SeeBeyond ICAN Suite

IMS eWay Intelligent Adapter User's Guide

Release 5.0.5

SEEBEYOND®

Version 20050513165204.

Contents

Chapter 6

Chapter 1

Introducing the IMS eWay

This document describes how to install and configure the IMS eWay Intelligent Adapter (called the IMS eWay throughout this document), as well as how to implement the eWay in a typical eGate environment.

This chapter provides a brief overview of operations and components, general features, and system requirements of the IMS eWay.

What's in This Chapter

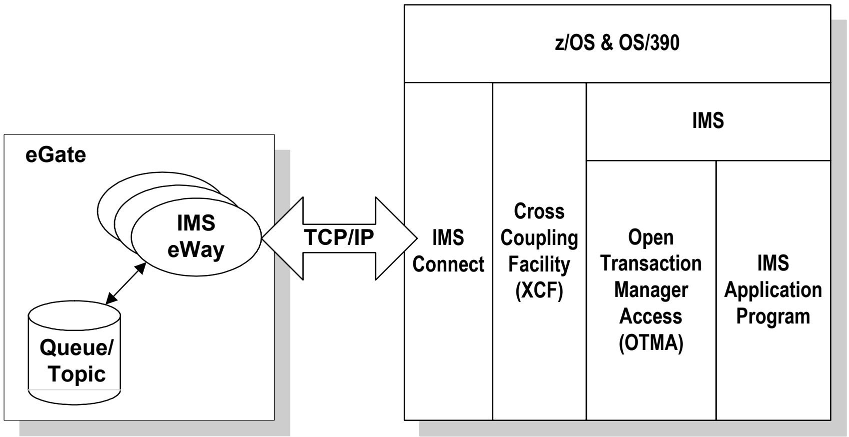
1.1 About Information Management System (IMS)

IBM's IMS, is a database and transaction management system that provides an interface for users to access information in various databases via on-line transactions. The IMS/TM (Transaction Manager) is a message-based transaction processor, that handles the execution of specific business application programs. The IMS/DB (Database) is an entirely separate component providing access to the IMS hierarchical database for applications running under the IMS/TM, as well as IMS transaction monitor and OS/390 batch jobs.

1.2 About the IMS eWay

The IMS eWay enables eGate to connect with IBM's IMS/TM mainframe applications through IBM's IMS Connect. (See **Figure 1 on page 8**).

Figure 1 IMS eWay and the IMS Environment.



The eWay provides access to the Input and Output Descriptors (MID/MOD) of the IMS applications without requiring changes to the application. By capturing the field contents before screen formatting, the eWay is not affected by cosmetic changes to the application's screen design.

The eWay includes the IMS Message Format Service (MFS) Wizard conversion utility to facilitate the creation of input and output Object Type Definitions (OTDs) from IMS MFS files.

The implementation of the IMS eWay is in accordance with IBM's *IMS Connect Guide and Reference*. These documents describe the OTMA protocol and contain important prerequisite information for the configuration IMS Connect on the mainframe.

A sample project for the IMS eWay is included on the installation CD-ROM which demonstrates how a non-conversational scenario (simple send/response) is managed.

1.3 About This Document

This section provides a brief outline of the IMS eWay User's Guide.

1.3.1. What's in This Document

This book includes the following chapters:

- **Chapter 1** **"Introducing the IMS eWay"** provides an overview of the IMS eWay Intelligent Adapter, including a brief description of the IMS. In addition, this chapter provides a brief outline of the eWay user's guide.
- **Chapter 2** **"Installing the IMS eWay"** lists both the supported operating systems and system requirements for the IMS eWay. It also includes directions for installing the IMS eWay as well as the IMS eWay documentation and sample projects.
- **Chapter 3** **"Configuring the IMS eWay"** describes the process of configuring the IMS eWay to run in your environment.

- **Chapter 4** **"Using the IMS eWay With eInsight"** describes how to use the IMS eWay with the ICAN Suite's eInsight Business Process Manager and the Web Services interface. Provides directions for importing and running the eInsight sample project and step by step directions for creating the sample project manually.
- **Chapter 5** **"Implementing an IMS eWay Project"** describes the features and functionality of the IMS eWay using the eGate Integrator and the Collaboration Editor (Java). It also provides directions for importing and running the JCS and MFS sample projects and step by step directions for creating the sample projects manually.

1.3.2. Scope

This user's guide provides a description of the IMS 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 IMS eWay Java methods, see the associated Javadoc.

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	▪ 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 ***bold italic***.	`bootstrap -p` ***password***
Hypertext links	**Blue** text	See **Document Conventions** on page 9
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*
- *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 IMS eWay

This chapter explains the procedures for installing the IMS eWay.

What's in This Chapter

2.1 Supported Operating Systems

The IMS eWay is available for the following operating systems:

- Windows 2000, Windows XP, and Windows Server 2003
- HP Tru64 5.1A
- HP-UX 11.0, 11i (PA-RISC), and 11i V2 (11.23)
- IBM AIX 5.1L and 5.2
- IBM z/OS V1.3 and V1.4
- Red Hat Enterprise Linux AS 2.1 (Intel x86)
- Red Hat Linux 8 (Intel x86)
- Sun Solaris 8 and 9

2.2 System Requirements

To set up and run the IMS eWay with the eGate Enterprise Designer, you need the following:

- A TCP/IP network connection

- Windows Server 2003, Windows 2000, or Windows XP (required for the User Interface)
- Microsoft Internet Explorer 6.0 or above

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

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

2.3 Supported External Applications

Software Requirements

- IMS Connect, Versions 2.1 and 2.2.

Note: *IMS Connect is an optional, purchasable, companion product to IMS. IMS Connect is not included with IMS.*

Software Requirements for IMS Connect:

- z/OS Version 1 Release 3 and z/OS Version 1 Release 4.
- IMS Versions 7, 8, or 9 with the required maintenance APARs applied. See the "IMS Connect and IMS Coexistence" section of the *IMS Connect Guide and Reference*, for more information about IMS and IMS Connect coexistence.
- Resource Access Control Facility (RACF) or an equivalent product.

Note: *This document uses the term RACF when referring to RACF (Resource Access Control Facility) or equivalent products.*

2.4 Before You Install

Open and review the **Readme.txt** file for the Oracle 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 IMS eWay

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

When the Repository is running on an HP NonStop Server, or UNIX operating system, eGate and the eWays are installed using the Enterprise Manager on a computer running Windows connected to the Repository server.

Installing the IMS eWay on an eGate supported system

The IMS eWay can be installed during or after the installation of the ICAN Suite. 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 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:
   - **IMSeWay.sar** (to install the IMS eWay)
   - **MFS.sar** (to install the MFS Wizard)
   - **FileeWay.sar** (to install the File eWay, used with the sample project)
3. Click on the**Browse** option of the Manifest File, browse to the Add-ons **Documentation** directory, select the **ProductsManifest.xml**, and click **Submit**. The available Add-on documentation list is now displayed.
4. From the **Documentation** directory, select and upload the following file:
   - **IMSeWayDocs.sar** (to upload the IMS eWay User's Guide, Javadoc, Readme, and sample projects to the Enterprise Manager)
5. Continue installation as directed in the SeeBeyond *ICAN Suite Installation Guide.*

Adding the eWay to an Existing ICAN Suite Installation

When installing the IMS eWay to an existing ICAN installation, complete the following steps:

1. Complete steps 1 through 5 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 Enterprise Designer (IDE) to complete the installation.

2.6 After Installation

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

Chapter 3

Configuring the IMS eWay

This chapter describes how to configure the IMS eWay properties, and provides a list of the eWay properties and their required values.

What's in This Chapter

3.1 Configuring the IMS eWay Properties

The IMS eWay includes a unique set of configuration parameters. After creating the eWays and the IMS External System in the Project's Environment, the property parameters can be modified for your specific system.

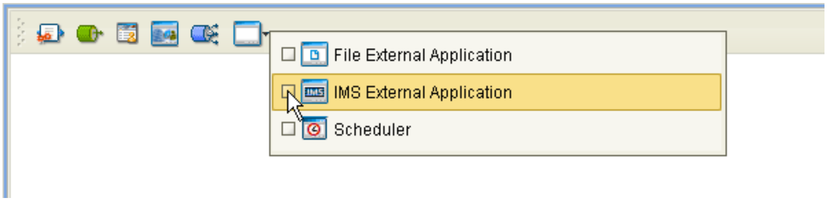
3.1.1 Selecting IMS as the External Application

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

To create the IMS External Application

1. From the Connectivity Map toolbar, click the External Applications icon.
2. Select the **IMS External Application** from the menu (see **Figure 2 on page 16**). The selected IMS External Application icon appears on the Connectivity Map toolbar.

Figure 2 External Applications Selection Menu

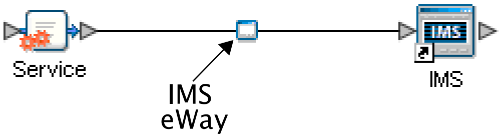


The new External System can now be dragged and dropped onto the Connectivity Map canvas and incorporated into a project.

3.1.2. Accessing the eWay Properties

When you connect an External Application to a Collaboration, the Enterprise Designer automatically assigns the appropriate eWay to the link (Figure 3). Each eWay is supplied with a template containing default configuration properties that are accessible from the Connectivity Map and Environment Explorer Tree.

Figure 3 eWay Location



3.1.3. Modifying the IMS eWay Properties

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

Modifying the eWay Connectivity Map Properties

The Connectivity Map parameters most commonly apply to a specific component eWay, and may vary from other eWays (of the same type) in the project.

1. From the Connectivity Map, double click the eWay icon, located in the link between the associated External Application and the Service.
2. The eWay **Properties Editor** opens with the IMS eWay Connectivity Map properties. Make any necessary modifications and click **OK** to save the settings.

Modifying the eWay Environment Properties

These parameters are globally common, applying to all eWays (of the same type) in the project. The saved properties are shared by all eWays for the specified External System.

1. From the Environment Explorer tree, right-click the IMS External System. Select **Properties** from the shortcut menu. The **Properties Editor** opens with the IMS eWay Environment properties.

2 Make any necessary modifications to the Environment properties, and click **OK** to save the settings.

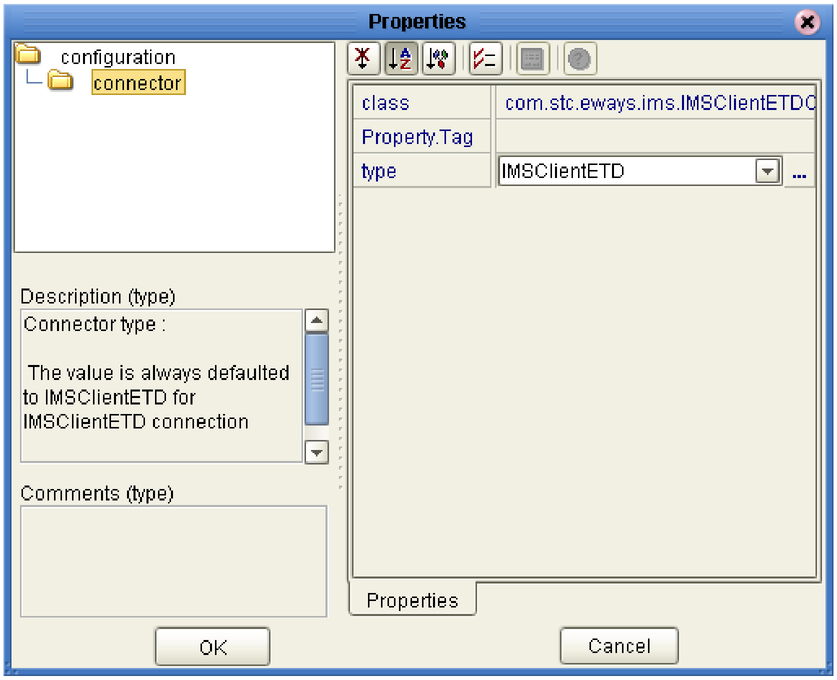
3.1.4. Using the Properties Editor

Modifications to the eWay properties are made using the IMS eWay Properties Editor.

Modifying the Default eWay Properties

1 From the upper-left pane of the Properties Editor, select a subdirectory of the configuration properties tree. The properties contained in that subdirectory are now displayed in the right pane of the Properties Editor. For example, click on the **connector** subdirectory to display the editable properties, as shown in Figure 4

Figure 4 Properties Editor - IMS eWay parameters



class	com.stc.eways.ims.IMSClientETDC
Property.Tag	
type	IMSClientETD

Description (type)

Connector type :

The value is always defaulted to IMSClientETD for IMSClientETD connection

Comments (type)

2 Click on any property field to make it editable. For example, click on the **type** property to edit its settings. If a parameter's value is true/false or multiple choice, the field reveals a submenu of property options.

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

3 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.

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

5 Click **OK** to close the Properties Editor and save the changes.

3.2 IMS eWay Connectivity Map Properties

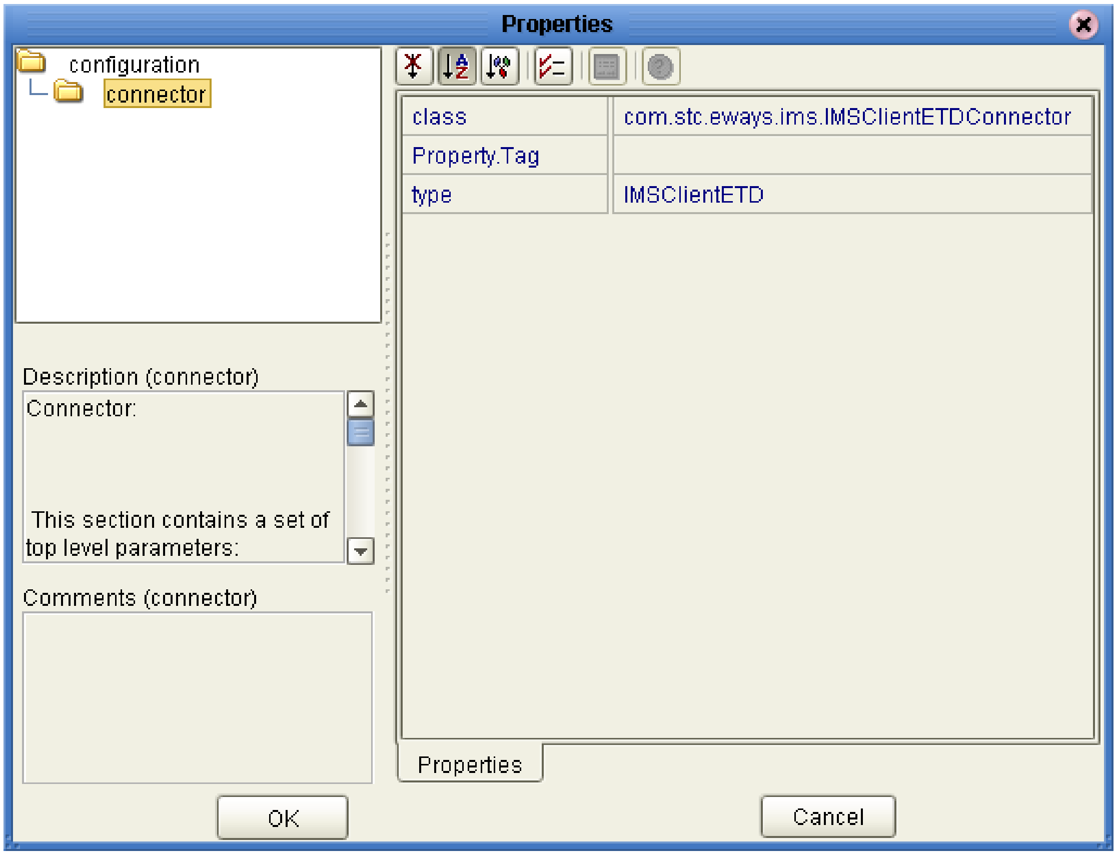
The IMS eWay configuration parameters, accessed from the Connectivity Map, are organized into the following section:

- **Connector** on page 18

3.2.1. Connector

The Connector section contains the top level parameters displayed in Figure 5.

Figure 5 Properties Editor - IMS eWay Connector parameters



Description (connector)

Connector:

This section contains a set of top level parameters:

Comments (connector)

class	com.stc.eways.ims.IMSClientETDConnector
Property.Tag	
type	IMSClientETD

Class

Description

Specifies the class name of the ETD connector object.

Required Values

A valid package name. The default is **com.stc.eways.ims.IMSClientETDConnector**.

Property.Tag

Description

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

Required Values

A valid data source package name.

Type

Description

Specifies the connector type.

Required Values

IMSClientETD by default for IMSClientETD connections.

3.3 IMS eWay Environment Properties

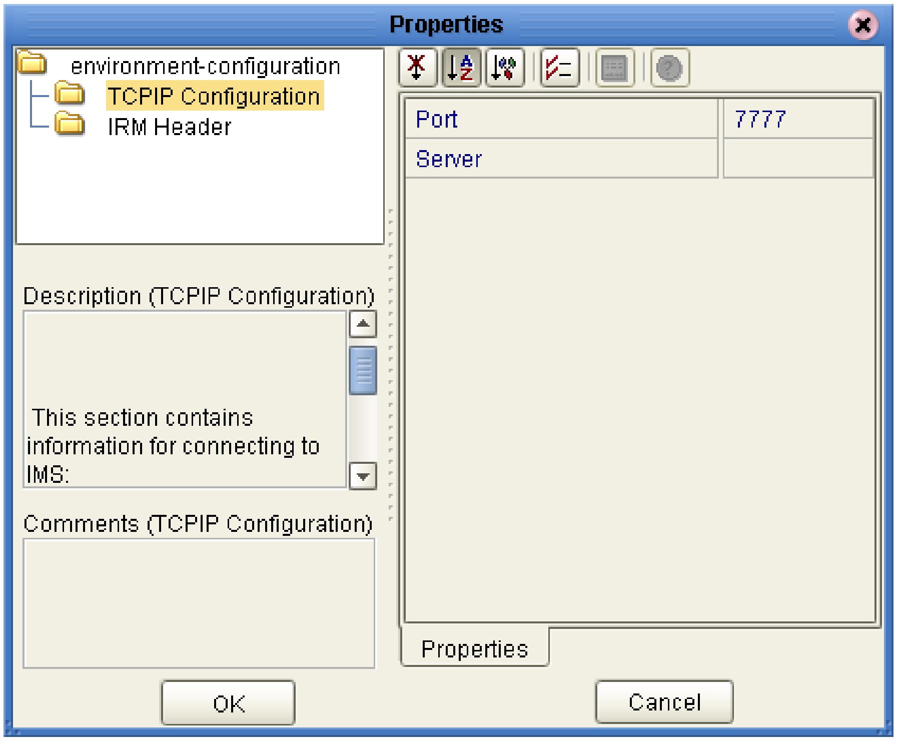
The IMS eWay configuration parameters, accessed from the Environment Explorer tree, are organized into the following sections:

- **TCPIP Configuration** on page 19
- **IRM Header** on page 21

3.3.1. TCPIP Configuration

The TCPIP Configuration section contains information for connecting to the Portal Infranet. This section contains the top level parameters, as displayed in Figure 6.

Figure 6 Properties Editor - IMS eWay TCPIP Configuration parameters


Properties
environment-configuration
TCPIP Configuration
IRM Header
Description (TCPIP Configuration)
Comments (TCPIP Configuration)
Properties
OK
Cancel

This section contains information for connecting to IMS:

Port	7777
Server	

Port

Description

Specifies the port that IMS Connect is listening on. This parameter is mandatory.

Required Values

A number indicating the port on which IMS Connect is listening.

Server

Description

Specifies the name of the server host. This parameter is mandatory.

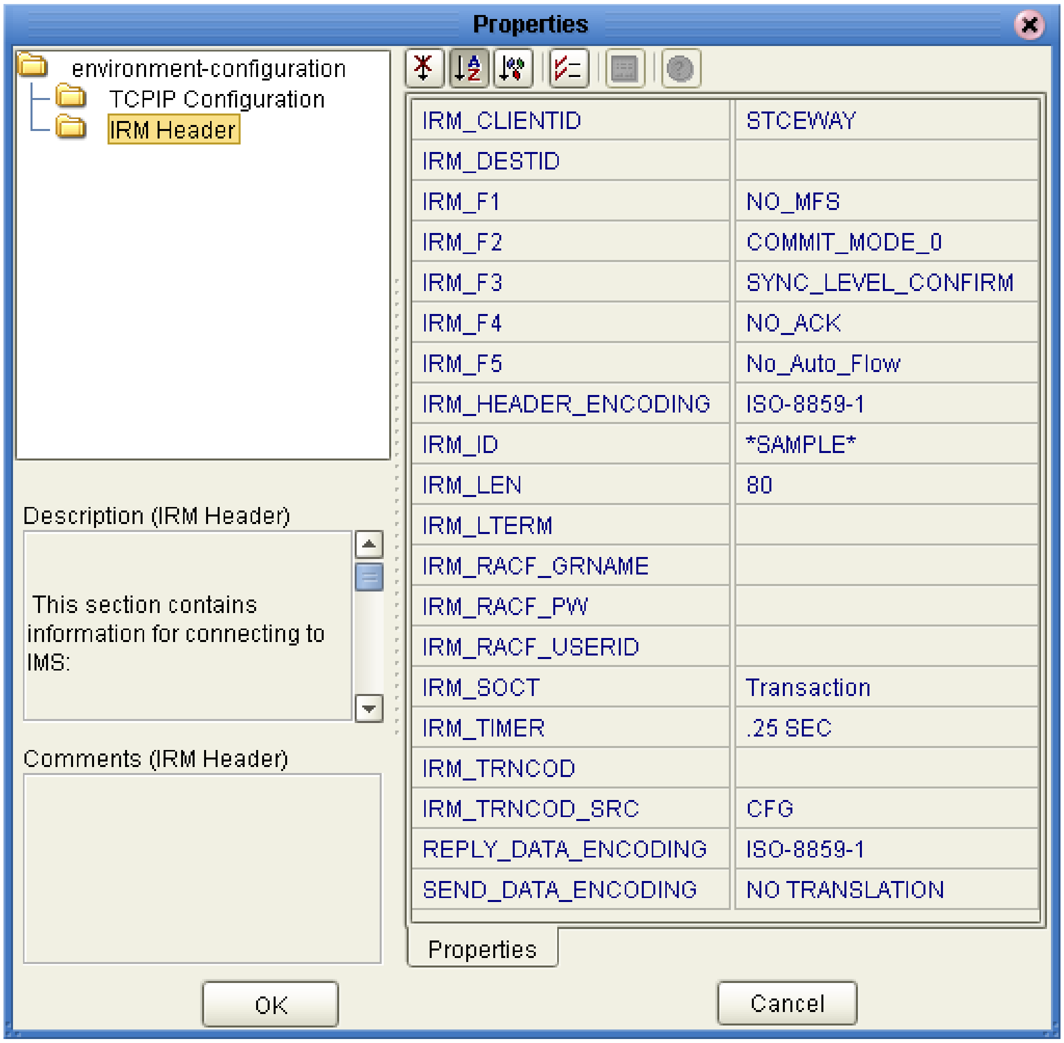
Required Values

The server host name.

3.3.2. IRM Header

The IRM (IMS Request Message) Header section contains the top level parameters: displayed in Figure 7.

Figure 7 Properties Editor - IMS eWay IRM Header parameters



This section contains information for connecting to IMS:

IRM_CLIENTID	STCEWAY
IRM_DESTID	
IRM_F1	NO_MFS
IRM_F2	COMMIT_MODE_0
IRM_F3	SYNC_LEVEL_CONFIRM
IRM_F4	NO_ACK
IRM_F5	No_Auto_Flow
IRM_HEADER_ENCODING	ISO-8859-1
IRM_ID	*SAMPLE*
IRM_LEN	80
IRM_LTERM	
IRM_RACF_GRNAME	
IRM_RACF_PW	
IRM_RACF_USERID	
IRM_SOCT	Transaction
IRM_TIMER	.25 SEC
IRM_TRNCOD	
IRM_TRNCOD_SRC	CFG
REPLY_DATA_ENCODING	ISO-8859-1
SEND_DATA_ENCODING	NO TRANSLATION

Note: *For a full description of the IRM header, see IBM's IMS Connect Guide and Reference (SC27-0946-00).*

IRM_CLIENTID

Description

Specifies the name of the client ID (character string) that is used by IMS Connect. If this string is not supplied from the client, then the user exit must generate it. The client ID is returned to IMS Connect from the exit in the EXIT PARMLIST field, EXPREA_CLID.

Required Values

The client ID used by IMS Connect. The configured default is STCEWAY.

IRM_DESTID

Description

Specifies the Datastore name (IMS destination ID). This field must be specified by the client.

Required Values

String-set. A Datastore name/IMS destination ID (character string).

IRM_F1 (MFS MOD Names)

Description

Specifies whether the MFS Message Output Descriptor (MOD) is returned as part of the output.

- **MFS**: The user requests that MFS MOD name be returned.
- **NO_MFS**: The user requests that no MFS MOD name be returned.

If this value is not supplied by the client, the user exit must use a default value. The MFS mod name flag is returned to IMS Connect from the exit in the EXIT PARMLIST field, EXPREA_UFLAG1.

When MFS is specified, a Request Mod Message (RMM) is returned as the first structure of the output message. This structure contains an ID of *REQMOD* followed by the MFS MOD name. For details, see IBM's *IMS Connect Guide and Reference*, (SC27-0946-00), page 59.

Required Values

MFS or **NO_MFS**. The configured default is **NO_MFS**.

IRM_F2 (COMMIT MODE)

Description

Specifies the **Commit Mode**. If this value is not supplied from the client, the user exit must use a default value. The MFS mod name flag is returned to IMS Connect from the exit in the EXIT PARMLIST field, EXPREA_UFLAG1.

- **COMMIT_MODE_0**
- **COMMIT_MODE_1**

Required Values

COMMIT_MODE_0 or **COMMIT_MODE_1**. The configured default is **COMMIT_MODE_0**.

Note: *For a full description of the IRM header, see IBM's IMS Connect Guide and Reference (SC27-0946-00).*

IRM_F3 (Sync Level)

Description

Specifies the Sync Level. If this value is not supplied from the client, the user exit must use a default value.

- **SYNC_LEVEL_CONFIRM**: Must be used when the IRM_F2 parameter (commit mode) is set to COMMIT_MODE_0.
- **SYNC_LEVEL_NONE**: No Sync level.

The commit mode flag is returned to IMS Connect from the exit in the OTMA header field, OMHDRSLV.

Required Values

SYNC_LEVEL_CONFIRM or **SYNC_LEVEL_NONE.** If the **IRM_F2** property is set to **COMMIT_MODE_0**, the Sync level must be set to **SYNC_LEVEL_CONFIRM**. The configured default is **SYNC_LEVEL_CONFIRM**.

IRM_F4 (ACK/NAK/ Response)

Description

Specifies the ACK/NAK (positive/negative acknowledgement) response expression sent to IMS Connect and forwarded to IMS. When the value is received and passed to the user exit, the exit builds the appropriate OTMA structure and returns it to IMS Connect. The ACK/NAK/DEALLOCATE/RESUME [A/N/D/R] response must be sent to IMS Connect with no data element.

- **NO_ACK**: No request for acknowledgment or deallocation. When a response mode transaction or conversational transaction is being sent to IMS Connect, IRM_F4 must be set to NO_ACK.
- **ACK**: Positive acknowledgment, used in response to a message sent to the client where the SYNC level is set to CONFIRM (SYNC _LEVEL_CONFIRM).
- **DEALLOCATE**: Deallocate connection. Used to terminate a conversation before the conversation is complete.
- **NACK**: Negative acknowledgment. Used in response to a message sent to the client where the SYNC level is set to CONFIRM (SYNC _LEVEL_CONFIRM).
- **RESUME**: Resume TPIPE. Used to request Asynchronous output data from IMS. Resume must execute on a transaction socket as COMMIT_MODE_0.
- **SENDONLY**: Send only, used for a non-response transaction and for sending data to IMS. SENDONLY must execute as COMMIT_MODE_0.

Required Values

Select one of the six options. The configured default is **NO_ACK**.

IRM_F5 (Flow Control)

Description

Specifies Flow Control properties.

Note: *SeeBeyond recommends using the default value* ***No_Auto_Flow****. Contact SeeBeyond Support before using any value other than* ***No_Auto_Flow****.*

- **Client_Translation**: Translation is done by the client.
- **Single_Message**: Returns only one message on receive following the resume TPIPE.
- **No_Auto_Flow**: No message auto flow (see meaning for No_Auto_Flow_Out).
- **Auto_Flow_Out**: Auto message flow. Returns all current messages, one at a time, and waits on the last receive for the next message for IRM_TIMER value. Set the IRM_TIMER high. Use this only for a dedicated output client.
- **No_Auto_Flow_Out**: No message auto flow. Returns all current messages one at a time, and waits on the last receive for the next message for IRM_TIMER value. Set the IRM_TIMER low. Use this only for a dedicated output client. This value is similar to Auto_Flow_Out, as described above, except that the IRM_TIMER causes the last receive to terminate.

Required Values

The recommended setting is No_Auto_Flow (default).

IRM_HEADER_ENCODING

Description

Specifies the encoding of the IRM Header properties sent to IMS Connect.

- Set the value to **ISO-8859-1** if the message body is ASCII text. The IMS Connect *SAMPLE* user exit converts the data to EBCDIC.
- Set the value to an EBCDIC code set, such as **cp500**, if the message is EBCDIC text or binary data. No data translation occurs.

Required Values

ISO-8859-1 for ASCII transaction content, or an EBCDIC code, such as **cp500**, for EBCDIC transaction content.

IRM_ID

Description

Specifies the identifier (character string) of the user exit that is driven after the complete message is received.

Note: *In a program, an exit is used to move from the called routine back to the calling routine. A routine can have more than one exit point, thus allowing termination based on various conditions.*

The following IDs are used by the IMS Connect-supplied user message exits:

- ***IRMREQ*** (for HWSIMSO0)
- ***SAMPLE*** (for HWSSMPL0)
- ***HWSJAV*** (for HWSJAVA0)

Required Values

The appropriate identifier character string. The configured default is ***SAMPLE***.

IRM_LEN

Description

Specifies the length of the IRM structure. The user written exits minimum size is **36**. HWSIMSO0 and HWSSMPL0 have a minimum IRM length of **80**.

Required Values

An integer indicating valid IRM structure length. The configured default is **80**.

IRM_LTERM

Description

Specifies the IMS LTERM override name. This field can be set to a name or blank.

Required Values

The appropriate LTERM name or blank.

IRM_RACF_GRNAME

Description

Specifies the RACF Group Name. The client must provide the RACF group name if RACF is to be used. The exit returns the RACF group name to IMS Connect from the OTMA header field, OMSECGRP.

Required Values

The appropriate RACF group name.

IRM_RACF_PW

Description

Specifies the RACF PASSTICKET. The client must provide the RACF PASSTICKET if RACF is to be used. The exit returns the PASSTICKET value to IMS Connect from the OTMA header field, OMUSR_PASSTICK.

Required Values

The appropriate RACF PASSTICKET.

IRM_RACF_USERID

Description

Specifies the RACF User ID. The client must provide the RACF user ID if RACF is used. The exit returns the RACF user ID name to IMS Connect from the OTMA header field, OMSECUID.

Required Values

A valid RACF user ID.

IRM_SOCT

Description

Specifies the socket connection type.

- **Transaction**: Transaction socket. The socket connection lasts across a single transaction.
- **Persistent**: Persistent socket. The socket connection lasts across multiple transactions.
- **Non_Persistent**: Non-persistent socket. The socket connection lasts for a single exchange consisting of one input and one output. Do not use Non_Persistent when implementing conversational transactions because this type causes multiple connects and disconnects.

Required Values

- Select one of the three options. The configured default is **Transaction**.

IRM_TIMER

Description

Specifies the time delay for the receive to the Datastore after an ACK or RESUME TPIPE.

- **.25 SEC**: .25 seconds.
- **No_Wait**: Timer is not set and no delay occurs.
- **Block**: The receive waits indefinitely. This setting is used to support the *Auto* option of the asynchronous output function.

Required Values

Select one of the three options. The configured default is **.25 SEC**.

IRM_TRNCOD

Description

Specifies the default IMS transaction code.

Required Values

A valid transaction code.

IRM_TRNCOD_SRC

Description

Specifies where the transaction code is taken.

- **CFG**: The transaction code is to be taken from the configuration file.
- **MESSAGE**: the transaction code is the first 8 bytes of the message.

Required Values

Select one of the two options. The configured default is CFG.

REPLY_DATA_ENCODING

Description

Specifies the encoding of the message body received back from IMS Connect.

- Set to **ISO-8859-1** if the message text is ASCII.
- Set to an EBCDIC code, such as **cp500**, if the return message is EBCDIC and/or no content translation is needed.
- If the content set is a double-byte character, such as Japanese, set the appropriate EBCDIC code page for that language (for example, **cp930** for Japanese).

Required Values

The appropriate code page:

- For ASCII transactions, enter **ISO-8859-1.**
- For EBCDIC transactions, enter an EBCDIC code, such as **cp500.**
- For double-byte character sets, enter the appropriate code page for that language (for example, **cp390** for Japanese).

SEND_DATA_ENCODING

Description

Specifies the encoding translation (if any) to apply to the message body sent to IMS Connect.

- Set to **NO TRANSLATION** to send the message body to IMS Connect without translation, or when using the ***SAMPLE*** user exit when the IRM Headers and message body are in ASCII.
- Set to an EBCDIC code, such as **cp500**, to translate the message body from ASCII to EBCDIC before sending to IMS Connect.
- If the content is a double-byte character set such as Japanese, set to the EBCDIC code page for that language (for example, **cp930** for Japanese).

Required Values

Enter **NO TRANSLATION** or the appropriate code page as follows:

- Enter **NO TRANSLATION** when using the ***SAMPLE*** user exit and IRM Headers and message content is in ASCII.
- Enter an EBCDIC code, such as **cp500**, to translate ASCII message content to EBCDIC before sending it to IMS Connect.
- For double-byte character sets, enter the appropriate code page for that language (for example, **cp390** for Japanese).

3.4 Alerting and Logging

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

The alerts/status notifications for the IMS eWay are currently limited to:

- Started
- Running
- Stopping
- Stopped

Chapter 4

Using the IMS eWay With eInsight

This chapter describes how to use the IMS eWay with 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

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:

- 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. Once 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 IMS eWay With eInsight

An eInsight Business Process Activity can be associated with the IMS eWay during the system design phase. To make this association, select the desired operators under the eWay (in the Enterprise Explore) and drag it onto the eInsight Business Process Designer canvas. Currently, the IMS eWay has only the **IMSRequest** operator available.

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 IMS eWay.

4.3 IMS eWay Considerations

The following error message could appear in the mainframe log, even when the IMS eWay runs successfully:

HWSP1435E Socket closed; Request message incomplete; M=SDRC

This is an erroneous error message that originates from IBM's IMS Connect program. The error message is resolved by the following PTFs from IBM:

- PQ53848 (UQ68076) for IMS Connect 1.1
- PQ62496 (UQ68078) for IMS Connect 1.2

4.4 IMS eWay eInsight Sample Project

The IMS eInsight Business Process carries out the same tasks as the **IMS_Collab** Collaboration used in IMS_BP_Sample project.

To create a sample project using an IMS eInsight Business Process, follow the directions for the **IMS_JCE_Sample** project (see **The IMS_JCE_Sample Project** on page 45), but rather than creating and inserting the Java Collaboration, simply drag and drop the completed Business Process into the Connectivity Map, substituting it for the **IMS_Collab** Collaboration used in the existing project (see **Creating Collaboration Bindings** on page 52). Then complete and run the project as directed

4.5 Importing a Sample Project

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

1. The sample files are uploaded with the eWay's documentation .sar file and downloaded from the Enterprise Manager's **Documentation** tab. Extract the samples from the Enterprise Manager to a local file.
2. Save all unsaved work in Enterprise Designer before importing a sample project.
3. From the Enterprise Designer's Project Explorer pane, right-click the Repository and select **Import** from the shortcut menu. The **Import Manager** appears.
4. Browse to the directory that contains the sample project zip file. Select the sample file (for this project, **IMS_BP_Sample.zip**) and click **Import**. After the sample project is successfully imported, click **Close**.

5. Before an imported sample project can be run you must do the following:
   - Create an **Environment** (see **Creating an Environment** on page 37)
   - Configure the eWays properties for your specific system (see **Configuring the eWays** on page 38)
   - Create a **Deployment Profile** (see **Creating and Activating the Deployment Profile** on page 40)

The following pages provide step by step directions for manually creating the IMS_BP_Sample project.

4.6 The IMS_BP_Sample Project

The **IMS_BP_Sample** project demonstrates connectivity with an external IMS system.

Sample Project Overview

1. The inbound File eWay receives a file from an external directory.
2. This file is published by the IMS eWay to an external IMS application.
3. The IMS eWay receives the file back from the external IMS application, and publishes the file to the outbound File eWay.

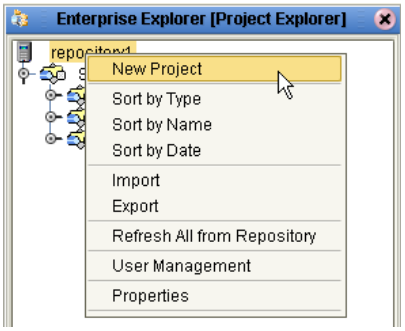
The outbound File eWay publishes the file to an external directory.

4.6.1. Creating a Project

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

1. Start the Enterprise Designer.
2. From the Enterprise Explorer’s Project Explorer tab, right-click the Repository and select **New Project** (see **Figure 8 on page 31**). A new project (Project1) appears on the Project Explorer tree.

Figure 8 Enterprise Explorer - New Project



3. Click twice (not a double-click) on **Project1** and rename the project (for this sample, **IMS_BP_Sample**).

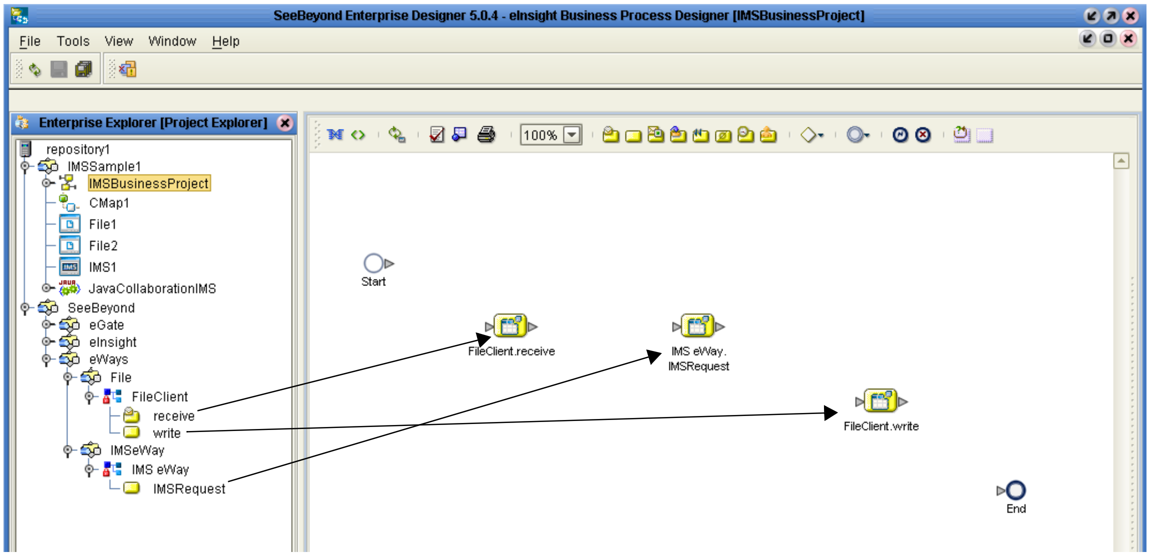
4.6.2 Creating a Business Process

To create the IMS eInsight Business Process, do the following:

Creating the Business Process Flow

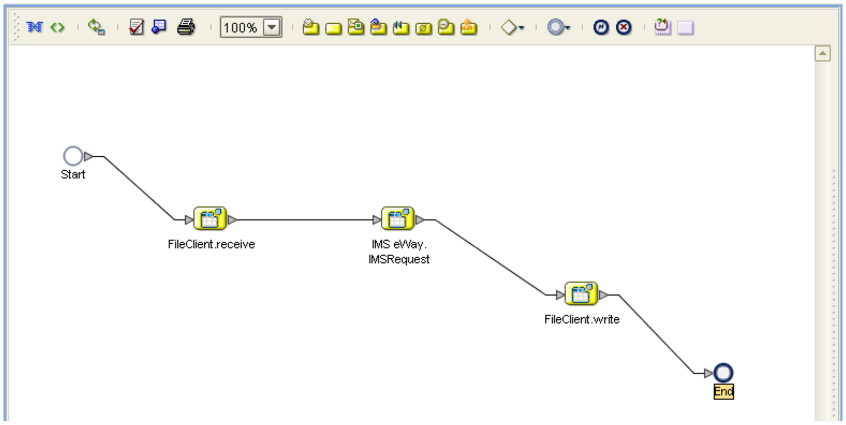
1. Right-click the existing project (for this sample **IMS_BP_Sample**) in the Project Explorer tree, and select **New** > **Business Process** from the shortcut menu. The eInsight Business Process Designer appears and **IMS_BP** is added to the tree.
2. From the Project Explorer tree, expand both of the following **eWays** nodes (as displayed in Figure 9):
   - IMSeWay > IMS_eWay
   - File > FileClient
3. Populate the eInsight Business Process Designer's modeling canvas with the following activities from the Project Explorer tree (as displayed in Figure 9):
   - **receive**, under SeeBeyond > eWays >File > FileClient
   - **IMSRequest**, under SeeBeyond > eWays > IMSeWay > IMS_eWay
   - **write**, under SeeBeyond > eWays >File > FileClient

Figure 9 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 10.
   - Start -> FileClient.receive
   - FileClient.receive -> IMS_eWay.IMSRequest
   - IMS_eWay.IMSRequest ->FileClient.write
   - FileClient.write -> End

Figure 10 eInsight Business Process Designer - Link the Modeling Elements



Configuring the Modeling Elements

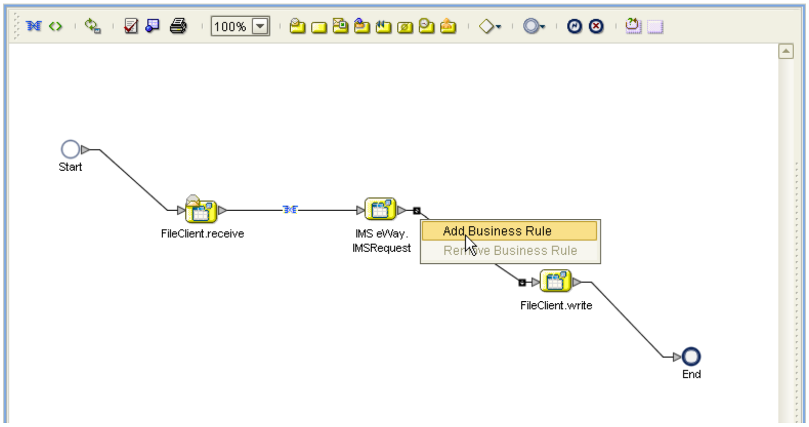
Business Rules, created between the Business Process Activities, allow you to configure the relationships between the input and output attributes of the Activities using the Business Process Editor's Business Rules Designer.

Adding Business Rules

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

2 Repeat step 1 for the link between **IMS_eWay.execute** and **JMS.send** (see Figure 11).

Figure 11 eInsight Business Process Designer - Adding Business Rules



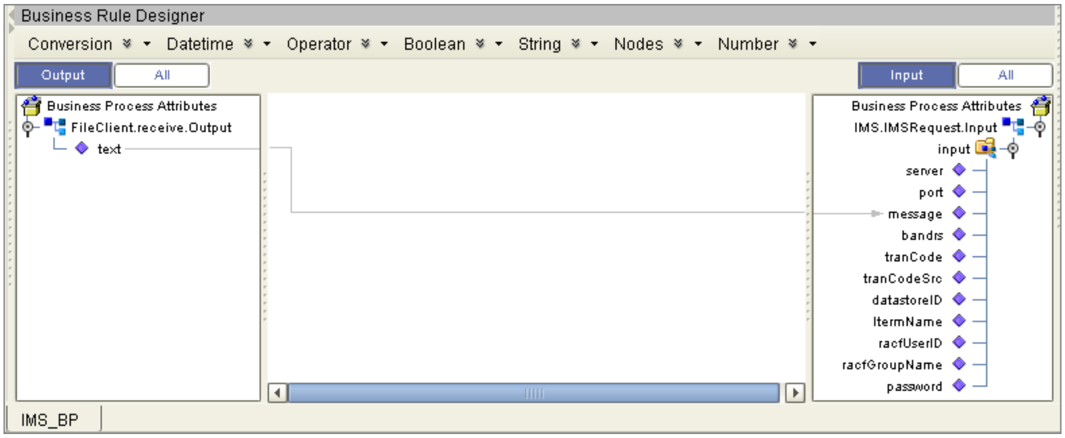
Using the Business Rules Designer

1 From the eInsight Business Process Designer toolbar, click the **Display Business Process Attributes** button. The Business Rules Designer appears at the bottom of the eInsight Business Process Designer.

2 Click on the **Business Rule** icon in the link between **FileClient.receive** and **IMS_eWay.IMSRequest** to display the Business Rule Input and Output Attributes in the Business Rules Designer. These attributes can now be modified.

3 Map **text**, under FileClient.receive.Output in the Output pane, to **message** under IMS eWay.IMSRequest.Input > input in the Input pane. This is done by clicking on **text** and dragging the cursor to **message** (see Figure 12).

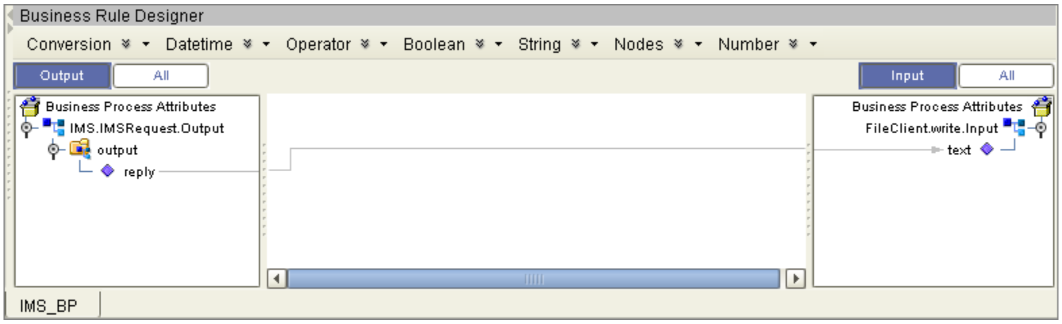
Figure 12 eInsight Business Rules Designer



4 To configure the second Business Rule, click on the Business Rule in the link between the **IMS_eWay.IMSRequest** and **FileClient.write** Activities. This Business Rule is now displayed in the Business Rules Designer.

5 Map **reply**, under **IMS_eWay.IMSRequest.Output > output** in the Output pane of the Business Rules Designer, to **text**, under **FileClient.write.Input** in the Input pane as displayed in Figure 13.

Figure 13 eInsight Business Rules Designer



6 When the Business Process is complete, from the eInsight Business Process Designer toolbar, click the **Synchronize Graphical Model and Business Process Code** icon to synchronize the graphical interface to the Business Process code.

7 Save your changes to the Repository.

4.6.3 Creating a Connectivity Map

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

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

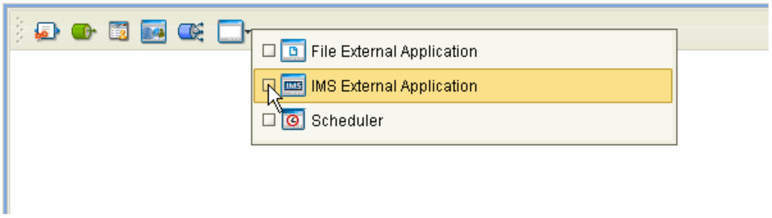
2 The New Connectivity Map appears and a node for the Connectivity Map is added under the project on the Project Explorer tree labeled **CMap1**. Rename the Connectivity Map **IMS_BP_Sample_CM**.

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

Selecting the External Applications

In the Connectivity Map, the eWays are associated with External Systems. For example, to establish a connection to IMS, you must first select IMS as an External System to use in your Connectivity Map (see Figure 14).

Figure 14 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, **IMS** and **File**). Icons representing the selected external systems are added to the Connectivity Map toolbar.

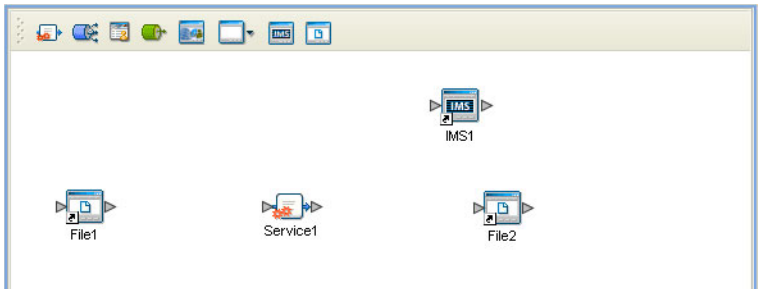
Populating the Connectivity Map

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

1 For this sample, drag the following components onto the Connectivity Map canvas as displayed in **Figure 15 on page 36**:

- **File External System** (2)
- **Service** (A service is a container for Java Collaborations, Business Processes, eTL processes, and so forth) This can also be created by dragging the Business Process over from the Project Explorer tree.
- **IMS External System**

Figure 15 Connectivity Map with Components



2 Rename the Connectivity Map objects by right-clicking the object's name, selecting **Rename** from the shortcut menu, and typing the new name. Rename the objects as follows:

- **File1** to **FileIn**
- **Service1** to **IMS_BPEL_Service**
- **IMS1** to **IMS**
- **File2** to **FileOut**

3 Save your changes to the Repository.

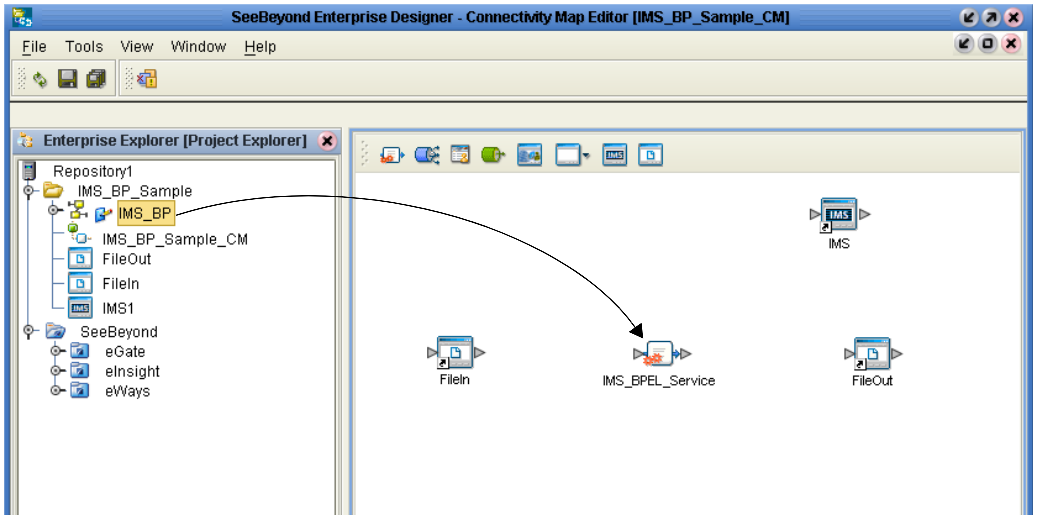
4.6.4. Binding the eWay Components

After the Business Processes have been written, the components are associated and bindings are created in the Connectivity Map.

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

2 Drag and drop the **IMS_BP** Business Process from the Project Explorer onto the Service (**IMS_BPEL_Service**). If the Business Process was successfully associated, the Service's icon changes to a Business Process icon (see Figure 16).

Figure 16 Connectivity Map - Binding the eWay Components



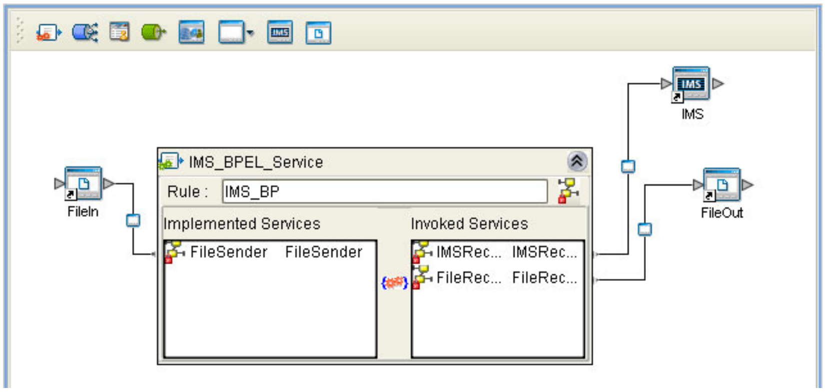
3 Double-click **IMS_BPEL_Service**. The **IMS_BPEL_Service Binding dialog box** appears using the **IMS_BP** Rule.

4 From the **IMS_BPEL_Service** Binding dialog box, drag **FileSender** (under Implemented Services) to the **FileIn** (File) External Application.

5 From the **IMS_BPEL_Service** Binding dialog box, drag **IMS_Receiver** (under Invoked Services) to the **IMS** External Application.

6 From the **IMS_BPEL_Service** Binding dialog box, drag **FileReceiver** to the **FileOut** External Application (see Figure 17). Minimize the **IMS_BPEL_Service** Binding dialog box by clicking the chevrons in the upper-right corner.

Figure 17 Connectivity Map - Associating (Binding) the Project's Components



Rule : IMS_BP

7 Save your current changes to the Repository.

4.6.5. Creating an Environment

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

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

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

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

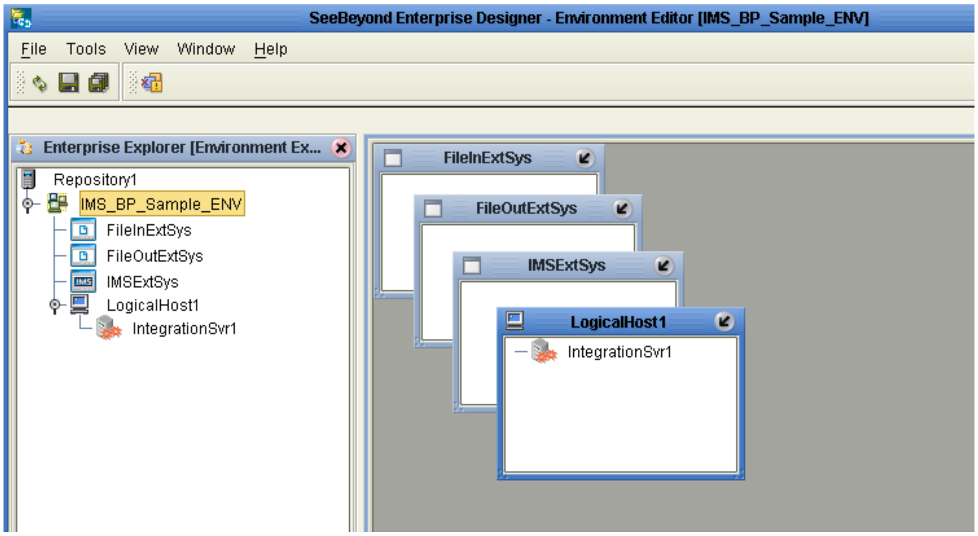
4 Right-click **IMS_BP_Sample_ENV** and select **New IMS External System**. Name the External System **IMSExtSys**. Click **OK**. **IMSExtSys** is added to the Environment Editor.

5 Right-click **IMS_BP_Sample_ENV** and select **New File External System**. Name the External System **FileExtSysIn** and select **Inbound File eWay** as the **External System Type**. Click **OK**. **FileExtSysIn** is added to the Environment Editor.

6 Right-click **IMS_BP_Sample_ENV** and select **New File External System**. Name this External System **FileExtSysOut** and select Outbound File eWay as the **External System Type**. **FileExtSysOut** is added to the Environment Editor.

7. Right-click **IMS_BP_Sample_ENV** 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 **LogicalHost1** and select **New SeeBeyond Integration Server**. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under LogicalHost1.
9. Save changes to the repository. The Environment Explorer and Environment Editor now appear as displayed in Figure 18.

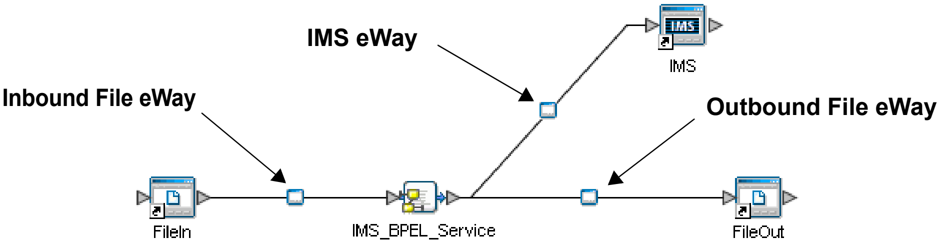
Figure 18 Environment Editor



4.6.6. Configuring the eWays

The IMS_BP_Sample project uses three eWays, each represented in the Connectivity Map as a node between an External Application and a Business Process (see Figure 19). eWays facilitate communication and movement of data between the external applications and the eGate system.

Figure 19 eWays



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

Configuring the File eWays

1. Double-click the inbound **File eWay**, select **Inbound File eWay** in the Templates dialog box and click **OK**.
2. The **Properties Sheet** opens to the inbound File eWay properties. Modify the properties for your system, including the settings in Table 2, and click **OK**.

Table 2 Inbound File eWay Settings

Inbound eWay Connection Parameters	
Directory	C:/temp
Input file name	Input*.txt

3. In the same way, modify the outbound File eWay configuration for your system, including the settings in Table 3, and click **OK**.

Table 3 Outbound File eWay Settings

Outbound eWay Connection Parameters	
Directory	C:/temp
Output file name	output%.dat

Configuring the IMS eWay

The IMS eWay properties must be set in both the Project Explorer and Environment Explorer. For more information on the IMS eWay properties and the Properties Sheet, see **Configuring the eWay Properties** on page 15 or see the *eGate Integrator User's Guide*.

For the IMS_BP_Sample project, do the following:

Modifying the IMS eWay Connectivity Map Properties

1. From the **Connectivity Map**, double-click the **IMS eWay**. The **Properties Sheet** opens to the IMS eWay project configuration properties.
2. Modify the IMS eWay (Project Explorer) configuration for your system, including the settings in Table 4, and click **OK**.

Table 4 IMS Project Explorer eWay Settings

IMS eWay Connectivity Map Properties	
Connector Set as directed, otherwise use the default settings	
Class	com.stc.eways.IMS.IMSClientConnector
Property.Tag	The data source
Type	IMSClientOTD

Modifying the IMS eWay Environment Explorer Properties

1. From the **Environment Explorer** tree, right-click the IMS External System (**IMSExtSys** in this sample), and select **Properties** from the shortcut menu. The **Properties Sheet** opens to the IMS eWay Environment Explorer properties.
2. Modify the IMS eWay Environment Explorer properties for your system, including the settings in Table 5, and click **OK**.

Table 5 IMS Environment Explorer eWay Settings

IMS eWay Environment Explorer Properties	
TCPIP Configuration Set as directed, otherwise use the default settings	
Port	*7777*
Server	*Server host name*
IRM Header Set as directed, otherwise use the default settings.	
IRM_CLIENTID	*Name of the client ID used by IMS Connect*
IRM_DESTID	*The Datastore name*
IRM_RACF_GRNAME	*The RACF group name - if RACF is to be used*
IRM_RACF_PW	*The RACF PASSTICKET - if RACF is to be used*
IRM_RACF_USERRID	*The RACF user ID - if RACF is to be used*

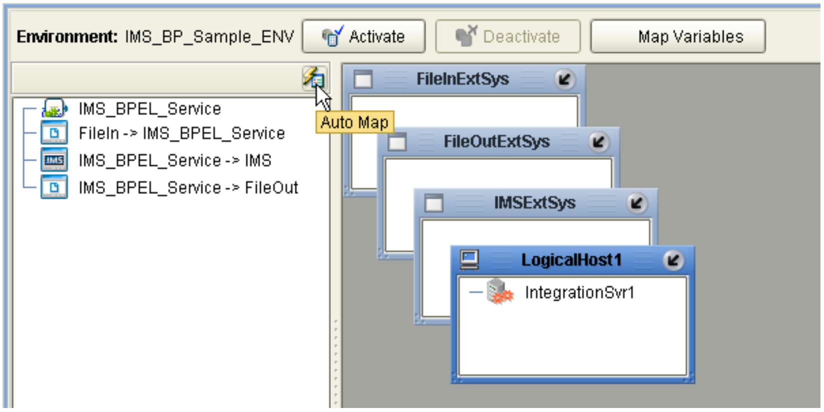
4.6.7 Creating and Activating the Deployment Profile

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

To create the IMS_BP_Sample Deployment Profile do the following:

1. From the Enterprise Explorer's Project Explorer, right-click the project (**IMS_BP_Sample**) and select **New** > **Deployment Profile** from the shortcut menu.
2. Enter a name for the Deployment Profile (for this project, **IMS_BP_Sample_DP**). Make sure that the selected Environment is **IMS_BP_Sample_ENV**. Click **OK**. The Deployment Profile Editor appears.
3. Click the **Auto Map** icon as displayed in **Figure 20 on page 41**. The projects components are automatically mapped to their system window as seen in **Figure 21 on page 41**. 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 8.

Figure 20 Deployment Profile - Auto Map



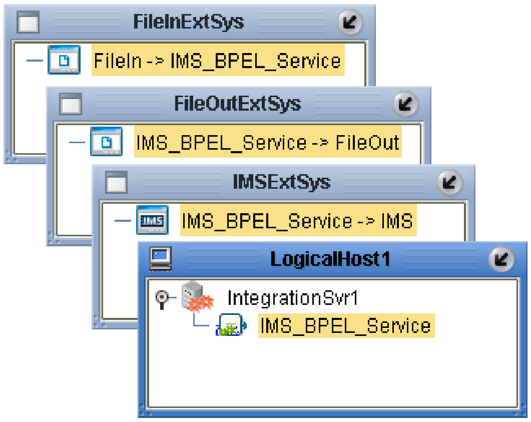
4 From the left pane of the Deployment Editor, drag the **FileIn -> IMS_BPEL_Service** (external application) object to the **FileExtSysIn** window.

5 Drag the **IMS_BPEL_Service -> FileOut** (external application) object to the **FileExtSysOut** window.

6 Drag the **IMS_BPEL_Service -> IMS** (external application) to the **IMSExtSys** window.

7 Drag the **IMS_BPEL_Service** Business Process to **IntegrationSvr1** in the **LogicalHost1** window (see Figure 21).

Figure 21 Deployment Profile



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

4.6.8. Running the Project

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

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

2 Extract the file to the **C:\ican50\logicalhost1** directory. You must specify the **logicalhost1** directory for it to be created.

3 Navigate to the **C:\ican50\logicalhost1\logicalhost\bootstrap\config** directory and open the logical-host.properties file using Notepad™.

4 Enter and save the following information in the appropriate fields:

- Logical Host root directory: **C:\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: **IMS_BP_Sample_Env**
- Logical Host name: **logicalhost1**

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

5 Run the **bootstrap.bat** file located in the **C:\ican50\logicalhost1\logicalhost\bootstrap\bin** directory.

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

Chapter 5

Implementing an IMS eWay Project

This chapter provides an introduction to the IMS eWay components and information on 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

5.1 IMS eWay Components

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

IMS eWay Configuration File

The properties file for the IMS eWay contains the parameters that are used to connect with a specific external system. These parameters are set using the Properties Sheet. For more information about the IMS eWay properties File and the Properties Sheet see **Creating and Configuring the IMS eWay** on page 15.

IMSClientETD OTD

The IMSClientETD OTD is provided with the eWay and contains methods and attributes that are used to create the Business Rules that invoke the IMS program. The nodes of the OTD take their initial value from the eWay Connection configuration parameters, but can be changed in the Collaboration.

IMS MFS Wizard

The IMS MFS Wizard builds an Object Type Definition from an IMS message format service (MFS) file. The wizard generates Object Type Definitions (OTDs) that map input and output message segments at the field level.

5.2 IMS eWay Considerations

The following error message could appear in the mainframe log, even when the IMS eWay runs successfully:

HWSP1435E Socket closed; Request message incomplete; M=SDRC

This is an erroneous error message that originates from IBM's IMS Connect program. The error message is resolved by the following PTFs from IBM:

- PQ53848 (UQ68076) for IMS Connect 1.1
- PQ62496 (UQ68078) for IMS Connect 1.2.

5.3 IMS eWay Sample Projects

Three sample projects are described in this chapter to demonstrate how eWay components are created and implemented in a production environment. These sample eWay projects are included as part of the installation CD-ROM package.

- **The IMS_JCE_Sample Project** on page 45, demonstrates connectivity with an IMS system.
- **The IMS_MFS_Sample Project** on page 58, demonstrates the IMS MFS Builder.
- **The MFS_IMS_Sample Project** on page 67, demonstrates the use of the MFS OTD to compose IVTNO transaction messages.

5.4 Importing a Sample Project

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

1. The sample files are uploaded with the eWay's documentation .sar file and downloaded from the Enterprise Manager's **Documentation** tab. Extract the samples from the Enterprise Manager to a local file.
2. Save all unsaved work in Enterprise Designer before importing a sample project.
3. From the Enterprise Designer's Project Explorer pane, right-click the Repository and select **Import** from the shortcut menu. The **Import Manager** appears.
4. Browse to the directory that contains the sample project zip file. Select the sample file (for this sample, **IMS_JCE_Sample.zip**) and click **Import**. After the sample project is successfully imported, click **Close**.
5. Before an imported sample project can be run you must do the following:
   - Create an **Environment** (see **Creating an Environment** on page 53)
   - Configure the eWays for your specific system (see **Configure the eWays** on page 54)

- Create a **Deployment Profile** (see Creating and Activating the Deployment Profile on page 56)

5.5 The IMS_JCE_Sample Project

The **IMS_JCE_Sample** project demonstrates connectivity with an external IMS system.

Sample Project Overview

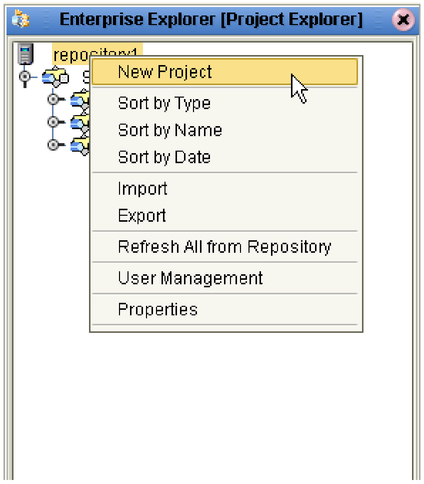
1. The inbound File eWay receives a file from an external directory.
2. This file is the published by the IMS eWay to an external IMS application.
3. The IMS eWay receives the file back from the external IMS application, and publishes the file to the outbound File eWay.
4. The outbound File eWay publishes the file to an external directory.

5.5.1. Create a Project

The first step is to create a new project in the eGate Enterprise Designer.

1. Start the Enterprise Designer.
2. From the Enterprise Explorer's Project Explorer tab, select your Repository on the Project Explorer tree, right-click the Repository and select **New Project** (see Figure 22). A new project (**Project1**) appears on the Project Explorer tree.

Figure 22 Enterprise Explorer - New Project



3. Click twice (not a double-click) on **Project1** and rename the project (for this sample, **IMS_JCE_Sample**).

5.5.2 Create a Connectivity Map

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

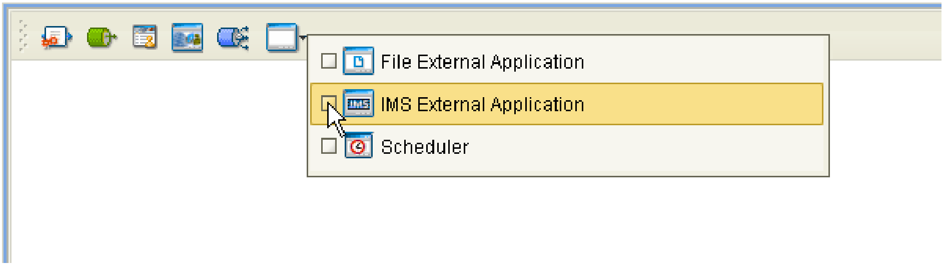
1. In Enterprise Explorer's Project Explorer, right-click the new project and select **New > New Connectivity Map** from the shortcut menu.
2. The New Connectivity Map appears and a node for the Connectivity Map is added to the Project Explorer tree labeled **CMap1**. Rename the Connectivity Map **IMS_JCE_Sample_CM**.

Select the External Applications

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

When creating a Connectivity Map, the eWays are associated with external systems. For example, to establish a connection to IMS, you must first select IMS as an External Application to use in your Connectivity Map (see Figure 23). The IMS External Application icon is then added to the Connectivity Map toolbar.

Figure 23 Connectivity Map - External Applications



To add the External Applications used with the IMS_JCE_Sample project, do the following:

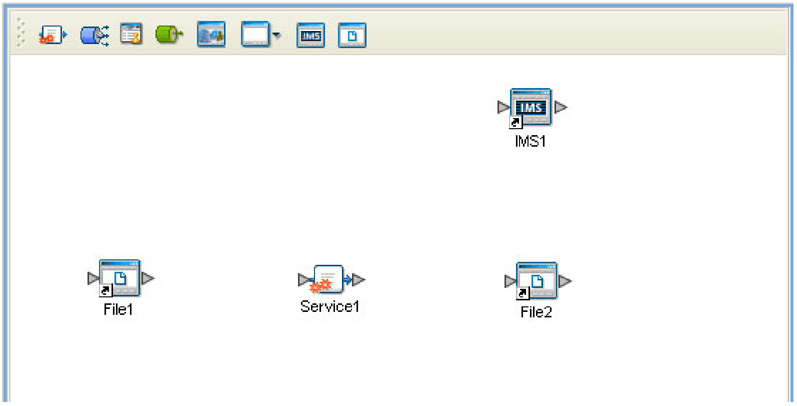
1. Click the **External Application** icon on the Connectivity Map toolbar.
2. Select the external systems that are necessary for your project (for this sample, **IMS** and **File**). Icons representing the selected external systems are added to the Connectivity Map toolbar.

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 the following components onto the Connectivity Map canvas as displayed in **Figure 24 on page 47**:
   - File External System (2)
   - Service (A Service is a container for Java Collaborations, Business Processes, eTL processes, and so forth.)
   - IMS External System

Figure 24 Connectivity Map with Components



2 Rename the objects by right-clicking the object, selecting **Rename** from the shortcut menu, and typing in the new name. Change the names as follows:

- **File1** to **FileIn**
- **Service1** to **IMS_Service**
- **File2** to **FileOut**
- **IMS1** to **IMS**

3 Save your current changes to the Repository.

5.5.3. Creating the Collaboration Definitions

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

The IMS_Collab Collaboration (Java)

The IMS_Collab Collaboration defines transactions from the inbound File eWay to the IMS eWay and from the IMS application to the outbound File eWay.

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

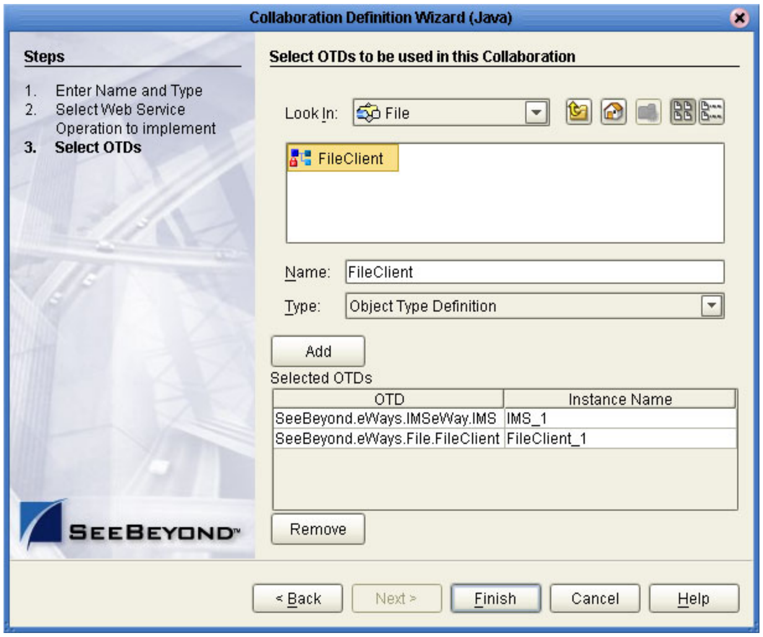
2 Enter a Collaboration Definition name (for this project, **IMS_Java_Collab**) and click **Next**.

3 For Step 2 of the wizard, **Select a Web Services Operation**, Double-click **SeeBeyond** > **eWays** > **File** > **FileClient** > **receive**. The File Name field now displays **receive.** Click **Next**.

4 For Step 3 of the wizard, from the Select OTDs selection window, double-click **SeeBeyond** > **eWays** > **IMSeWay** > **IMS**. The **IMS** OTD is added to the Selected OTDs field.

5 Click the **Up One Level** button to return to the Repository. Double-click **SeeBeyond** > **eWays** > **File** > **FileClient**. The **FileClient** OTD is added to the Selected OTDs field (see Figure 25).

Figure 25 Collaboration Definition Wizard (Java) - Select OTDs



Collaboration Definition Wizard (Java)
Steps
1. Enter Name and Type
2. Select Web Service Operation to implement
3. Select OTDs
SEEBEYOND™
Select OTDs to be used in this Collaboration
Add
Remove
< Back
Next >
Finish
Cancel
Help

Look In: File

FileClient

Name: FileClient

Type: Object Type Definition

Selected OTDs

OTD	Instance Name
SeeBeyond.eWays.IMSeWay.IMS	IMS_1
SeeBeyond.eWays.File.FileClient	FileClient_1

6 Click **Finish**. The Collaboration Editor (Java) appears in the left pane of the Enterprise Designer and the **IMS_Java_Collab** is added to the Project Explorer tree.

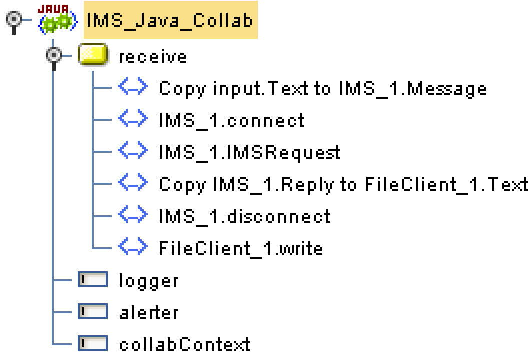
5.5.4. Using the Collaboration Editor (Java)

The next step in the sample is to create the Business Rules of the Collaborations using the Collaboration Editor.

Creating the IMS_Java_Collab Collaboration

Be mindful to open all nodes specified in the directions, so all connections are made to the correct items. The **IMS_Java_Collab** Collaboration contains the Business Rule displayed in Figure 26.

Figure 26 **IMS_Java_Collab** Collaboration Business Rules



IMS_Java_Collab
- receive
  - Copy input.Text to IMS_1.Message
  - IMS_1.connect
  - IMS_1.IMSRequest
  - Copy IMS_1.Reply to FileClient_1.Text
  - IMS_1.disconnect
  - FileClient_1.write
- logger
- alerter
- collabContext

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

1. From the Project Explorer tree, double-click **IMS_Java_Collab** to open the Collaboration Editor to the IMS_Java_Collab Collaboration.

2. To create comments for the Business Rules, from the Business Rules toolbar, click the **comment** icon. The **Enter a Comment** dialog box appears. Enter the comment and click **OK**. The comment is placed on the Business Rules tree under the last selected item. Once the Comment is created, it can be moved by clicking the comment and dragging it up or down the Business Rules tree to a new location.

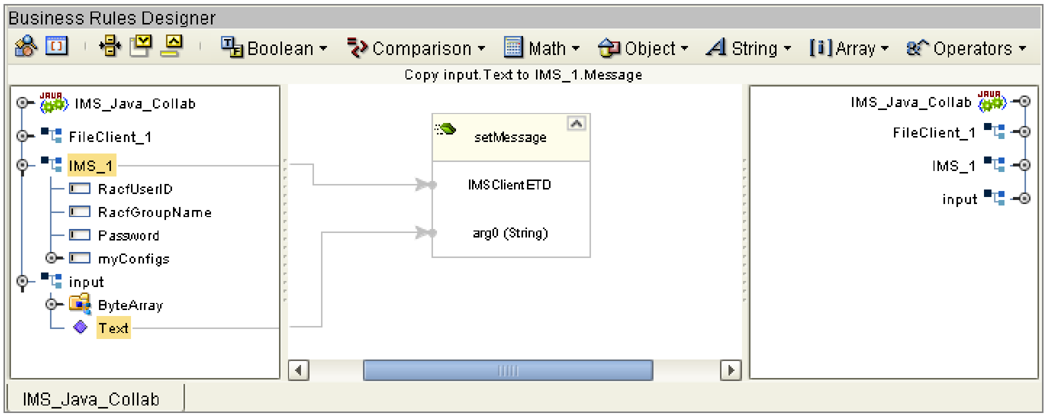
3. To create the **Copy input.Text to IMS_1.Message** rule do the following:

   A. From the left pane of the Business Rules Designer, right-click the **IMS_1** node and select **Select method to call** from the shortcut menu. The method selection box appears.

   B. From the method selection window select **setMessage(String arg0)**. The setMessage method box appears in the Business Rules Designer canvas.

   C. Map **Text**, under **input** in the left pane of the Business Rules Designer, to the **arg0** input node of the setMessage method box. To do this, click on **Text**, under **input** in the left pane of the Business Rules Designer, and drag the cursor to the **arg0** input node of the setMessage method box (see Figure 27).

Figure 27 Copy input.Text to IMS_1.Message Business Rule



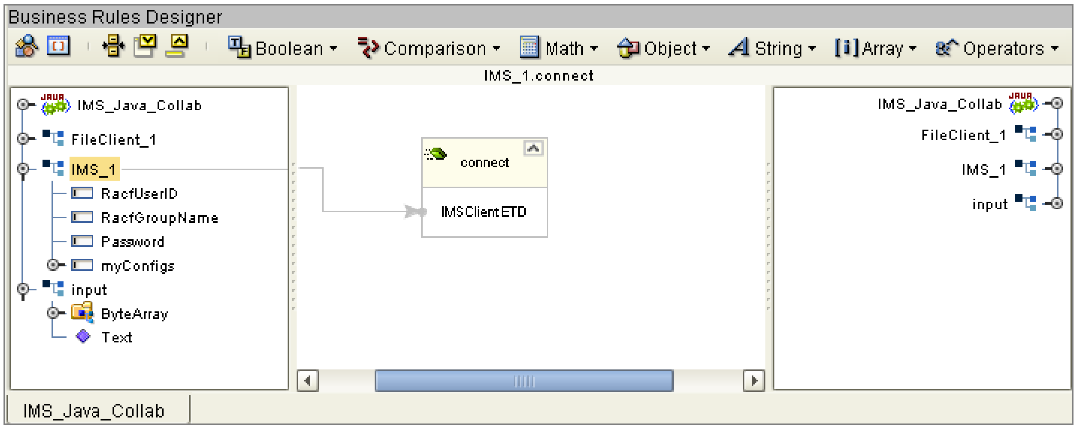
4 To create the **IMS_1.connect** rule do the following:

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

B Right-click the **IMS_1** node in the left pane of the Business Rules Designer and select **Select method to call** from the shortcut menu. The method selection box appears.

C Select **connect()** from the method selection box. The **connect** method box appears in the Business Rules Designer canvas (see Figure 28 on page 50).

Figure 28 IMS_1.connect Business Rule



5 To create the **IMS_1.IMSRequest** Business Rule do the following:

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

B Right-click the **IMS_1** node in the left pane of the Business Rules Designer, and select **Select a method to call** from the shortcut menu. The method selection box appears.

C Select **IMSRequest()** from the method selection box. The **IMSRequest** method box appears in the Business Rules Designer canvas.

6 To create the **Copy IMS_1.Reply to FileClient_1.Text** rule do the following:

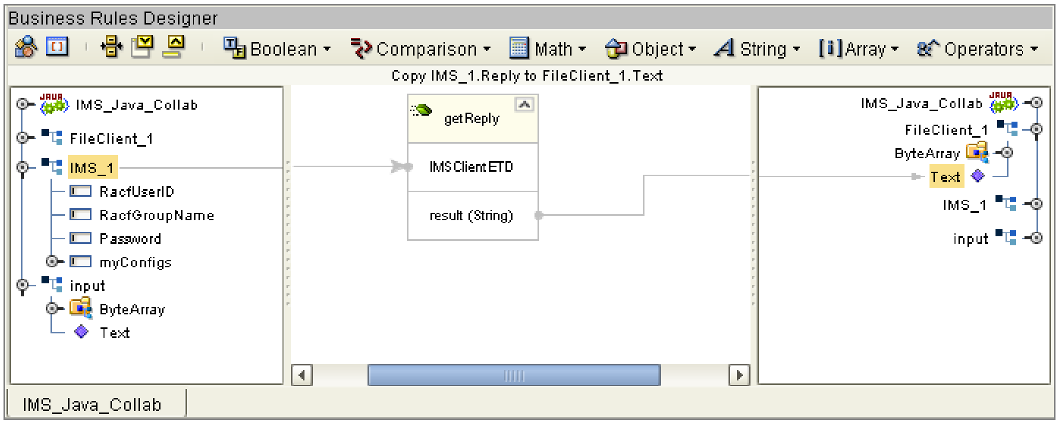
A From the Business Rules toolbar click the **rule** button to add a new rule.

B Right-click the **IMS_1** node in the left pane of the Business Rules Designer, and select **Select a method to call** from the shortcut menu. The method selection box appears.

C Select **IMSReply()** from the method selection box. The **IMSReply** method box appears in the Business Rules Designer canvas.

D Map the **Result (String)** output node of the **IMSReply** method box to **Text**, under **FileClient_1** in the right pane of the Designer (see Figure 29).

Figure 29 Copy IMS_1.Reply to FileClient_1.Text Business Rule



7 To create the **IMS__1.disconnect** rule do the following:

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

B Right-click **IMS_1** in the left pane of the Business Rules Designer, and select **Select a method to call** from the shortcut menu. The method selection box appears.

C Select **disconnect()** from the method selection box. The **disconnect** method box appears in the Business Rules Designer canvas

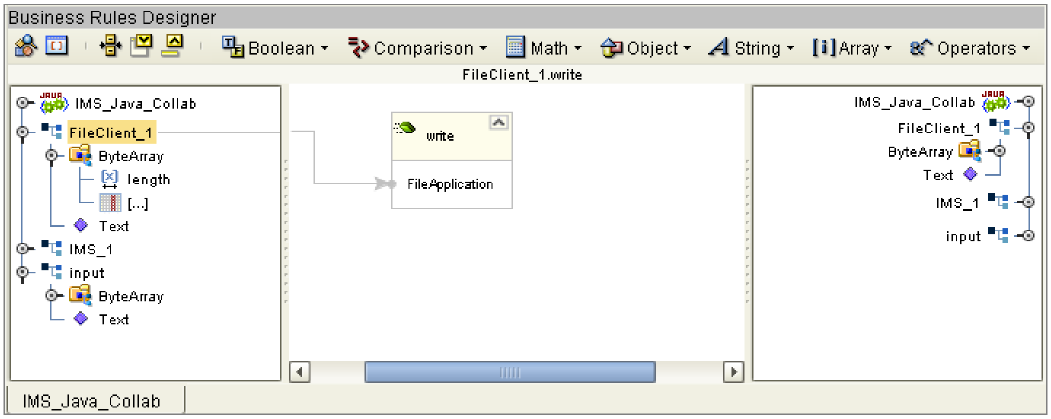
8 To create the **FileClient_1.write** Business Rule do the following:

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

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

C Select **write()** from the method selection menu. The **write** method box appears in the Business Rules Designer canvas (see Figure 30).

Figure 30 FileClient_1.write Business Rule



9 From the Collaboration Editor toolbar, click **Validate** to check the Collaboration for errors.

10 Save your current changes to the Repository.

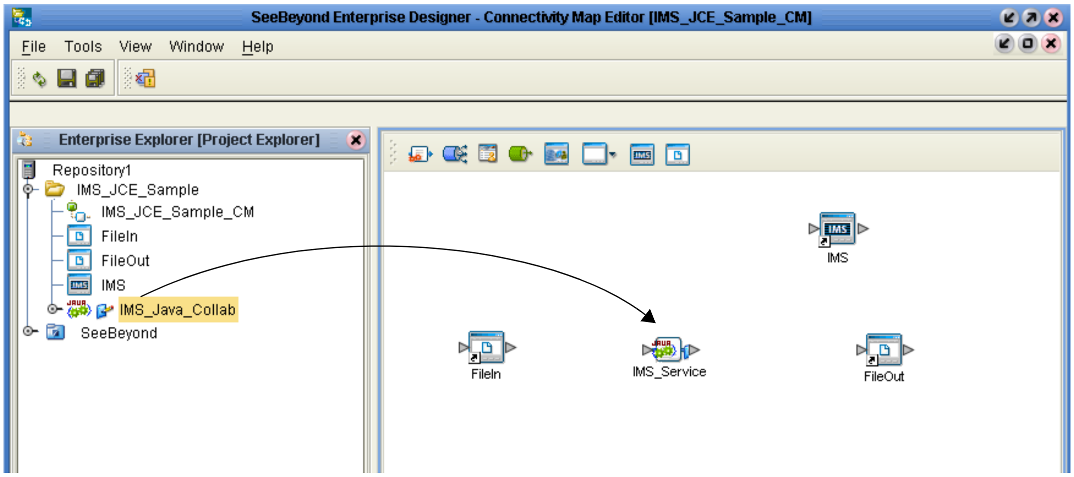
5.5.5. Creating Collaboration Bindings

After the Collaboration has been written, the components are associated and Bindings are created in the Connectivity Map.

1 From the Project Explorer tree, double-click **IMS_JCE_Sample_CM** to display the Connectivity Map.

2 Drag and drop the **IMS_Java_Collab** Collaboration from the Project Explorer tree to the **IMS_Service**. If the Collaboration is successfully associated, the Service icon's "gears" change from red to green (see Figure 31).

Figure 31 Connectivity Map - Binding the Collaborations



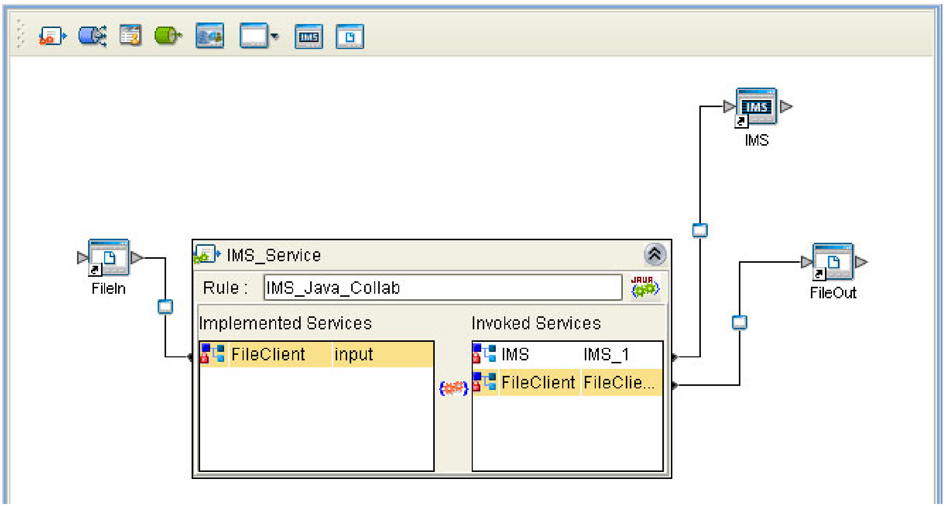
3 Double-click **IMS_Service** in the Connectivity Map. The **IMS_Service** Binding dialog box appears.

4 From the **IMS_Service** Binding dialog box, map **FileClient input** (under Implemented Services) to the **FileIn** External Application, by clicking on **input** and dragging the cursor to the **FileIn** application.

5 From the **IMS_Service** Binding dialog box, map **IMS IMS_1** (under Invoked Services) to the **IMS** External Application.

6 From the **IMS_Service** Binding dialog box, map **FileClient FileClient_1** to the **File2** External Application (see **Figure 32 on page 53**).

Figure 32 Connectivity Map - Connecting the Project's Components



7 Minimize the **IMS_Service** dialog box, and save your current changes.

5.5.6. Creating an Environment

Environments include the external systems, Logical Hosts, integration servers and message servers used by a project, as well as containing the configuration information for these components. Environments are created using the Enterprise Designer's Environment Explorer and Environment Editor.

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

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

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

4 Right-click **IMS_JCE_Sample_Env** and select **New IMS External System**. Name the External System **IMSExtSys**. **IMSExtSys** is added to the Environment Editor.

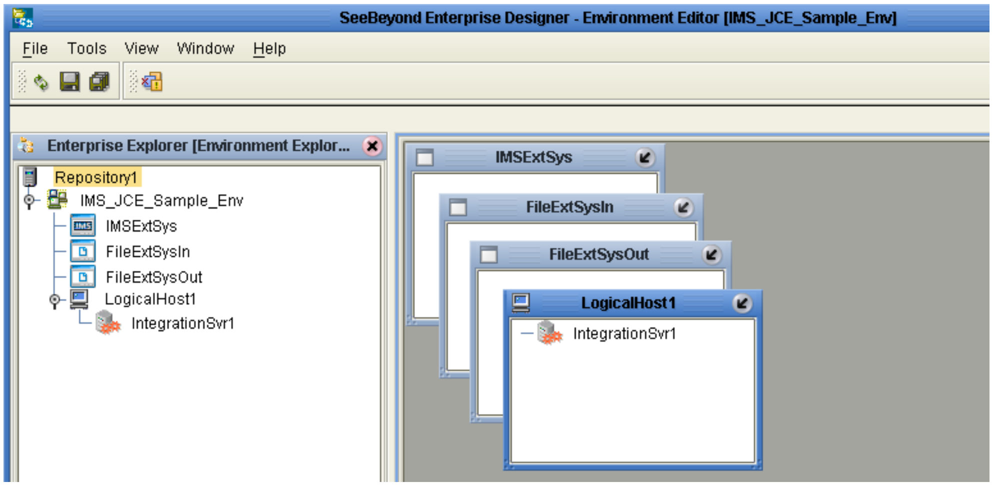
5 Right-click **IMS_JCE_Sample_Env** and select **New File External System**. From the **Create an External System** dialog box, enter **FileExtSysIn** as the name and select **Inbound File eWay** as the type. Click **OK**. **FileExtSysIn** is added to the Environment Editor.

6 Right-click **IMS_JCE_Sample_Env** and select **New File External System** again. Enter **FileExtSysOut** as the name and select **Outbound File eWay** as the type. **FileExtSysOut** is added to the Environment Editor.

7 Right-click **IMS_JCE_Sample_Env** 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 Explorere tree, right-click **LogicalHost1** and select **New SeeBeyond Integration Server** from the shortcut menu. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under LogicalHost1 (see Figure 33 on page 54).

Figure 33 Environment Editor

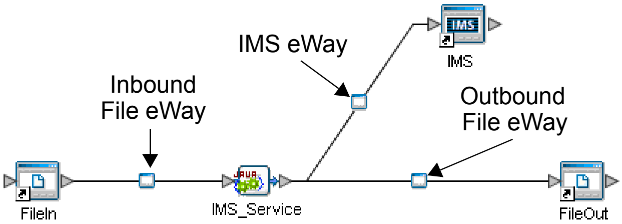


9 Save your current changes to the Repository.

5.5.7. Configure the eWays

The IMS_JCE_Sample project uses three eWays, each represented in the Connectivity Map as a node between an External Application and the Collaboration (see Figure 34). eWays facilitate communication and movement of data between the External Applications and the eGate system.

Figure 34 eWays



Configuring the File eWays

1 Double-click the **inbound File eWay** (see Figure 34). The Templates dialog box appears. Select **Inbound File eWay** in the Templates dialog box and click **OK**.

2 The Properties Sheet opens to the inbound File eWay properties. Modify the properties for your system, including the settings in Table 6, and click **OK**. The properties settings are saved for the eWay.

Table 6 Inbound File eWay Settings

Inbound eWay Configuration Parameters	
Directory	C:/temp
Input file name	Input*.txt

3 Double-click the outbound **File eWay** located between **IMS_Service** and the outbound **File2** External Application, and select **Outbound File eWay**. Modify the outbound **File eWay** configuration for your system, including the settings in Table 7, and click **OK**.

Table 7 Outbound File eWay Settings

Outbound eWay Configuration Parameters	
Directory	C:/temp
Output file name	output%.dat

Configuring the IMS eWay

The IMS eWay properties must be set in both the Project Explorer and Environment Explorer. For more information on the IMS eWay configuration properties and the Properties Sheet, see **Creating and Configuring the IMS eWay** on page 15, or refer to the *eGate Integrator User's Guide*.

Modifying the IMS eWay Connectivity Map Properties

1 From the **IMS_JCE_Sample_CM** Connectivity Map, double-click the **IMS** eWay. The Properties Sheet opens to the IMS eWay Project Explorer configuration properties.

2 Modify the **IMS** eWay (Project Explorer) properties for your system, including the settings in Table 8, and click **OK**.

3 Double-click the **IMS eWay** located between **IMS_Service** and the **IMS1** External Application. Modify the IMS eWay configuration for your system, including the settings in Table 8, and click **OK**.

Table 8 IMS eWay Project Explorer Configuration Settings

IMS eWay (Project Explorer) Configuration Parameters	
Connector Set as directed, otherwise use the default settings	
class	com.stc.eways.ims.IMSClientETDConnector
type	IMSClientETD

Modifying the IMS eWay Environment Explorer Properties

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

2 Modify the IMS eWay environment-configuration for your system, including the settings in Table 9, and click **OK**.

Table 9 IMS eWay Environment Explorer Configuration Settings

IMS eWay (Environment Explorer) Configuration Parameters	
TCPIP Configuration Set as directed, otherwise use the default settings	
Port	*7777*
Server	*Server name*
IRM Header Settings Set as directed, otherwise use the default settings.	
IRM_RACF_GRNAME	*The RACF group name - if RACF is to be used*
IRM_RACF_PW	*The RACF PASSTICKET - if RACF is to be used*
IRM_RACF_USERRID	*The RACF user ID - if RACF is to be used*

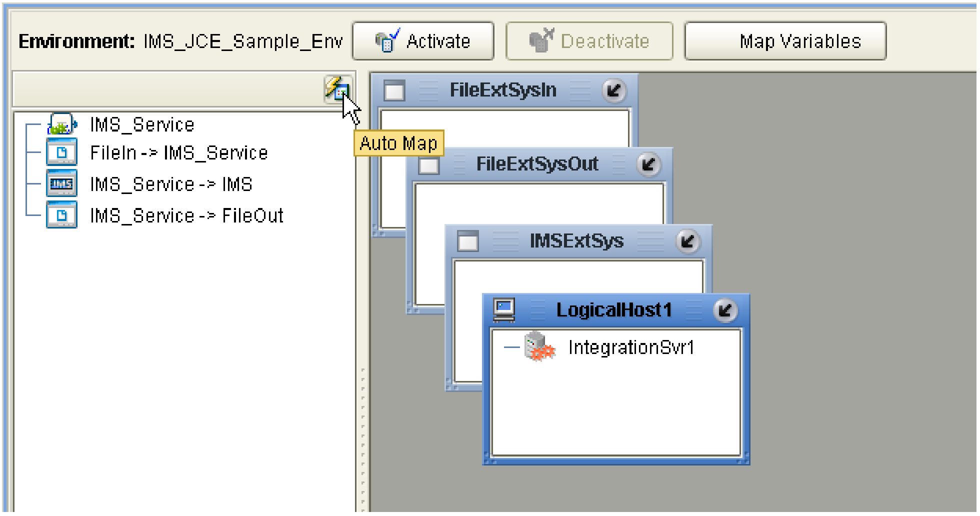
5.5.8 Creating and Activating the Deployment Profile

Deployment Profiles are specific instances of a project in a particular Environment. A Deployment Profile is created using the Enterprise Designer's Deployment Editor.

To create the **IMS_JCE_Sample_DP** Deployment Profile do the following:

1. From the Enterprise Explorer's Project Explorer, right-click the project (**IMS_JCE_Sample**) and select **New** > **Deployment Profile**.
2. From the **Create Deployment Profile for IMS_JCE_Sample** dialog box, enter a name for the Deployment Profile (for this sample **IMS_JCE_Sample_DP**), and select **IMS_JCE_Sample_Env** as the Environment. Click **OK**.
3. Click the **Auto Map** icon as displayed in Figure 35 on page 56. The project's components are automatically mapped to their system window as seen in Figure 36 on page 57. 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 8.

Figure 35 Deployment Profile - Auto Map



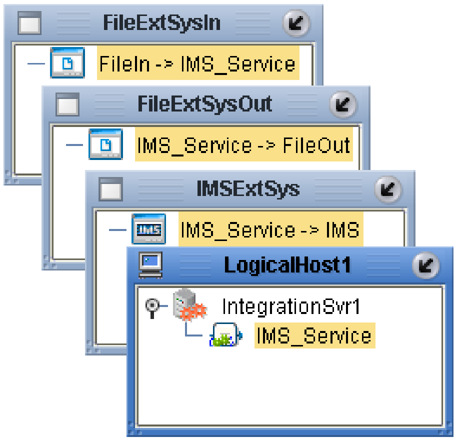
4 From the left pane of the Deployment Editor, drag **IMS_Service -> IMS** to the **IMSExtSys** window.

5 From the left pane of the Deployment Editor, drag **FileIn -> IMS_Service** to the **FileExtSysIn** window.

6 From the left pane of the Deployment Editor, drag **IMS_Service -> FileOut** to the **FileExtSysOut** window.

7 Drag **IMS_Service** to **IntegrationSvr1** in the **LogicalHost1** window (see **Figure 36 on page 57**).

Figure 36 Deployment Profile



8 Click **Activate**, then save the changes to the Repository.

5.5.9. Running the Project

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

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

2 Extract the file to the **C:\ican50\logicalhost2** directory. You must specify the **logicalhost2** directory for it to be created.

3 Navigate to the **C:\ican50\logicalhost2\logicalhost\bootstrap\config** directory and open the logical-host.properties file using Notepad™.

4 Enter and save the following information in the appropriate fields:

- Logical Host root directory: **C:\ican50\logicalhost2\logicalhost**
- Repository URL: **http://localhost:*port number/repository name***
- Repository user name and password: ***Your user name and password***
- Logical Host Environment name: **IMS_JCE_Sample_Env**
- Logical Host name: **logicalhost2**

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

5. Run the **bootstrap.bat** file located in the **C:\ican50\logicalhost2\logicalhost\bootstrap\bin** directory.
6. Copy the sample input data file to the input directory.

5.6 The IMS_MFS_Sample Project

The **IMS_MFS_Sample** project demonstrates the use of the IMS MFS builder to create an OTD from an MFS file and use it in a simple project.

1. Input data, formatted as content to an MFS message output control block, is subscribed to by the inbound File eWay from an external directory.
2. This data (message) is deserialized into an OTD generated from an MFS message specification.
3. The Collaboration demonstrates field-level data access to the message, concatenates and spools specific field values. The output produced by the Collaboration is the concatenation of all 8 message fields in the output control block, producing content identical to the input data.
4. The outbound receives the input data and publishes the message to an external directory.

5.6.1. Create a project

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

1. From the Enterprise Explorer pane of the Enterprise Designer, right-click the Repository and select **New Project**. A new project appears on the Project Explorer tree.
2. Rename the project (for this sample, **IMS_MFS_Sample**).

5.6.2 Create a Connectivity Map

1. In Enterprise Explorer, right-click the new project and select **New > Connectivity Map** from the shortcut menu.
2. The new Connectivity Map appears. From the Project Explorer, rename the new Connectivity Map **IMS_MFS_Sample_CM**.

Select the External Applications

For this sample project, only the File External Application is required.

1. Click the **External Application** icon on the Connectivity Map toolbar.
2. Select the **File External Application** from the selection menu. The File External System is added to the Connectivity Map toolbar.

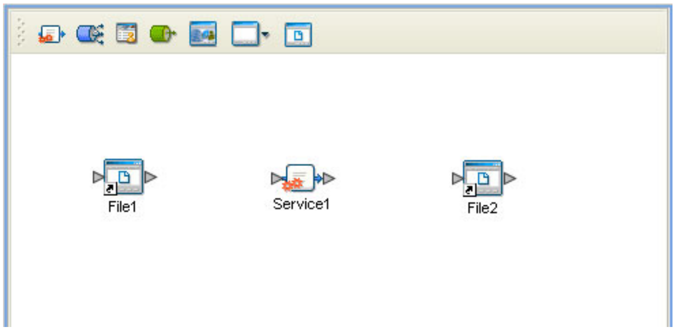
Populate the Connectivity Map

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

1 For this project, add the following components to the Connectivity Map canvas as displayed in **Figure 37 on page 59**:

- File External System (2)
- Service

Figure 37 Connectivity Map with Components



2 Save your current changes to the Repository.

5.6.3. Create an OTD Using the IMS MFS Wizard

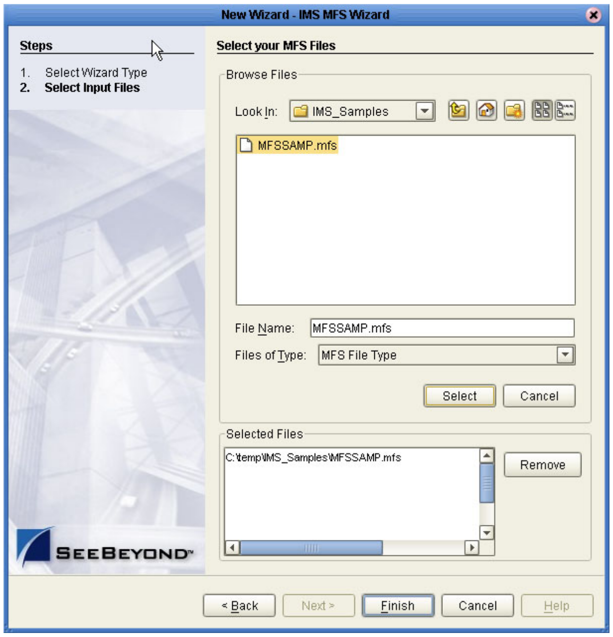
The IMS MFS Wizard creates an OTD from an MFS data file. A sample MFS file, **MFSSAMP.mfs** is provided on the Installation CD-ROM with the IMS eWay sample at ..\samples\ewims.. Copy this file to a temporary directory.

1 From the Project Explorer tree, right-click the **IMS_MFS_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 **IMS MFS Wizard** and click **Next**.

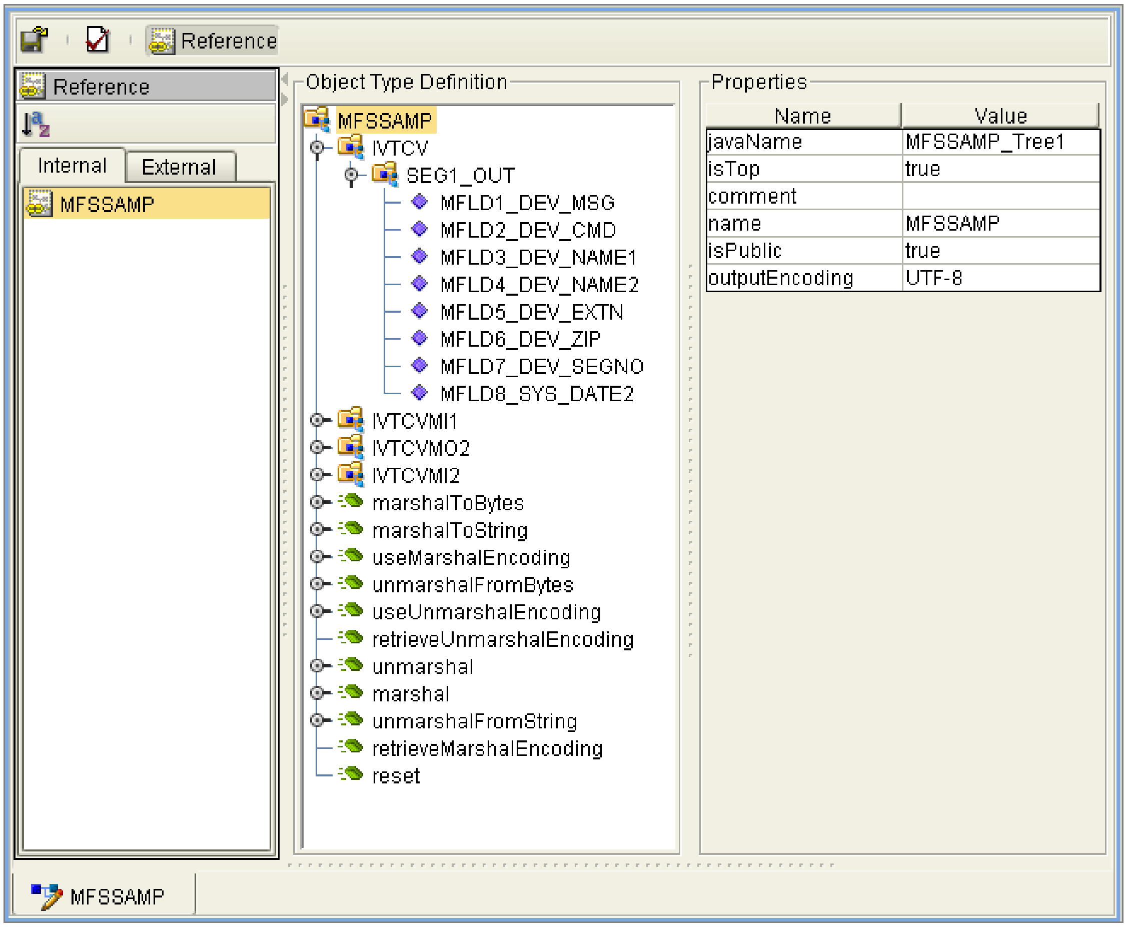
3 For step two of the wizard, Select Input Files, browse to the sample MFS file, **MFSSAMP.mfs**. Click **Select** to add the file to the Selected Files box (see Figure 38).

Figure 38 IMS MFS Wizard - Select MSF file



4 Click **Finish** and save your changes to the Repository. The OTD Editor now displays the successfully created **MFSSAMP** OTD and the OTD is added to the Project Explorer tree (see Figure 39).

Figure 39 OTD Editor - MFSSAMP OTD



Name	Value
javaName	MFSSAMP_Tree1
isTop	true
comment	
name	MFSSAMP
isPublic	true
outputEncoding	UTF-8

5.6.4. Create the Collaboration Definitions

The next step in the sample is to create a Collaboration using the Collaboration Definition Wizard (Java), and Business Rules using the Collaboration Editor.

The JavaCollaborationMFS Collaboration

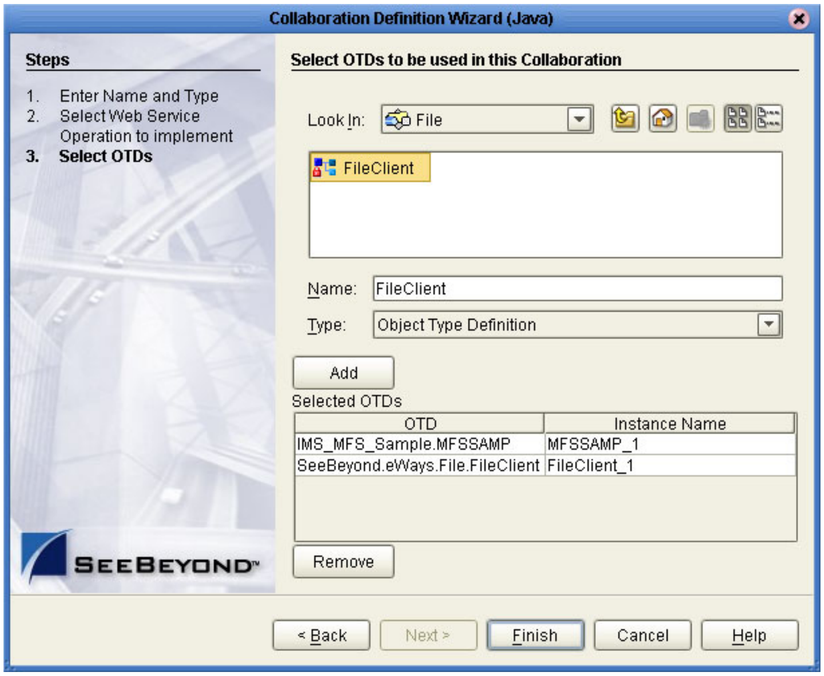
The **JavaCollaborationMFS** Collaboration defines transactions from the inbound File application, converts the MFS data and sends the data to the outbound File application.

1. From the Project Explorer, right-click the current project and select **New** > **Collaboration Definition (Java)** from the shortcut menu. The **Collaboration Definition Wizard** appears.
2. Enter a Collaboration Definition name (for this project, **JavaCollaborationMFS**) and click **Next**.
3. For Step 2 of the wizard, **Select a Web Services Operation**, double-click **SeeBeyond** > **eWays** > **File** > **FileClient** > **receive**. The File Name field now displays **receive.** Click **Next**.

4 For Step 3 of the wizard, from the Select OTDs selection window, double-click **IMS_MFS_Sample** > **MFSSAMP.** The **MFSSAMP** OTD is added to the Selected OTDs field.

5 Click the **Up One Level** button to return to the Repository. From the Select OTDs selection window, double-click **SeeBeyond** > **eWays** > **File > FileClient.** The **FileClient** OTD is added to the Selected OTDs field (see Figure 40).

Figure 40 Collaboration Definition Wizard (Java) - Select OTDs



Collaboration Definition Wizard (Java)

Steps

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

Select OTDs to be used in this Collaboration

Look In: File

Name: FileClient

Type: Object Type Definition

Selected OTDs

OTD	Instance Name
IMS_MFS_Sample.MFSSAMP	MFSSAMP_1
SeeBeyond.eWays.File.FileClient	FileClient_1

6 Click **Finish**. The Collaboration Editor (Java) appears in the left pane of the Enterprise Designer and the **IMS_MFS_Collab** Collaboration (Java) is added to the Project Explorer tree.

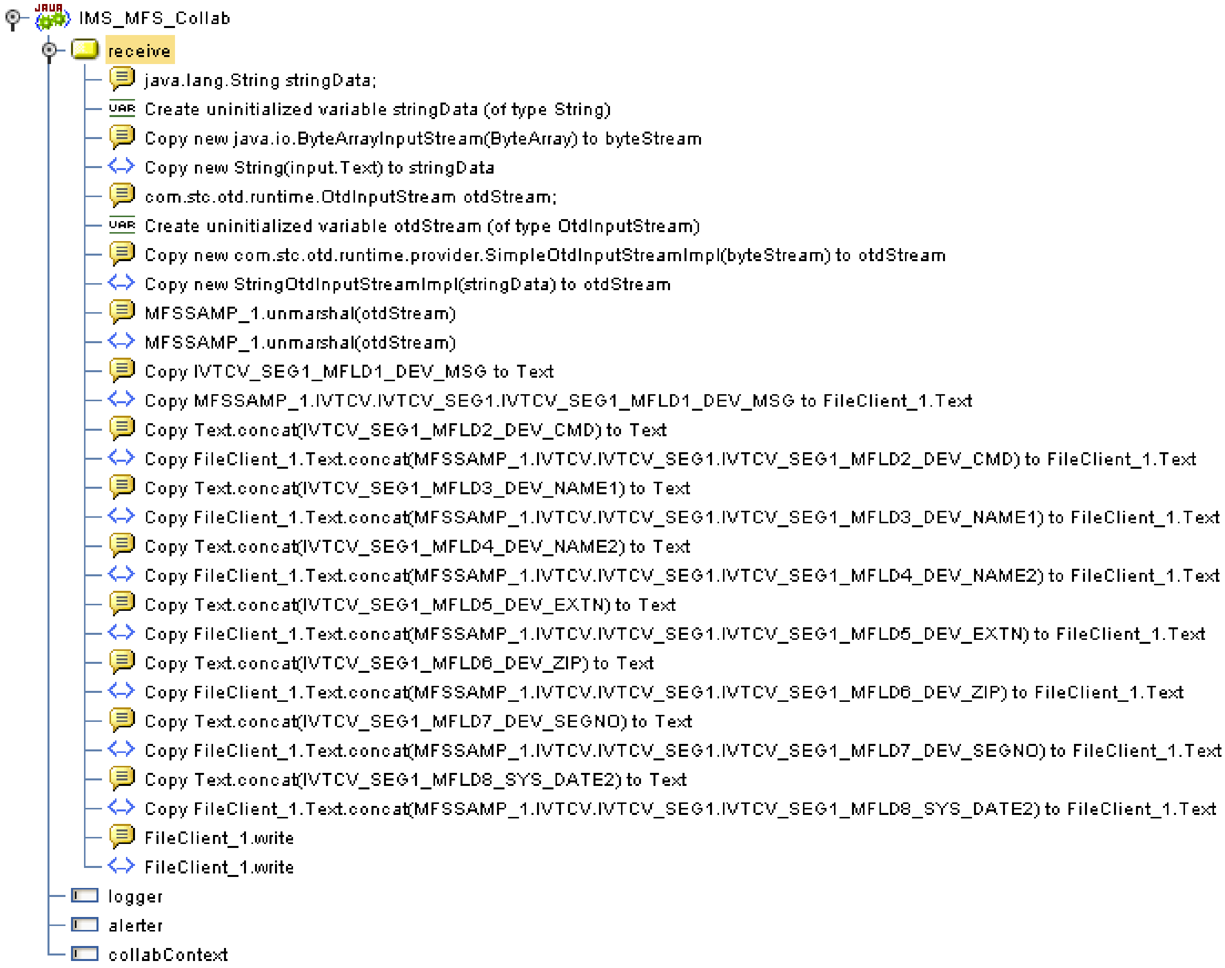
5.6.5. The IMS_MFS_Collab Business Rules

The the **IMS_MFS_Sample** project uses one Collaboration created in the previous section, **IMS_MFS_Collab**.

IMS_MFS_Collab Business Rules

The IMS_MFS_Collab Collaboration contains the Business Rules displayed in **Figure 41 on page 63**. These rules are created using the Collaboration Editor.

Figure 41 IMS_MFS_Collab Collaboration Business Rules



```
IMS_MFS_Collab
  receive
    java.lang.String stringData;
    Create uninitialized variable stringData (of type String)
    Copy new java.io.ByteArrayInputStream(ByteArray) to byteStream
    Copy new String(input.Text) to stringData
    com.stc.otd.runtime.OtdInputStream otdStream;
    Create uninitialized variable otdStream (of type OtdInputStream)
    Copy new com.stc.otd.runtime.provider.SimpleOtdInputStreamImpl(byteStream) to otdStream
    Copy new StringOtdInputStreamImpl(stringData) to otdStream
    MFSSAMP_1.unmarshal(otdStream)
    MFSSAMP_1.unmarshal(otdStream)
    Copy IVTCV_SEG1_MFLD1_DEV_MSG to Text
    Copy MFSSAMP_1.IVTCV.IVTCV_SEG1.IVTCV_SEG1_MFLD1_DEV_MSG to FileClient_1.Text
    Copy Text.concat(IVTCV_SEG1_MFLD2_DEV_CMD) to Text
    Copy FileClient_1.Text.concat(MFSSAMP_1.IVTCV.IVTCV_SEG1.IVTCV_SEG1_MFLD2_DEV_CMD) to FileClient_1.Text
    Copy Text.concat(IVTCV_SEG1_MFLD3_DEV_NAME1) to Text
    Copy FileClient_1.Text.concat(MFSSAMP_1.IVTCV.IVTCV_SEG1.IVTCV_SEG1_MFLD3_DEV_NAME1) to FileClient_1.Text
    Copy Text.concat(IVTCV_SEG1_MFLD4_DEV_NAME2) to Text
    Copy FileClient_1.Text.concat(MFSSAMP_1.IVTCV.IVTCV_SEG1.IVTCV_SEG1_MFLD4_DEV_NAME2) to FileClient_1.Text
    Copy Text.concat(IVTCV_SEG1_MFLD5_DEV_EXTN) to Text
    Copy FileClient_1.Text.concat(MFSSAMP_1.IVTCV.IVTCV_SEG1.IVTCV_SEG1_MFLD5_DEV_EXTN) to FileClient_1.Text
    Copy Text.concat(IVTCV_SEG1_MFLD6_DEV_ZIP) to Text
    Copy FileClient_1.Text.concat(MFSSAMP_1.IVTCV.IVTCV_SEG1.IVTCV_SEG1_MFLD6_DEV_ZIP) to FileClient_1.Text
    Copy Text.concat(IVTCV_SEG1_MFLD7_DEV_SEGNO) to Text
    Copy FileClient_1.Text.concat(MFSSAMP_1.IVTCV.IVTCV_SEG1.IVTCV_SEG1_MFLD7_DEV_SEGNO) to FileClient_1.Text
    Copy Text.concat(IVTCV_SEG1_MFLD8_SYS_DATE2) to Text
    Copy FileClient_1.Text.concat(MFSSAMP_1.IVTCV.IVTCV_SEG1.IVTCV_SEG1_MFLD8_SYS_DATE2) to FileClient_1.Text
    FileClient_1.write
    FileClient_1.write
  logger
  alerter
  collabContext
```

For more information on how to create Business Rules using the Collaboration Editor see **Using the Collaboration Editor (Java)** on page 48 or refer to the *eGate Integrator User's Guide*.

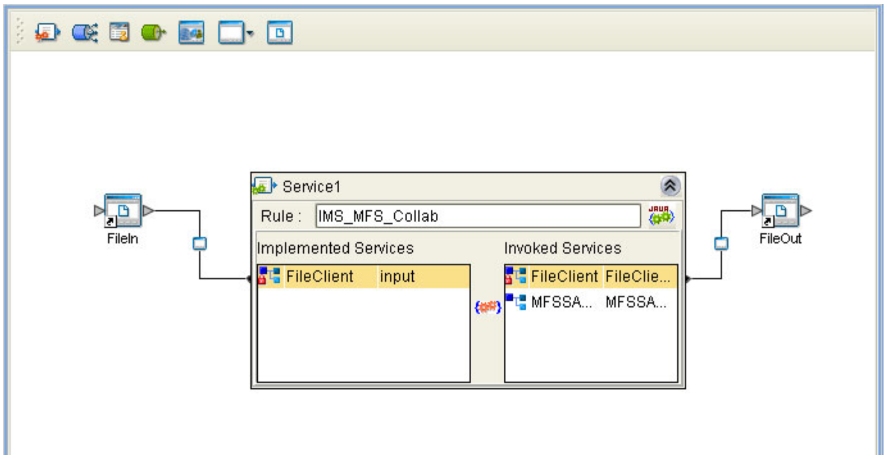
5.6.6. Creating Collaboration Bindings

After the Collaboration has been written, the components are associated and Bindings are created in the Connectivity Map.

1. From the Project Explorer tree, double-click **IMS_MFS_Sample_CM** to display the Connectivity Map.
2. Drag and drop the **IMS_MFS_Collab** Collaboration from the Project Explorer onto the Service (**Service1**).
3. Double-click **Service1** in the Connectivity Map. The **Service1** Binding dialog box appears.
4. From the **Service1** Binding dialog box, drag **FileClient input** (under Implemented Services) to the FileIn External Application.

5 From the **Service1** Binding dialog box, drag **FileClient_1** (under Invoked Services) to the FileOut External Application (see Figure 42 on page 64).

Figure 42 Connectivity Map - Connecting the Project's Components



6 Minimize the **Service1** Binding dialog box and save your current changes to the Repository.

5.6.7. 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.

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

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

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

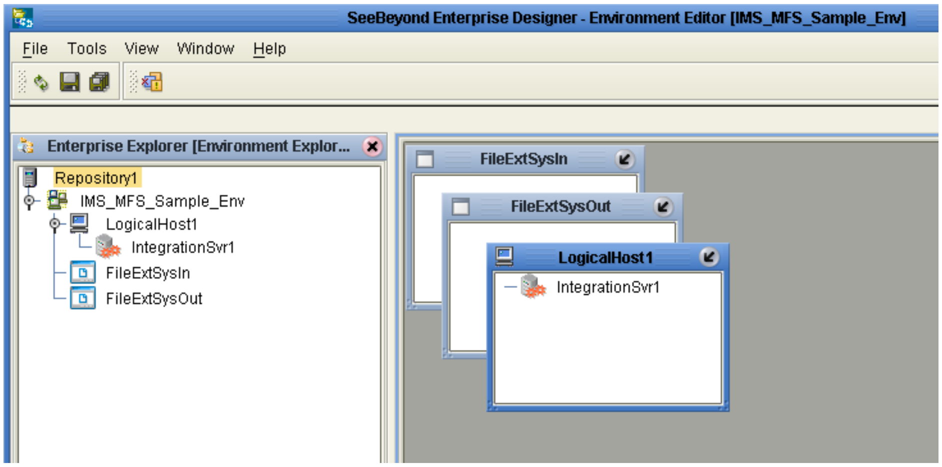
4 Right-click **IMS_MFS_Sample_Env** and select **New File External System**. Name the External System **FileExtSysIn** and select **Inbound File eWay** as the External System Type.

5 Right-click **IMS_MFS_Sample_Env** and select **New File External System**. Name the External System **FileExtSysOut** and select **Outbound File eWay** as the External System Type.

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

7 Right-click **LogicalHost1** and select **New SeeBeyond Integration Server** from the shortcut menu. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under LogicalHost (see Figure 43 on page 65).

Figure 43 Environment Editor


SeeBeyond Enterprise Designer - Environment Editor [IMS_MFS_Sample_Env]
File Tools View Window Help
Enterprise Explorer [Environment Explor...
Repository1
IMS_MFS_Sample_Env
LogicalHost1
IntegrationSvr1
FileExtSysIn
FileExtSysOut
FileExtSysIn
FileExtSysOut
LogicalHost1
IntegrationSvr1

8 Save changes to the Repository.

5.6.8. Configure the eWays

The **IMS_MFS_Sample** project uses two eWays, each represented in the Connectivity Map as a node between an External Application and the Collaboration.

1 Double-click the inbound **File eWay** located between the **File1** External Application and the Collaboration in the Connectivity Map. Select **Inbound File eWay** in the Templates dialog box and click **OK**.

2 The Properties Sheet opens to the **Inbound File eWay** properties. Modify the properties for your system, include the settings for the **Inbound File eWay** in Table 10, and click **OK**.

Table 10 Inbound File eWay Settings

Inbound eWay Connection Parameters	
Directory	C:/temp
Input file name	Input*.txt

3 Double-click the **Outbound File eWay** located between Service and the outbound File External Application, and select **Outbound File eWay**. Modify the File eWay properties for your system, including the settings in Table 11, and click **OK**.

Table 11 Outbound File eWay Settings

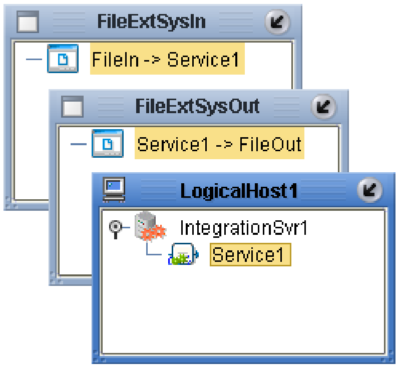
Outbound eWay Connection Parameters	
Directory	C:/temp
Output file name	output%.dat

5.6.9 Creating and Activating the Deployment Profile

A Deployment Profile is used to assign services and message destinations to the integration server and message server.

1. From the Project Explorer tree, right-click the **IMS_MFS_Sample** project and select **New** > **Deployment Profile**.
2. Enter a name for the Deployment Profile (for this project, **IMS_MFS_Sample_DP**). Make sure that the selected Environment is **IMS_MFS_Sample_Env**. Click **OK**.
3. From the Deployment Editor toolbar, click the **Auto Map** icon. The project's components are automatically mapped to their system window as seen in Figure 36. 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 7.
4. From the left pane of the Deployment Editor, drag the **FileIn -> Service1** to the **FileExtSysIn** box.
5. From the left pane of the Deployment Editor, drag the **Service1 -> FileOut** to the **FileExtSysOut** box.
6. Drag **Service1** to **IntegrationSvr1** in the **LogicalHost1** box (see Figure 44).

Figure 44 Deployment Profile



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

5.6.10.Running the Project

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

1. From the Enterprise Manager Downloads tab, double-click **Logical Host - for win32**.
2. Extract the file to the **C:\ican50\logicalhost3** directory. You must specify the **logicalhost3** directory for it to be created.
3. Navigate to the **C:\ican50\logicalhost3\logicalhost\bootstrap\config** directory and open the logical-host.properties file using Notepad™.

4 Enter and save the following information in the appropriate fields:

- Logical Host root directory: **C:\ican50\logicalhost3\logicalhost**
- Repository URL: **http://localhost:***port number/repository name*
- Repository user name and password: ***Your user name and password***
- Logical Host Environment name: **IMS_MFS_Sample_Env**
- Logical Host name: **logicalhost3**

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

5 Run the **bootstrap.bat** file located in the **C:\ican50\logicalhost3\logicalhost\bootstrap\bin** directory.

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

5.7 The MFS_IMS_Sample Project

The **MFS_IMS_Sample** project uses the MFS OTD to compose IVTNO transaction messages.

1 An inbound File eWay supplies last names with which to create IVTNO "DISPLAY" transactions.

2 The MFS OTD is used to compose a transaction message for each supplied name, populating the OTD with the data and the transaction command "DISPLAY"

3 The OTD formats the message, which is passed through the IMS eWay to the IMS application.

4 The response is sent back to the MFS OTD, allowing field-by-field access to its content.

5 An outbound File eWay spools select portions of each transaction response to a flat file.

IMS Application Response Error Messages

The Java Collaboration in the **MFS_IMS_Sample** project includes a rule that causes each IMS application response to be logged as an error message in the Integration Server's log. This it is not an actual error, but is logged as such to display the response content as part of the demonstration without the need for the user to enable debug logging and contend with more extraneous messages.

5.7.1. Importing the MFS_IMS_Sample Project

To import the **MFS_IMS_Sample** project follow the directions from **Importing a Sample Project** on page 44, using the **MFS_IMS_Sample.zip** file.

5.7.2. Creating the Environment

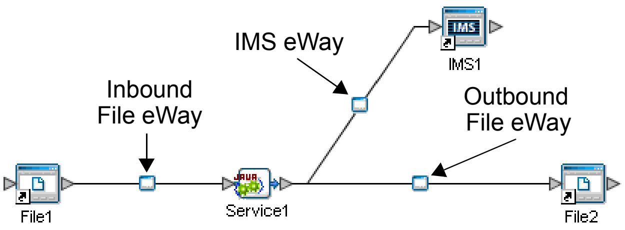
Environments include the external systems, Logical Hosts, integration servers and message servers used by a project, as well as containing the configuration information for these components.

1. From the Enterprise Designer's Enterprise Explorer, click the **Environment Explorer** tab.
2. Right-click the Repository and select **New Environment**. A new Environment is added to the Environment Explorer tree.
3. Rename the new Environment to **MFS_IMS_Sample_Env**.
4. Right-click **MFS_IMS_Sample_Env** and select **New IMS External System**. Name the External System **IMSExtSys**. **IMSExtSys** is added to the Environment Editor.
5. Right-click **MFS_IMS_Sample_Env** and select **New File External System**. Name the External System **FileExtSysIn** and select **Inbound File eWay** as the External System Type.
6. Right-click **MFS_IMS_Sample_Env** and select **New File External System**. Name the External System **FileExtSysOut** and select **Outbound File eWay** as the External System Type.
7. Right-click **MFS_IMS_Sample_Env** and select **New Logical Host**. The **LogicalHost1** box is added to the Environment and **LogicalHost1** is added to the Environment Editor tree.
8. Right-click **LogicalHost1** and select **New SeeBeyond Integration Server** from the shortcut menu. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under LogicalHost.

5.7.3. Configure the eWays

The **MFS_IMS_Sample** project uses three eWays, each represented in the Connectivity Map as a node between an External Application and the Collaboration (see Figure 45). eWays facilitate communication and movement of data between the External Applications and the eGate system.

Figure 45 eWays



Configuring the File eWays

1. Double-click the **inbound File eWay** (see Figure 45). The Templates dialog box appears. Select **Inbound File eWay** in the Templates dialog box and click **OK**.

2 The Properties Sheet opens to the inbound File eWay properties. Modify the properties for your system, including the settings in Table 12, and click **OK**. The properties settings are saved for the eWay.

Table 12 Inbound File eWay Settings

Inbound eWay Configuration Parameters	
Directory	C:/temp
Multiple records per file	True

3 Double-click the **outbound File eWay** located between **Service1** and the outbound **File2** External Application, and select **Outbound File eWay**. Modify the outbound File eWay configuration for your system, including the settings in **Table 13 on page 69**, and click **OK**.

Table 13 Outbound File eWay Settings

Outbound eWay Configuration Parameters	
Directory	C:/temp
Multiple records per file	True
Output file name	output%.dat

Configuring the IMS eWay

The IMS eWay properties must be set in both the Project Explorer and Environment Explorer. For more information on the IMS eWay configuration properties and the Properties Sheet, see **Creating and Configuring the IMS eWay** on page 15, or see the *eGate Integrator User's Guide*.

Modifying the IMS eWay Environment Explorer Properties

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

2 Modify the IMS eWay environment-configuration for your system, including the settings in Table 14, and click **OK**.

Table 14 IMS eWay Environment Explorer Configuration Settings

IMS eWay (Environment Explorer) Configuration Parameters	
TCPIP Configuration Set as directed, otherwise use the default settings	
Port	*Server port number*
Server	*Server name*
IRM Header Settings Set as directed, otherwise use the default settings.	
IRM_TRNCOD	IVTNO
IRM_TRNCOD_SRC	MESSAGE

5.7.4 Creating and Activating the Deployment Profile

Deployment Profiles are specific instances of a project in a particular Environment. A Deployment Profile is created using the Enterprise Designer's Deployment Editor.

To create the **MFS_IMS_Sample_DP** Deployment Profile do the following:

1. From the Enterprise Explorer's Project Explorer, right-click the project (**MSFS_IMS_Sample**) and select **New** > **Deployment Profile**.
2. From the **Create Deployment Profile for MFS_IMS_Sample** dialog box, enter a name for the Deployment Profile (for this sample **MFS_IMS_Sample_DP**), and select **MFS_IMS_Sample_Env** as the Environment. Click **OK**.
3. From the Deployment Editor toolbar, click the **Auto Map** icon. The project's components are automatically mapped to their system window. 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 8.
4. From the left pane of the Deployment Editor, drag **Service1 -> IMS1** to the **IMSExtSys** window.
5. From the left pane of the Deployment Editor, drag **File1 -> Service1** to the **FileExtSysIn** window.
6. From the left pane of the Deployment Editor, drag **Service1 -> File2** to the **FileExtSysOut** window.
7. Drag **Service1** to **IntegrationSvr1** in the **LogicalHost1** window.
8. Click **Activate**. When activation succeeds, save the changes to the Repository.

5.7.5. Running the Project

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

1. From the Enterprise Manager Downloads tab, double-click **Logical Host - for win32**.
2. Extract the file to the **C:\ican50\logicalhost4** directory. You must specify the **logicalhost4** directory for it to be created.
3. Navigate to the **C:\ican50\logicalhost4\logicalhost\bootstrap\config** directory and open the **logical-host.properties** file using Notepad™.
4. Enter and save the following information in the appropriate fields:
   - Logical Host root directory: **C:\ican50\logicalhost4\logicalhost**
   - Repository URL: **http://localhost:*port number/repository name***
   - Repository user name and password: ***Your user name and password***
   - Logical Host Environment name: **MFS_IMS_Sample_Env**
   - Logical Host name: **logicalhost4**

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

5 Run the **bootstrap.bat** file located in the **C:\ican50\logicalhost4\logicalhost\bootstrap\bin** directory.

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

Chapter 6

Java Methods and Classes for the IMS eWay

The IMS eWay contains Java methods that are used to extend the functionality of the eWay.

What’s in This Chapter

6.1 IMS Classes and Methods

The IMS eWay contains the IMSClientETD Class. The exposed methods of the IMSClientETD Class are documented in the IMS Javadoc.

```
java.lang.Object
      |
     + -- com.stc.eways.ims.EwayConnectionETDImpl
            |
           + -- com.stc.eways.ims.IMSClientETD.
```

public class **IMSClientETD**

Extends com.stc.eway.ims.EwayConnectionETDImpl.

Implements com.stc.jcsre.ETDExt.

All implemented interfaces: com.stc.jcsre.ETD, com.stc.jcsre.ETDConstants, com.stc.jcsre.ETDExt.

6.2 IMS Javadoc

The Javadoc is uploaded with the eWay’s documentation file (**IMSeWayDocs.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.

6.3 IMS MFS OTD Methods

The methods available to MFS OTDs created using the IMS MFS OTD Wizard are described in detail on the following pages:

useUnmarshalEncoding(String enc)

Description

Designates the OTD's current unmarshal encoding. The current encoding is used subsequently whenever data is flowed into the OTD using the unmarshal(byte[]) or unmarshal(OtdInputStream) methods. The encoding specified is used throughout the lifetime of the OTD instance or until a subsequent **useUnmarshalEncoding** call declares a different encoding.

If an MFS OTD is used before first using this method to designate a character set encoding, the OTD uses the platform default charset encoding; more precisely, it uses the encoding reported by an internal call to **java.io.InputStreamReader.getEncoding**.

Parameters

Name	Type	Description
enc	java.lang.String	The name of a supported charset. Every implementation of the Java platform is required to support US-ASCII, ISO-8859-1, UTF-8, UTF-16BE, UTF-16LE, UTF-16. Consult the release documentation for your specific implementation for a list of other supported charsets.

Return Values

None.

Throws

None.

useMarshalEncoding(String enc)

Description

Designates the OTD's current marshal encoding. The current marshal encoding is used subsequently whenever data is flowed out from the OTD using the marshal() or marshal(OtdOutputStream) methods. The encoding specified is used throughout the lifetime of the OTD instance or until a subsequent **useMarshalEncoding** call declares a different encoding.

If a MFS OTD is used before first using this method to designate a character set encoding, the OTD uses the platform default charset encoding; more precisely, it uses the encoding reported by an internal call to **java.io.InputStreamReader.getEncoding**.

Parameters

Name	Type	Description
enc	java.lang.String	The name of a supported charset. Every implementation of the Java platform is required to support US-ASCII, ISO-8859-1, UTF-8, UTF-16BE, UTF-16LE, UTF-16. Consult the release documentation for your specific implementation for a list of other supported charsets.

Return Values

None.

Throws

None.

retrieveUnmarshalEncoding()

Description

Reports the currently designated character set encoding used by the OTD to decode loaded data.

Parameters

None.

Return Values

Charset encoding name specified by the most recent prior call to useUnmarshalEncoding, or the platform default charset.

Throws

None.

retrieveMarshalEncoding()

Description

Reports the currently designated character set encoding that the OTD uses to encode output data.

Parameters

None.

Return Values

Charset encoding name specified by the most recent prior call to useMarshalEncoding, or the platform default charset.

Throws

None.

marshalToString()

Description

Produces the OTD content.

Parameters

None.

Return Values

The content of the OTD, as a String.

Throws

com.stc.otd.runtime.MarshalException
Thrown if the OTD content cannot be marshalled because the current OTD marshal encoding is unsupported or cannot encoding the OTD content.

unmarshalFromString(String str)

Description

Loads String data into the OTD.

Parameters

Name	Type	Description
str	java.lang.String	The input data.

Return Values

None.

Throws

com.stc.otd.runtime.UnmarshalException
Thrown if data is not valid for a given OTD, if data is not compatible with the current unmarshal encoding, or if the current unmarshal encoding is unsupported.

marshal(OtdOutputStream out)

Description

Writes the OTD content to the supplied output stream encoded using the current marshal encoding.

Parameters

Name	Type	Description
out	OtdOutputStream	Output stream to which the content is written.

Return Values

None.

Throws

com.stc.otd.runtime.MarshalException
Thrown if the content cannot be marshalled because the content is not encodable using the current OTD marshal encoding, or if the current marshal encoding is unsupported.

marshal()

Description

Produces the OTD content as a sequence of bytes encoded with the OTD's current marshal encoding.

Parameters

None.

Return Values

None.

Throws

com.stc.otd.runtime.MarshalException
Thrown if the content cannot be marshalled because the content is not encodable using the OTD's current marshal encoding, or if the current marshal encoding is unsupported.

unmarshal(OtdInputStream in)

Description

Loads the OTD with data read from the supplied input stream. The input is decoded using the OTD's current unmarshal encoding.

Parameters

Name	Type	Description
in	OtdInputStream	Input stream from which data is loaded into the OTD.

Return Values

None.

Throws

com.stc.otd.runtime.UnmarshalException
Thrown if the data is not valid for a given OTD, if the data is not decodable using the OTD's current unmarshal encoding, or if the unmarshal encoding is unsupported.

unmarshal(byte[] bytes)

Description

Loads the OTD with data read from the byte array. The input is decoded using the OTD's current unmarshal encoding.

Parameters

Name	Type	Description
bytes	byte[]	Array of bytes to use as the source of data.

Return Values

None.

Throws

com.stc.otd.runtime.UnmarshalException
Thrown if the data is not valid for a given OTD, if the data is not decodable using the OTD's current unmarshal encoding, or if the current unmarshal encoding is unsupported.

Index