration. The Collaboration unmarshals the data, converts it to a series of arguments, and then marshals the data. The data is published to the outbound TigerLogic eWay. The TigerLogic eWay inserts the data in an existing XML data file.

Insert Operation Connectivity Map

The figure below shows a sample Connectivity Map illustrating the Insert operation.

Figure 9 Insert Operation Connectivity Map



jcdInsert Collaboration Definition (Java)

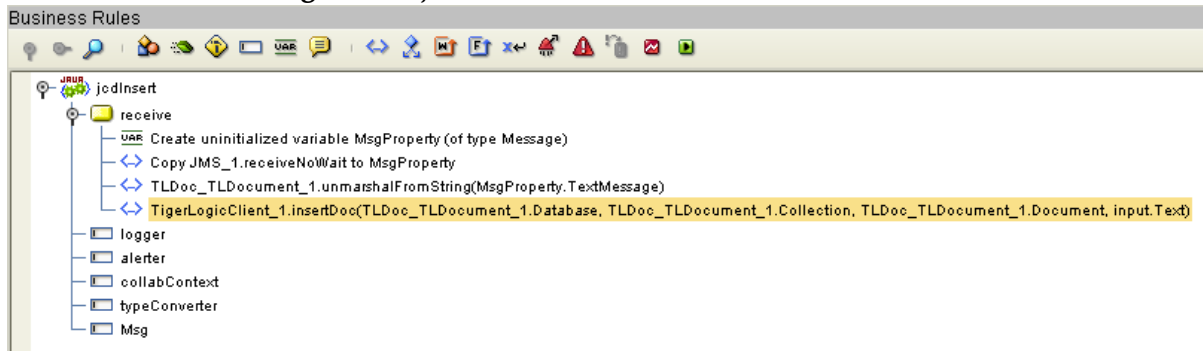
The jcdInsert Collaboration Definition uses the Insert operation to add a new record to the sample TigerLogic XML database. The text input is added as a string literal to confirm the completion before writing to an output file.

jcdInsert Collaboration Business Rules

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

The **jcdInsert** Collaboration contains the Business Rules displayed in Figure 10.

Figure 10 jcdInsert Collaboration Business Rules

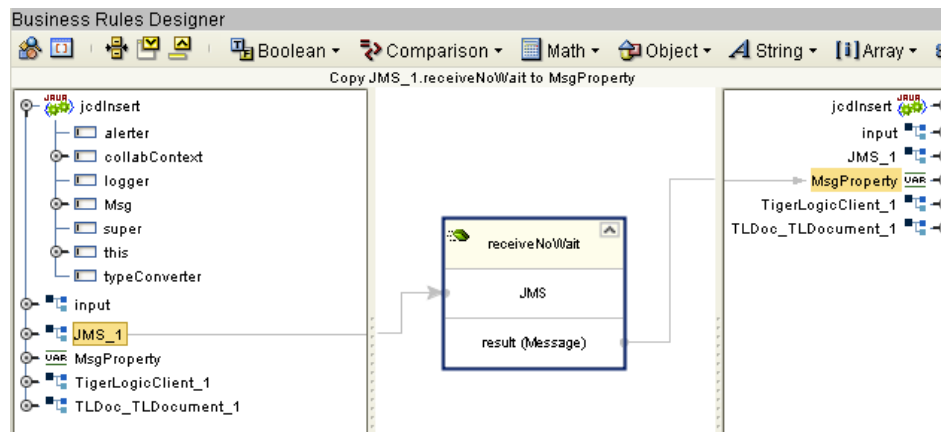


To create the jcdInsert Collaboration Business Rules, do the following:

- 1 Create the **Create uninitialized variable MsgProperty (of type Message)** Business Rule:
 - A From the Business Rules toolbar, click **Local Variable**. The **Create Variable** dialog box appears.
 - B Enter **MsgProperty** as the variable name, select **Class**, type **com.stc.connectors.jms.Message** in the **Class** field, and click **OK**.
- 2 Create the **CopyJMS_1.receiveNoWait to MsgProperty** Business Rule:
 - A Right-click the **JMS_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
 - B Select **receiveNoWait()** from the method selection menu. The **receiveNoWait()** method box appears with an arrow connecting **JMS_1**, in the left pane of the Business Rule Designer, to the **MsgProperty** input node (see Figure 11).
 - C This creates the following Java code shown in the Java Source Editor:


```
MsgProperty=JMS_1.receiveNoWait();
```

Figure 11 Insert Operation Business Rules Designer

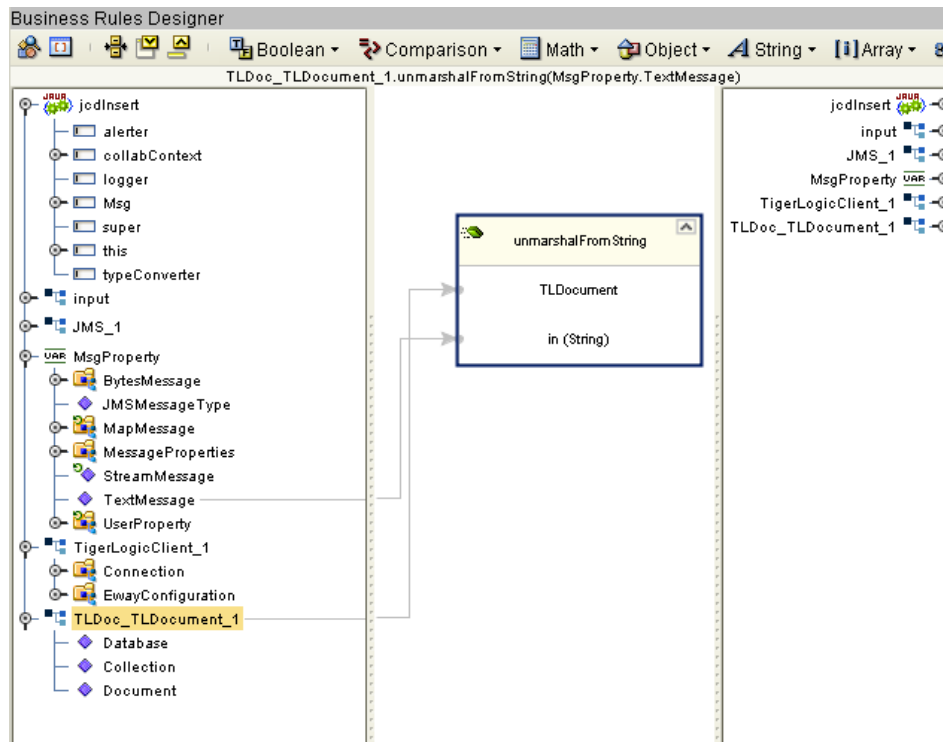


- 3 Create the **TLDoc_TLDocument_1.unmarshalFromString(MsgProperty.TextMessage)** Business Rule:

- A From the Business Rules toolbar click the **rule** button to add a new rule.
- B Right-click the **TLDoc_TLDocument_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
- C Select **unmarshalFromString(String[] in)** from the method selection menu. The **unmarshalFromString** method box appears with an arrow connecting **TLDoc_TLDocument_1**, in the left pane of the Business Rule Designer, to the **TLDocument** input node.
- D From the left pane of the Business Rules Designer, map **TextMessage**, under **MsgProperty**, to the **in(String)** input node of the **unmarshalFromString** method box. To do this, click on **TextMessage** in the left pane of the Business Rules Designer and drag the cursor to the **in (String)** input node of the **unmarshalFromString** method box (see Figure 12).
- E This creates the following Java code shown in the Java Source Editor:


```
TLDoc_TLDocument_1.unmarshalFromString (
MsgProperty.getTextMessage() );
```

Figure 12 Insert Operation Business Rules Designer

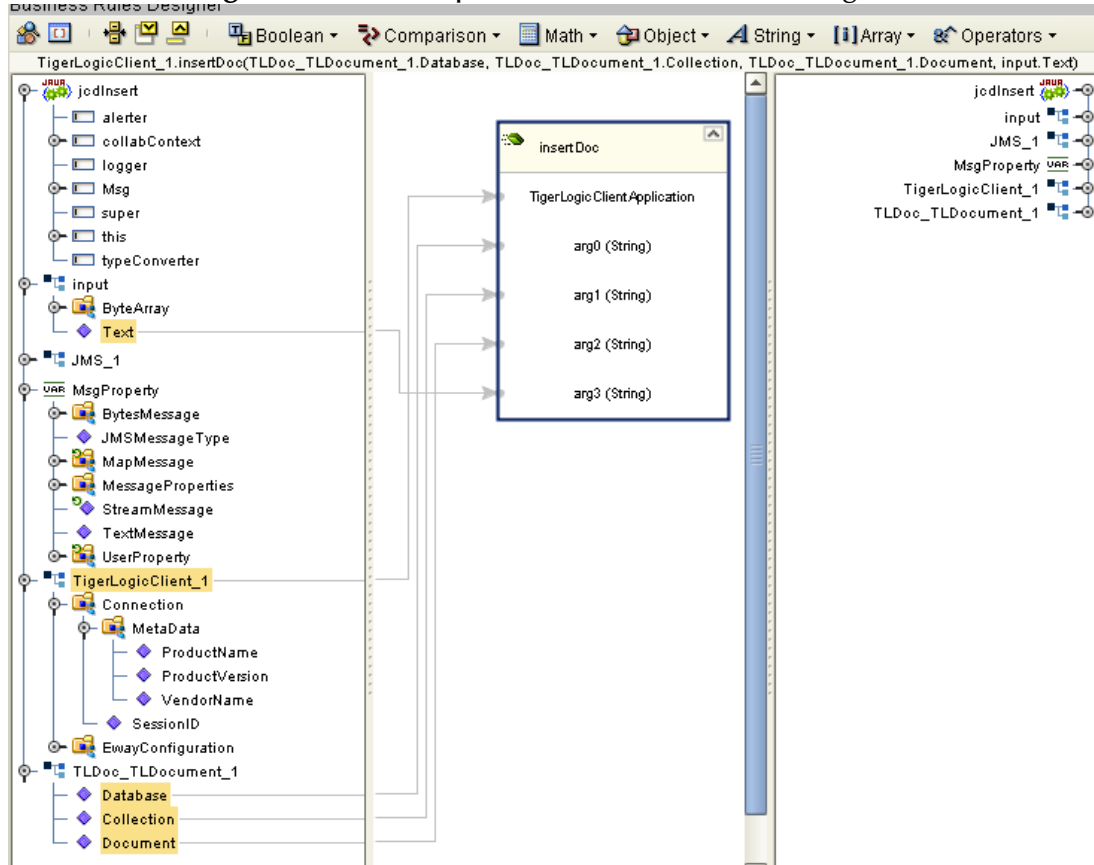


- 4 Create the **TigerLogicClient_1.insertDoc(TLDoc_TLDocument_1.Database, TLDoc_TLDocument_1.Collection, TLDoc_TLDocument_1.Document, input.Text)** Business Rule:
 - A From the Business Rules toolbar click the **rule** button to add a new rule.
 - B Right-click the **TigerLogicClient_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.

- C Select `insertDoc(String arg0, String arg1, String arg2, String arg3)` from the method selection menu. The `insertDoc` method box appears with an arrow connecting `TigerLogicClient_1`, in the left pane of the Business Rule Designer, to the `TigerLogicClient Application` input node.
- D From the left pane of the Business Rules Designer, map `Database`, under `TLDoc_TLDocument_1`, to the `arg0 (String)` input node of the `insertDoc` method box. To do this, click on `Database` in the left pane of the Business Rules Designer and drag the cursor to the `arg0 (String)` input node of the `unmarshalFromString` method box.
- E Repeat step D for `Collection` and `Document` to the `arg1 (String)` and `arg2 (String)` input nodes, respectively.
- F From the left pane of the Business Rules Designer, map `Text`, under `input`, to the `arg3 (String)` input node of the `insertDoc` method box (see Figure 13).
- G This creates the following Java code shown in the Java Source Editor:

```
TigerLogicClient_1.insertDoc(  
TLDoc_TLDocument_1.getDatabase(),  
TLDoc_TLDocument_1.getCollection(),  
TLDoc_TLDocument_1.getDocument(), input.getText());
```

Figure 13 Insert Operation Business Rules Designer



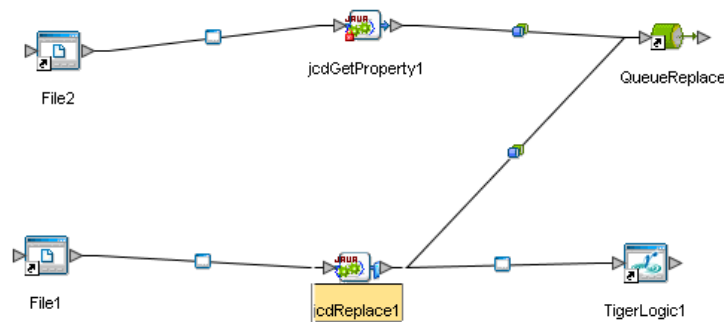
4.5.3 Sample Project Replace Operation

In the Replace operation, the eWay subscribes to an XML input file. Using a File eWay as a trigger, XML data is picked up and published to a Java Collaboration. The Collaboration unmarshals the data, converts it to a series of arguments, and then marshals the data. The data is published to the outbound TigerLogic eWay. The TigerLogic eWay replaces the specified data record in an existing XML data file.

Replace Operation Connectivity Map

The figure below shows a sample Connectivity Map illustrating the Replace operation.

Figure 14 Replace Operation Connectivity Map



jcdReplace Collaboration Definition (Java)

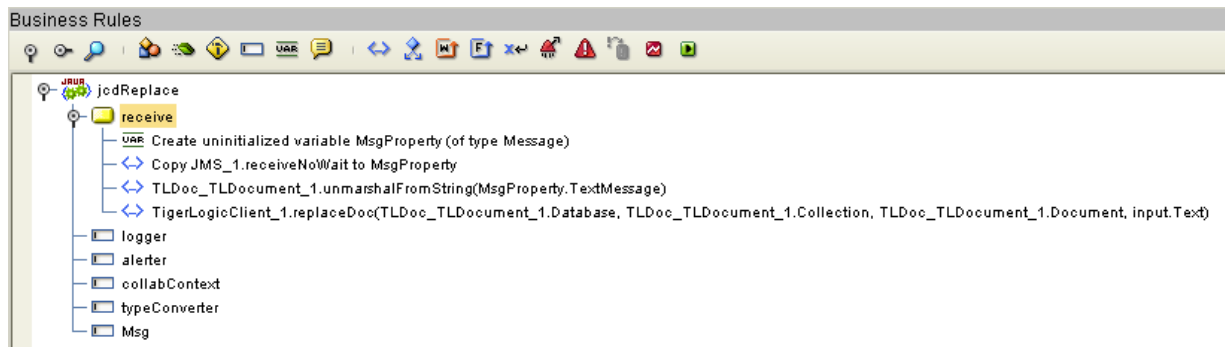
The jcdReplace Collaboration Definition uses the Replace operation to replace or update a data set in the sample TigerLogic XML database. The text input is added as a string literal to confirm the completion before writing to an output file.

jcdReplace Collaboration Business Rules

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

The **jcdReplace** Collaboration contains the Business Rules displayed in Figure 15.

Figure 15 jcdReplace Collaboration Business Rules



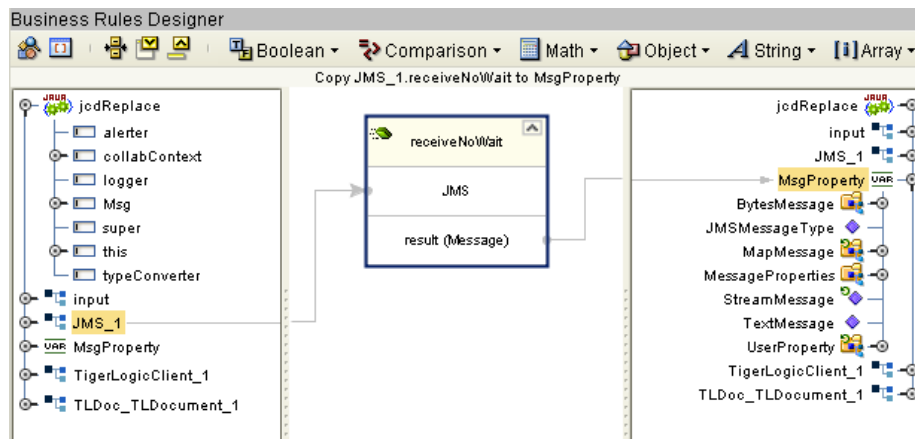
To create the jcdReplace Collaboration Business Rules, do the following:

- 1 Create the **Create uninitialized variable MsgProperty (of type Message)** Business Rule:
 - A From the Business Rules toolbar, click **Local Variable**. The **Create Variable** dialog box appears.
 - B Enter **MsgProperty** as the variable name, select **Class**, type **com.stc.connectors.jms.Message** in the **Class** field, and click **OK**.

- 2 Create the **Copy JMS_1.receiveNoWait to MsgProperty** Business Rule:
 - A From the Business Rules toolbar click the **rule** button to add a new rule.
 - B Right-click the **JMS_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
 - C Select **receiveNoWait()** from the method selection menu. The **receiveNoWait()** method box appears with an arrow connecting **JMS_1**, in the left pane of the Business Rule Designer, to the **MsgProperty** input node (see Figure 16).
 - D This creates the following Java code shown in the Java Source Editor:


```
MsgProperty = JMS_1.receiveNoWait();
```

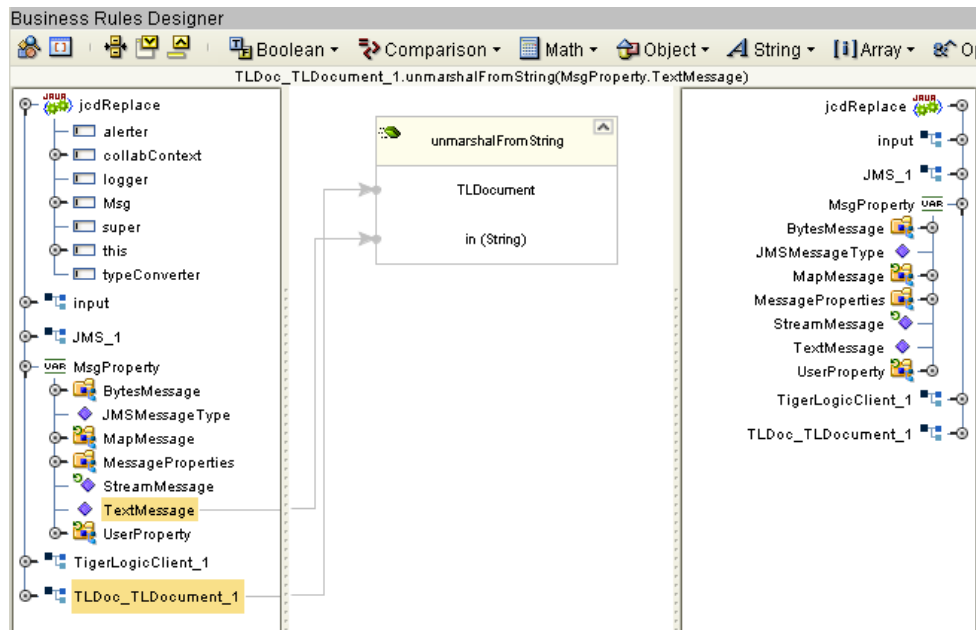
Figure 16 Replace Operation Business Rules Designer



- 3 Create the **TLDoc_TLDocument_1.unmarshalFromString(MsgProperty.TextMessage)** Business Rule:
 - A From the Business Rules toolbar click the **rule** button to add a new rule.
 - B Right-click the **TLDoc_TLDocument_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
 - C Select **unmarshalFromString(String[] in)** from the method selection menu. The **unmarshalFromString** method box appears with an arrow connecting **TLDoc_TLDocument_1**, in the left pane of the Business Rule Designer, to the **TLDocument** input node.
 - D From the left pane of the Business Rules Designer, map **TextMessage**, under **MsgProperty**, to the **in(String)** input node of the **unmarshalFromString** method box. To do this, click on **TextMessage** in the left pane of the Business Rules Designer and drag the cursor to the **in (String)** input node of the **unmarshalFromString** method box (see Figure 17).
 - E This creates the following Java code shown in the Java Source Editor:


```
TLDoc_TLDocument_1.unmarshalFromString (
MsgProperty.getTextMessage() );
```

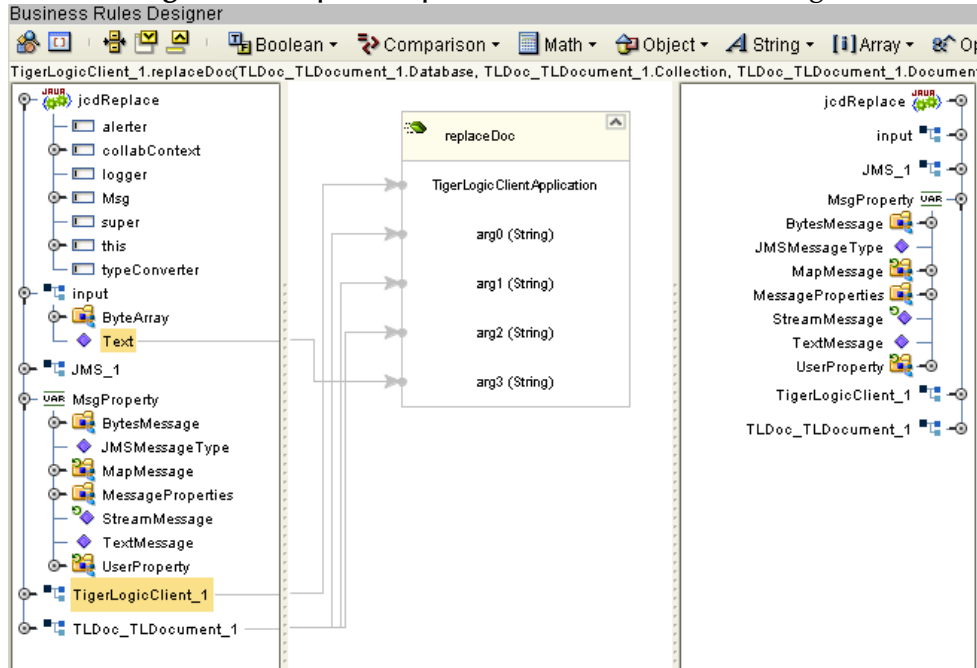

Figure 17 Replace Operation Business Rules Designer



- 4 Create the **TigerLogicClient_1.replaceDoc(TLDoc_TLDocument_1.Database, TLDoc_TLDocument_1.Collection, TLDoc_TLDocument_1.Document, input.Text)** Business Rule:
 - A From the Business Rules toolbar click the **rule** button to add a new rule.
 - B Right-click the **TigerLogicClient_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
 - C Select **replaceDoc(String arg0, String arg1, String arg2, String arg3)** from the method selection menu. The **replaceDoc** method box appears with an arrow connecting **TigerLogicClient_1**, in the left pane of the Business Rule Designer, to the **TigerLogicClient Application** input node.
 - D From the left pane of the Business Rules Designer, map **Database**, under **TLDoc_TLDocument_1**, to the **arg0 (String)** input node of the **insertDoc** method box. To do this, click on **Database** in the left pane of the Business Rules Designer and drag the cursor to the **arg0 (String)** input node of the **unmarshalFromString** method box.
 - E Repeat step D for **Collection** and **Document** to the **arg1 (String)** and **arg2 (String)** input nodes, respectively.
 - F From the left pane of the Business Rules Designer, map **Text**, under **input**, to the **arg3 (String)** input node of the **replaceDoc** method box (see Figure 18).
 - G This creates the following Java code shown in the Java Source Editor:


```
TigerLogicClient_1.replaceDoc(
TLDoc_TLDocument_1.getDatabase(),
TLDoc_TLDocument_1.getCollection(),
TLDoc_TLDocument_1.getDocument(), input.getText());
```

Figure 18 Replace Operation Business Rules Designer



4.5.4 Sample Project Delete Operation

In the Delete operation, the eWay subscribes to an XML input file. Using a File eWay as a trigger, XML data is picked up and published to a Java Collaboration. The Collaboration unmarshals the data, converts it to a series of arguments, and then marshals the data. The new data is published to the outbound TigerLogic eWay. The TigerLogic eWay supplies the identification of the document to be deleted in an existing XML data file, effectively deleting the requested data set.

Delete Operation Connectivity Map

The figure below shows a sample Connectivity Map illustrating the Delete operation.

Figure 19 Delete Operation Connectivity Map



jcdDelete Collaboration Definition (Java)

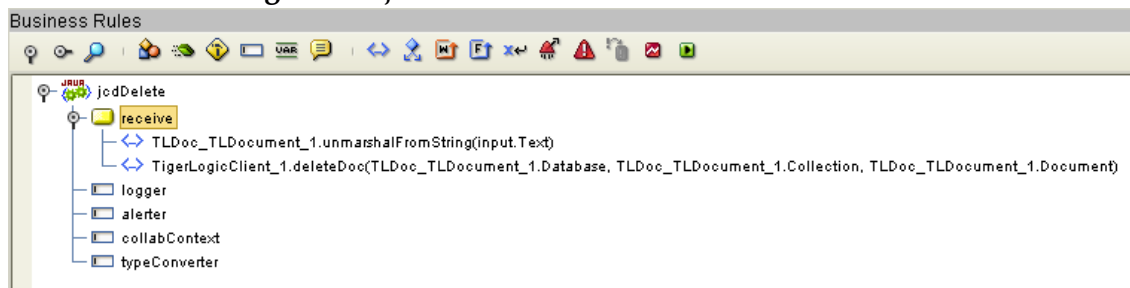
The jcdDelete Collaboration Definition uses the Delete operation to delete a data set from the sample TigerLogic XML database. The text input is added as a string literal to confirm the completion of the operation.

jcdDelete Collaboration Business Rules

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

The jcdDelete Collaboration contains the Business Rules displayed in Figure 20.

Figure 20 jcdDelete Collaboration Business Rules

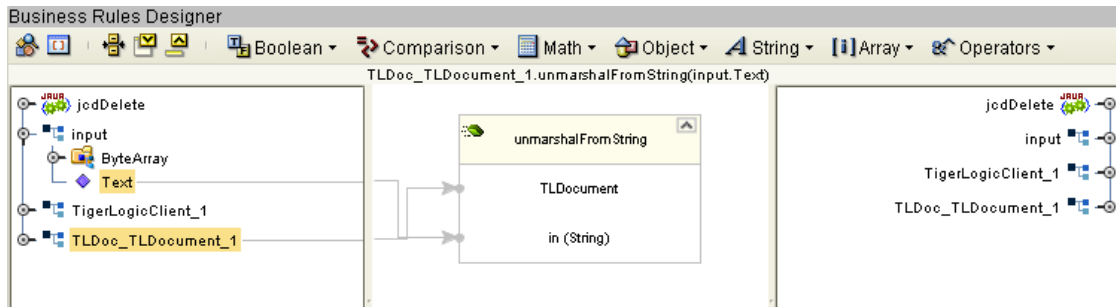


To create the jcdDelete Collaboration Business Rules, do the following:

- 1 Create the **TLDoc_TLDocument_1.unmarshalFromString(input.Text)** Business Rule:
 - A From the Business Rules toolbar click the **rule** button to add a new rule.
 - B Right-click the **TLDoc_TLDocument_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
 - C Select **unmarshalFromString(String[] in)** from the method selection menu. The **unmarshalFromString** method box appears with an arrow connecting **TLDoc_TLDocument_1**, in the left pane of the Business Rule Designer, to the **TLDocument** input node.
 - D From the left pane of the Business Rules Designer, map **Text**, under input, to the **in(String)** input node of the **unmarshalFromString** method box. To do this, click on **Text** in the left pane of the Business Rules Designer and drag the cursor to the **in (String)** input node of the **unmarshalFromString** method box (see Figure 21).
 - E This creates the following Java code shown in the Java Source Editor:

```
TLDoc_TLDocument_1.unmarshalFromString( input.getText() );
```

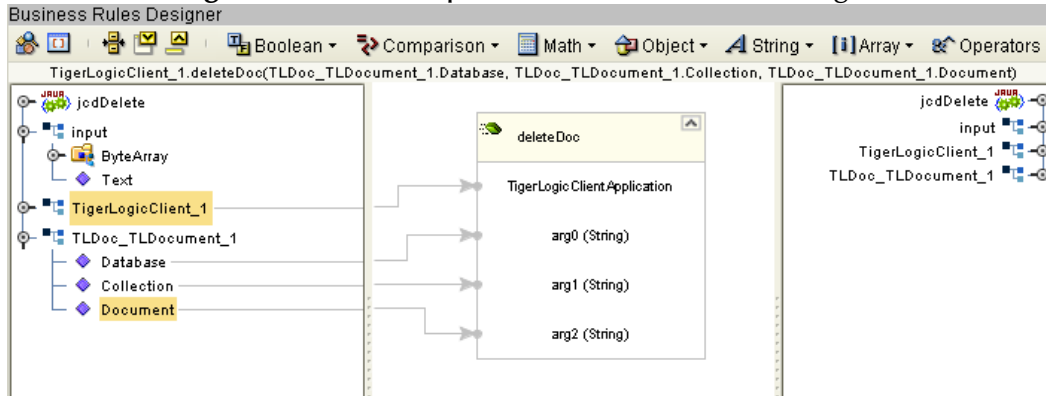
Figure 21 Delete Operation Business Rules Designer



- 2 Create the **TigerLogicClient_1.deleteDoc(TLDoc_TLDocument_1.Database, TLDoc_TLDocument_1.Collection, TLDoc_TLDocument_1.Document)** Business Rule:
 - A From the Business Rules toolbar click the **rule** button to add a new rule.
 - B Right-click the **TigerLogicClient_1** node in the left pane of the Business Rule Designer, and click **Select method to call**. The method selection menu appears.
 - C Select **deleteDoc(String arg0, String arg1, String arg2)** from the method selection menu. The **deleteDoc** method box appears with an arrow connecting **TigerLogicClient_1**, in the left pane of the Business Rule Designer, to the **TigerLogicClient Application** input node.
 - D From the left pane of the Business Rules Designer, map **Database**, under **TLDoc_TLDocument_1**, to the **arg0 (String)** input node of the **insertDoc** method box. To do this, click on **Database** in the left pane of the Business Rules Designer and drag the cursor to the **arg0 (String)** input node of the **unmarshalFromString** method box.
 - E Repeat step D for **Collection** and **Document** to the **arg1 (String)** and **arg2 (String)** input nodes, respectively (see Figure 22).
 - F This creates the following Java code shown in the Java Source Editor:


```
TigerLogicClient_1.deleteDoc (
TLDoc_TLDocument_1.getDatabase () ,
TLDoc_TLDocument_1.getCollection () ,
TLDoc_TLDocument_1.getDocument () ) ;
```

Figure 22 Delete Operation Business Rules Designer



4.5.5 Sample Project Query Operations

There are two Query operations in the TigerLogic eWay Sample Project. Query1 searches for the available data within the Sample Collection. The XQuery for this Query operation is stored in an input XML file. Query2 searches for a data set using a specific parameter. This specific parameter, such as a personal name “John Smith”, is stored in an input XML file that serves as the value for the **where** clause of the Query operation. The XML XQuery files can be manipulated to fit the needs of the user.

Query Operations Connectivity Maps

The figures below show sample Connectivity Maps illustrating the Query1 and Query 2 operations.

Figure 23 Query1 Operation Connectivity Map

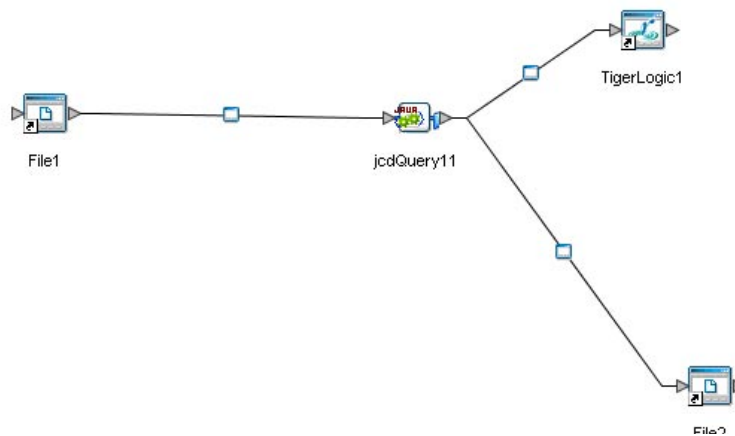
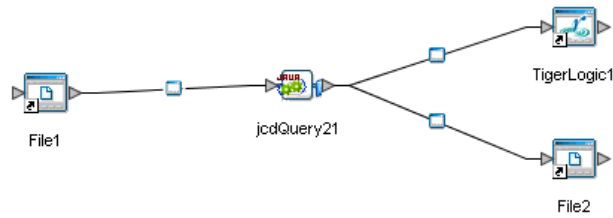


Figure 24 Query2 Operation Connectivity Map



jcdQuery Collaboration Definitions (Java)

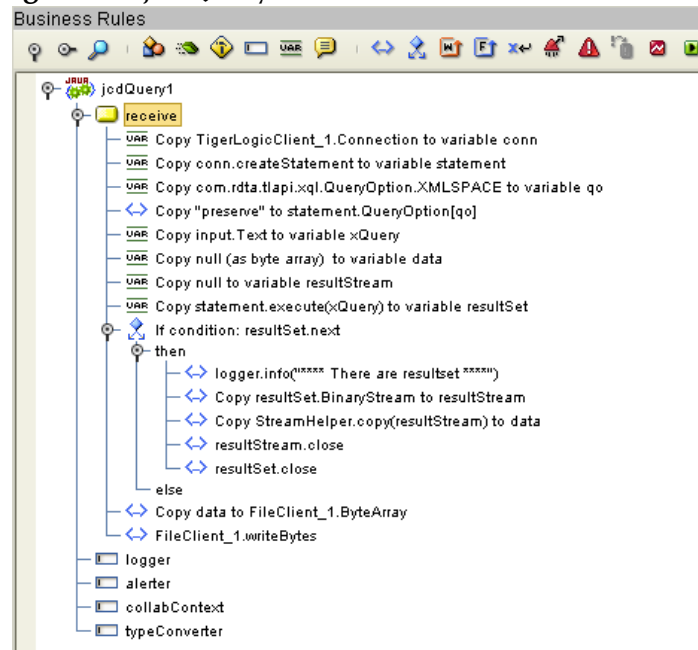
The jcdQuery1 and jcdQuery 2 Collaboration Definitions uses the Query operation to retrieve a particular data set from the sample TigerLogic XML database. The matching data set is concatenated with the corresponding query definition and written to an output file.

jcdQuery1 Collaboration Business Rules

The Business Rules for the **jcdQuery1** Collaboration are created using the Collaboration Editor (Java). To open the Collaboration Editor to the **jcdQuery1** Collaboration, double-click the **jcdQuery1** Collaboration from the Project Explorer tree. The Collaboration Editor opens to the selected Collaboration.

The **jcdQuery1** Collaboration contains the Business Rules displayed in Figure 25.

Figure 25 jcdQuery1 Collaboration Business Rules



The corresponding XQuery for the Query1 operation of the Sample Project is as follows:

```

for $q in collection('tig:///ICANSample/SampleCollection')
return (
  <Address>
  <Name data="{ $q//Address/Name }"></Name>
  <Street data="{ $q//Address/Street }"></Street>
  <City data="{ $q//Address/City }"></City>
  <State data="{ $q//Address/State }"></State>
  <Zip data="{ $q//Address/Zip }"></Zip>
</Address>
)

```

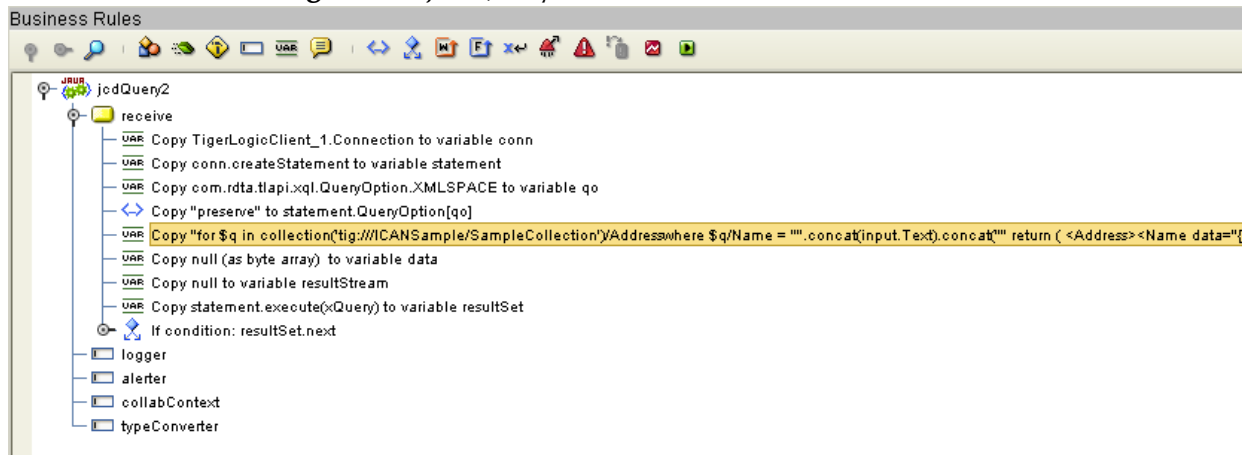
The xQuery may be edited in TigerLogic’s XQuery Editor or any XML file editing tool. For further information, see the *TigerLogic XQuery Editor Guide*.

jcdQuery2 Collaboration Business Rules

The Business Rules for the **jcdQuery2** Collaboration are created using the Collaboration Editor (Java). To open the Collaboration Editor to the **jcdQuery2** Collaboration, double-click the **jcdQuery2** Collaboration from the Project Explorer tree. The Collaboration Editor opens to the selected Collaboration.

The **jcdQuery2** Collaboration contains the Business Rules displayed in Figure 26.

Figure 26 jcdQuery2 Collaboration Business Rules

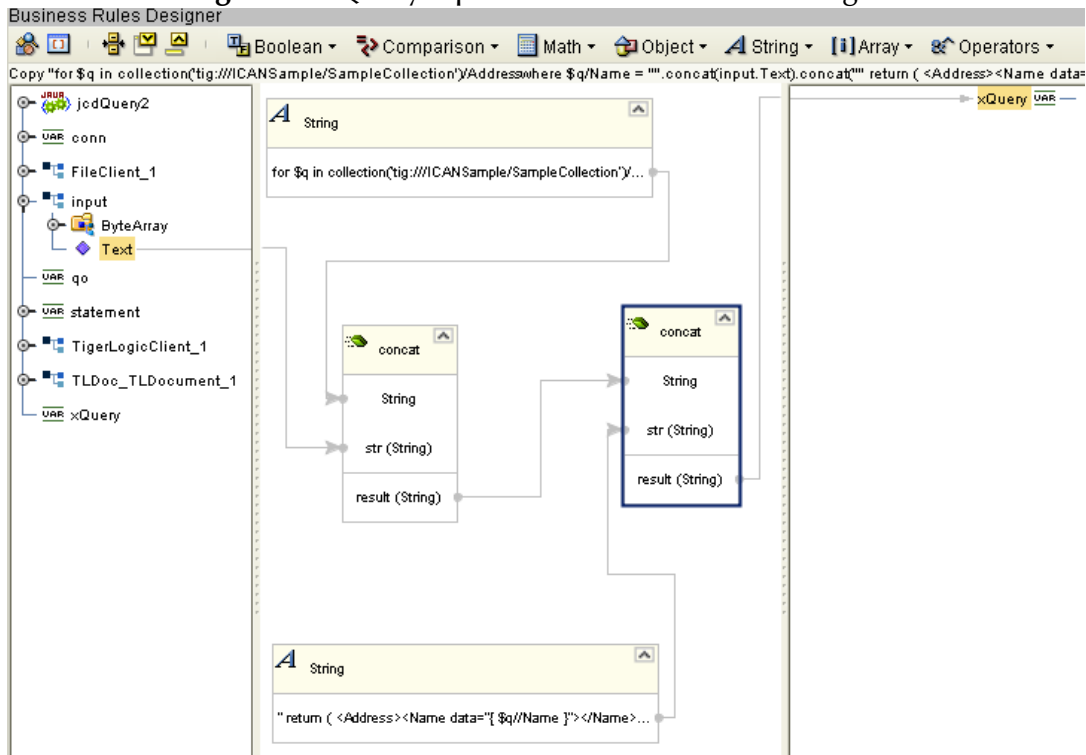


- 1 Create the XQuery variable:
 - A From the Business Rules toolbar click the **Local Variable** button to add a new variable.
 - B From the **String** pull-down menu on the Business Rules Designer toolbar, click **Literal String**. Add the following value to the Literal String: for \$q in collection('tig:///ICANSample/SampleCollection')/Address where \$q/Name = "
 - C From the **String** pull-down menu on the Business Rules Designer toolbar, click the **Concat** operation.
 - D Map the Literal String of Step B to the **String** input of the **Concat** operation.

- E From the left pane of the Business Rules Designer, map **Text**, under **input**, to the **str (String)** input of the **Concat** operation.
- F Create a second **Concat** operation from the **String** pull-down menu on the Business Rules Designer toolbar.
- G Map the result value of the **Concat** operation created in Step C to the new **Concat** operation.
- H From the **String** pull-down menu on the Business Rules Designer toolbar, click **Literal String**. Add the following value to the Literal String:


```
"return (
<Address>
<Name data="{ $q//Name }"></Name>
<Street data="{ $q//Street }"></Street>
<City data="{ $q//City }"></City>
<State data="{ $q//State }"></State>
<Zip data="{ $q//Zip }"></Zip>
</Address>
)
```
- I Map the Literal String of Step H to the **str (String)** input of the second **Concat** operation.
- J Map the result value of the second **Concat** operation created in Step F to the **xQuery** variable in the right pane of the Business Rules Designer (see Figure 27).

Figure 27 Query Operation Business Rules Designer



The corresponding XQuery for the Query2 operation of the Sample Project is as follows:

```
for $q in collection('tig:///ICANSample/SampleCollection')/
Address
where $q/Name = "John Smith"
  return (
<Address>
<Name data="{ $q//Name }"></Name>
<Street data="{ $q//Street }"></Street>
<City data="{ $q//City }"></City>
<State data="{ $q//State }"></State>
<Zip data="{ $q//Zip }"></Zip>
</Address>
)
```

The xQuery may be edited in TigerLogic's XQuery Editor or any XML file editing tool. For further information, see the *TigerLogic XQuery Editor Guide*.

4.6 Running the Sample Project

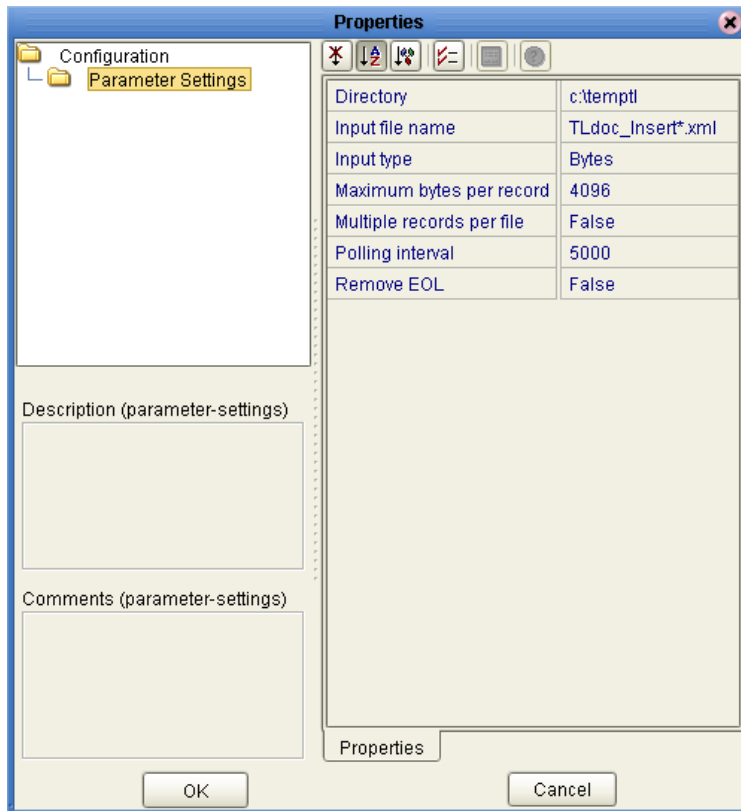
The following directions assume that the Enterprise Designer was downloaded to **C:\ican50**. If this is not the case, replace that location in the following directions with the appropriate location.

- 1 From the Enterprise Manager **Downloads** tab, download **Logical Host - for win32**.
- 2 Extract the file to the **ican50** 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: **ican50\LogicalHost1\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: **logicalhost**

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 **Insert** operation, first copy the sample **TLdoc_Insert** file to its designated input folder. You will need to rename the extension of the file to **.xml**. See Figure 28 below.

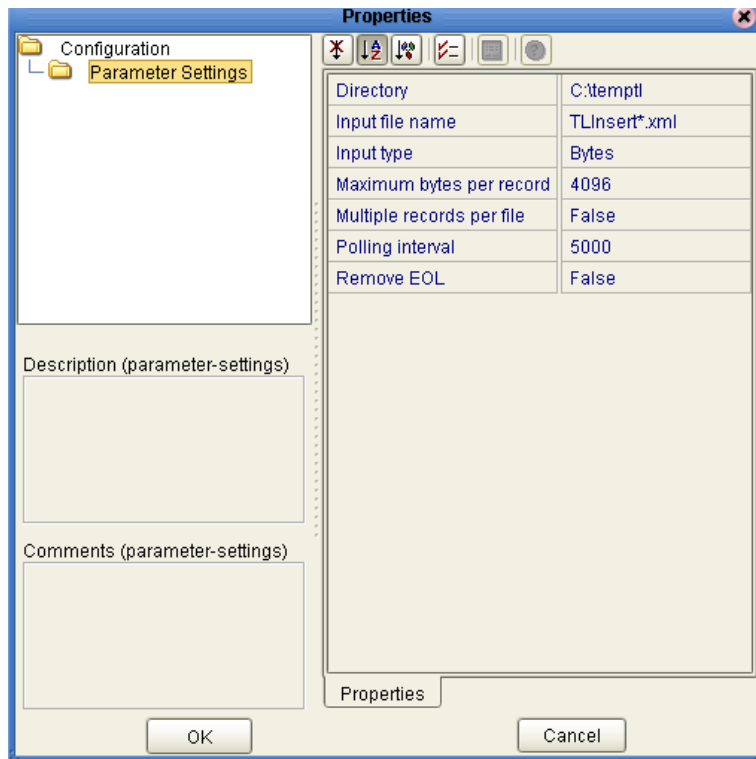
Figure 28 Insert Operation Input File Settings



Then copy the sample **TLInsert** file to the input folder. You will need to rename the extension of the file to **.xml**.

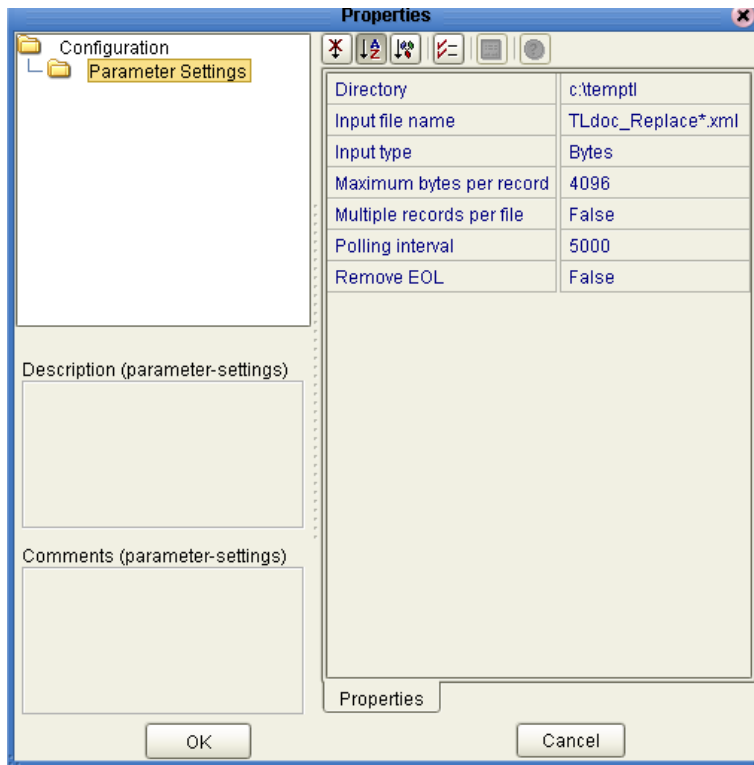
The file name and input folder location are specified in the inbound File eWay properties window of the **Insert** operation Connectivity Map. See Figure 29 below.

Figure 29 Insert Operation Input File Settings



- 7 To run the **Replace** operation, first copy the sample **TLdoc_Replace** file to its designated input folder. You will need to rename the extension of the file to **.xml**. See Figure 30 below.

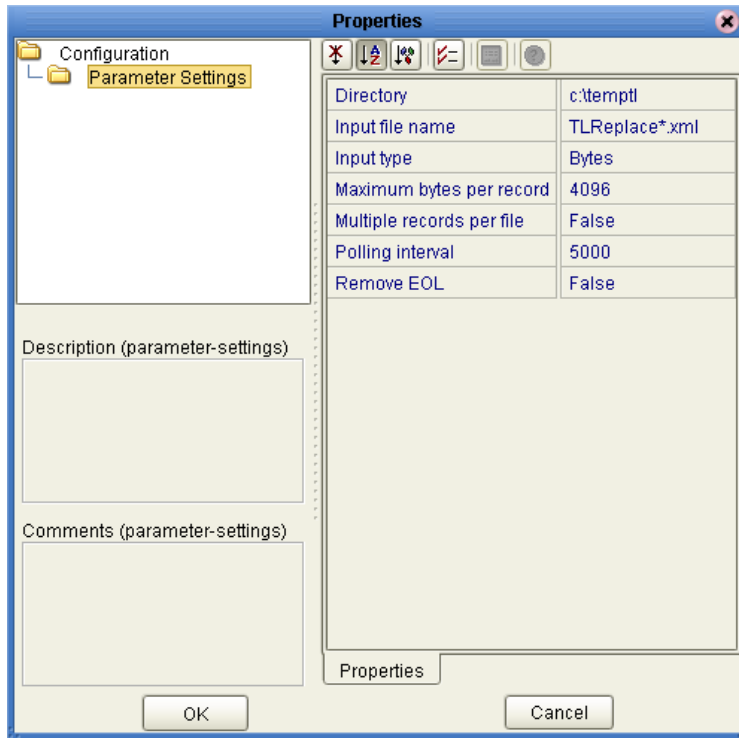
Figure 30 Replace Operation Input File Settings



Then copy the sample **TLReplace** file to the input folder. You will need to rename the extension of the file to **.xml**.

The file name and input folder location are specified in the inbound File eWay properties window of the **Insert** operation Connectivity Map. See Figure 31 below.

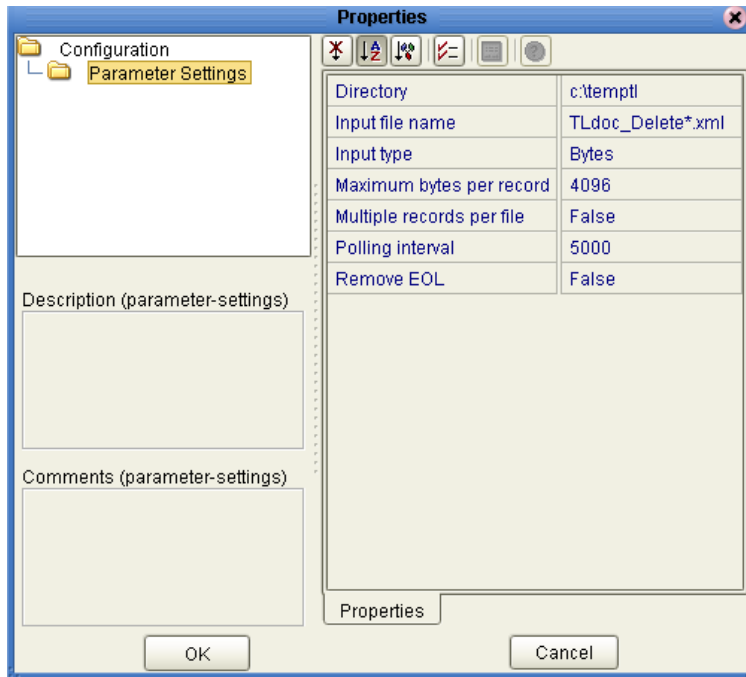
Figure 31 Replace Operation Input File Settings



- 8 To run the **Delete** operation, copy the sample **TLdoc_Delete** file to the input folder. You will need to rename the extension of the file to **.xml**.

The file name and input folder location are specified in the inbound File eWay properties window of the **Delete** operation Connectivity Map. See Figure 32 below.

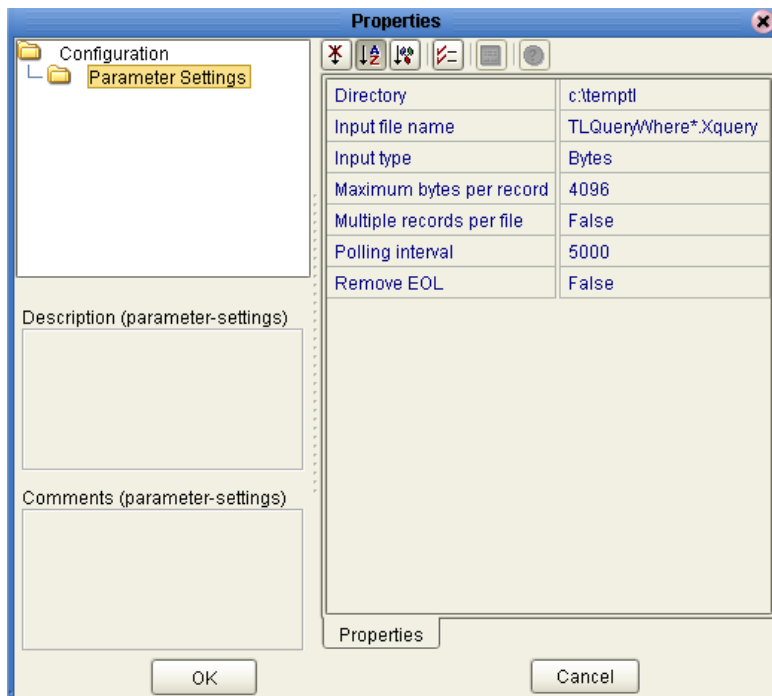
Figure 32 Delete Operation Input File Settings



- 9 To run the **Query** operation, copy the sample **TLQueryWhere** file to the input folder. You will need to rename the extension of the query file to **.Xquery**.

The query file name and input folder location are specified in the inbound File eWay properties window of the **Query** operation Connectivity Map. See Figure 33 below.

Figure 33 Query Operation Input File Settings



Index

A

Alerting 20
auto map 25

C

Client Listener Port Number 17
Configuring eWay Connections 16, 20
connectivity map, outbound properties 15
conventions, document 8

D

Delete Operation 34, 35
 Connectivity Map 34
Deployment Profile
 Auto Map 25
deployment profile 24
document conventions 8
document, scope 7
documents, related 8

E

eDesigner Library
 JAR files 12
environment, properties 17
eWay properties 15
eWay properties, Environment 16
external system requirements 11

H

HP-UX 10

I

IBM AIX 10
Inputs
 Sample Project 25
Insert Operation 26
 Connectivity Map 26
installation
 eWay product files 11

J

JAR files 12
 eDesigner Library 12
 Logical Host 12
 uploading 12

L

Linux
 Red Hat 10
Logical Host
 JAR files 12

M

manual mode 18

O

operating systems 10
 HP-UX 10
 IBM AIX 10
 Red Hat Linux 10
 Sun Solaris 10
 Windows 10
Outbound Environment Properties
 Password 17

P

Password 17
properties 17
Property settings, Outbound Environment
 Password 17

Q

Query Operations 37, 38, 39
 Connectivity Maps 37

R

Red Hat Linux 10
Replace Operation 30, 31
 Connectivity Map 30
requirements
 system 10
 third-party 11

S

Sample Project

Index

- Delete Operation 34
- Inputs 25
- Insert Operation 26
- Query Operations 37
- Replace Operation 30
- sample project
 - creating environment profile 23
 - deployment profile 24
 - importing 22
 - jcdDelete business rules 35
 - jcdDelete collaboration 35
 - jcdInsert business rules 26
 - jcdInsert collaboration 26
 - jcdQuery1 business rules 38
 - jcdQuery2 business rules 39
 - jcdReplace business rules 31
 - jcdReplace collaboration 31
 - overview 21
 - Query collaboration 38
 - running 41
 - setting eWay properties 23
- screenshots 8
- Server IP Address 17
- Server PortNumber 18
- Setting 15, 16
- Setting eWay Connections 15
- Sun Solaris 10
- system requirements
 - ICAN 10

T

- third-party
 - requirements 11
- TigerLogic
 - connection mode 16
 - installing 11
 - manual configuration 18
 - properties 17
 - TLJDK library 12
- TLJDK library 12

U

- UserName 18

W

- Windows 10