

***SeeBeyond ICAN Suite***

# **CORBA Client eWay Intelligent Adapter User's Guide**

***Release 5.0***



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

SeeBeyond, e\*Gate, and e\*Way are the registered trademarks of SeeBeyond Technology Corporation in the United States and select foreign countries; the SeeBeyond logo, e\*Insight, and e\*Xchange are trademarks of SeeBeyond Technology Corporation. The absence of a trademark from this list does not constitute a waiver of SeeBeyond Technology Corporation's intellectual property rights concerning that trademark. This document may contain references to other company, brand, and product names. These company, brand, and product names are used herein for identification purposes only and may be the trademarks of their respective owners.

© 2003 by SeeBeyond Technology Corporation. All Rights Reserved. This work is protected as an unpublished work under the copyright laws.

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

Version 20031015144017.

# Contents

---

## Chapter 1

<b>Introducing the CORBA Client eWay</b>	<b>6</b>
Overview	6
Supported Operating Systems	8
System Requirements	8
External System Requirements	8

---

## Chapter 2

<b>Installing the CORBA Client eWay</b>	<b>9</b>
Installing on Windows and UNIX Operating Systems	9
Installing eGate	9
Installing the CORBA Client eWay on an eGate-supported System	10
After Installation	10

---

## Chapter 3

<b>Setting CORBA Client eWay Properties</b>	<b>11</b>
CORBA Client eWay Properties Sheet	11
CORBA Object Properties	13
Interoperable Object Reference (IOR)	13
URI Formats for CORBA Object References	13
Locate Object Using Property	14
Setting Properties	16
Corba Client Settings	16
Locate Object Using	17
Corbaname Settings	18
Corbaname host	18
Corbaname name service	18
Corbaname port	19
Corbaname stringified name	19
Corbaloc Settings	19
Corbaloc host	20
Corbaloc object key	20
Corbaloc port	20

Naming Service IOR Settings	20
Naming IOR file URL	21
Naming IOR ftp password	22
Naming IOR ftp URL	22
Naming IOR ftp user name	22
Naming IOR http URL	22
Object key or stringified name	23
IOR Settings	23
IOR file reference	23
IOR ftp password	24
IOR ftp URL reference	24
IOR ftp user name	24
IOR http URL reference	25
IOR reference	25

---

## Chapter 4

<b>Building a CORBA Client eWay Project</b>	<b>26</b>
Project Canvas	26
Setting Up the eWay	27
eWay Operation	27
eWay Design Overview	28
Using the CORBA Client OTD Wizard	29
OTD Wizard Procedures	29
CORBA Interfaces	34
Overview of CORBA Client eWay Sample Projects	35
Basic eWay Components	36
Building the Project: CORBAHello Sample	36
Sample Description	36
Project Components	37
Project Operation	37
Creating a New Project	37
Creating a Connectivity Map	38
Selecting External Applications	38
Populating the Connectivity Map	38
Creating an OTD	39
Creating the Java Collaboration Definitions	39
Creating Business Rules Within a Collaboration Definition	42
Defining the Collaboration	44
Using the Collaboration Binding Window	45
Creating the Project's Environment	46
Setting eWay Properties	47
Building the Project: CORBAGrid Sample	48
Sample Description	48
Project Components	48
Project Operation	49
Creating Project, Connectivity Map, and External Application	49
Populating the Connectivity Map	50
Creating the Java Collaboration Definitions	51

## Contents

Creating Business Rules Within a Collaboration Definition	52
Defining the Collaboration	56
Using the Collaboration Binding Window	56
Creating the Project's Environment	57
Setting eWay Properties	58
Deploying a Project	59
Basic Steps	59
Alerting and Logging	60
<b>Index</b>	<b>61</b>

# Introducing the CORBA Client eWay

This guide explains how to install, set properties for, and operate the SeeBeyond Technology Corporation's (SeeBeyond) CORBA Client eWay Intelligent Adapter, referred to as the CORBA Client eWay throughout this document.

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

## Chapter Topics

- [“Overview” on page 6](#)
- [“Supported Operating Systems” on page 8](#)
- [“System Requirements” on page 8](#)
- [“External System Requirements” on page 8](#)

---

## 1.1 Overview

The CORBA Client eWay allows non-CORBA systems to communicate with CORBA server applications. CORBA stands for Common Object Request Broker Architecture, a standard for creating, distributing, and managing software objects.

For details on operating and using eGate Integrator and its user interface, see the *eGate Integrator User's Guide*.

### CORBA systems overview

The eWay allows you to integrate non-CORBA systems with CORBA server applications without the need for custom coding. The eWay enables eGate to make requests to a CORBA-compliant object from within a Collaboration.

During operation, the Collaboration can take messages from other sources, for example, the eGate Java Messaging Service (JMS). These messages may contain the properties for calls to different methods on a CORBA server object. The results of these calls are formatted into a single message and published (see [Figure 3 on page 28](#)).

A service is an implementation of one or more interfaces managed by an Object Request Broker (ORB). ORBs generally contain many services consisting of applications, custom components written in-house, and/or services installed with the ORB.

Each service in an ORB is described by an Interface Definition Language (.idl) file. These files are readable text files and usually consist of a module declaration and one or more interfaces.

### eWay general operation

You can select any .idl file (that you provide) containing the interface that represents your desired service. The eGate Enterprise Designer allows you to create a corresponding eGate Object Type Definition (OTD).

A convenient CORBA Client OTD wizard allows you to compile and generate a CORBA client OTD (see [“Using the CORBA Client OTD Wizard” on page 29](#)). Additionally, the OTD creation, via the OTD wizard, uses its own .idl file-to-Java compiler.

This OTD represents a CORBA interface and other interfaces referenced by the interface. One or more of these OTDs are then used in a Collaboration to enable it to make requests to the corresponding CORBA service.

The OTD represents the methods and attributes on the CORBA service. The eWay’s operation in eGate allows it to act as a client to a CORBA server, using the services of a CORBA object from within the Collaboration.

A CORBA Client eWay connects to an eGate Collaboration. The Collaboration can publish or subscribe to single or multiple sources that are part of your implementation. When creating an eGate Connectivity Map in the Enterprise Designer, you must associate the inbound or outbound sources that are part of your implementation with an external application, to establish an eWay.

Some examples of inbound or outbound sources are:

- Inbound or outbound files
- JMS
- Another inbound eWay

### eWay implementation

To create a CORBA Client eWay, you must first select CORBA Client as the outbound external application to use in the current Connectivity Map in the eGate Enterprise Designer. For more information, see the *eGate Tutorial*.

The eWay operates in conjunction with a default client ORB. Therefore, the eWay’s operation and correct implementation require that an ORB run-time environment be installed and properly configured on at least one eGate Logical Host. This process is automatic when you install the eWay.

The properties for the CORBA Client eWay, for convenience, allow you to set necessary parameters of operation using the eWay **Properties** sheet in the eGate Enterprise Designer. See [Chapter 3](#) for details on how to set the CORBA Client eWay properties.

## Supported CORBA data types

The CORBA Client eWay supports the following CORBA data types:

- **short**
- **double**
- **string**
- **wstring**

*Note:* The eWay does not support *.idl* files with a *#pragma* prefix preprocessor directive.

---

## 1.2 Supported Operating Systems

The CORBA Client eWay is available for the following operating systems:

- Windows XP SP1a
- Windows 2000 SP3
- Windows Server 2003
- IBM AIX 5.1 and 5.2
- Sun Solaris 8 and 9

---

## 1.3 System Requirements

To use the CORBA Client eWay, you need:

- eGate Logical Host, version 5.0.
- TCP/IP network connection.

### Logical Host requirements

The eWay must have its properties set and be administered using the Enterprise Designer. For complete information on the Enterprise Designer system requirements, see the *eGate Integrator Installation Guide*.

---

## 1.4 External System Requirements

You need a CORBA server system accessible to eGate. You also need the corresponding *.idl* files.



# Installing the CORBA Client eWay

This chapter explains how to install the CORBA Client eWay.

### Chapter Topics

- [“Installing on Windows and UNIX Operating Systems” on page 9](#)

---

## 2.1 Installing on Windows and UNIX Operating Systems

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

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

### 2.1.1 Installing eGate

The eGate installation process includes the following operations:

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

## 2.1.2. Installing the CORBA Client eWay on an eGate-supported System

The CORBA Client eWay is installed during the installation of eGate Integrator. Follow the instructions in the *eGate Integrator Installation Guide* for installing eGate and include the following steps:

- 1 During the procedures for uploading files to the eGate Repository using the Enterprise Manager, after uploading the **eGate.sar** file, select and upload the following files:
  - ♦ **CorbaClienteWay.sar** (to install the CORBA Client eWay)
  - ♦ **FileeWay.sar** (to install the File eWay, used in the sample Projects)
- 2 Continue installing eGate Integrator as instructed in the *eGate Integrator Installation Guide*.

## 2.1.3. After Installation

Once you have installed and set properties for the eWay, you must then incorporate it into a Project before it can perform its intended functions. See the *eGate Integrator User's Guide* for more information on incorporating the eWay into an eGate Project.

# Setting CORBA Client eWay Properties

This chapter explains how to set properties for the CORBA Client eWay.

## Chapter Topics

- “CORBA Client eWay Properties Sheet” on page 11
- “Setting Properties” on page 16

---

## 3.1 CORBA Client eWay Properties Sheet

When you install the CORBA Client eWay, a default properties template for the eWay is also installed. The template’s default properties are accessible via the eGate Enterprise Designer. These default settings apply to all CORBA Client eWays you use within your current Project.

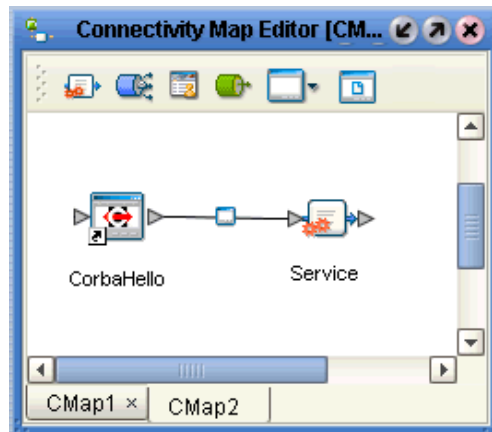
You can set properties for each individual eWay using the Enterprise Designer’s eWay **Properties** sheet. This section describes general procedures on how to change these default properties for the eWay. For details on these steps, see the *eGate Integrator User’s Guide*.

### To set properties for the CORBA Client eWay

- 1 From the eGate Enterprise Designer’s **Project Explorer** create at least one Connectivity Map.
- 2 Create the desired external systems for your one or more Connectivity Maps.

Using the Connectivity Map, you can access the eWay **Properties** sheet for the **Project Explorer** by double-clicking the **eWay** icon. This icon is located on the link between an **External Application** icon and a **Service** icon on the Connectivity Map canvas. See [Figure 1 on page 12](#).

Figure 1 eWay Icon



For the CORBA Client eWay, this **Properties** sheet *contains no editable properties*. However, you must open this dialog box for each **eWay** icon anyway, then click **OK** to close and save it, in order to activate the eWay.

- 3 To access the eWay's editable properties, click the **Environment Explorer** tab (at the bottom of the left pane).
- 4 Create an environment for your Project, then create external systems on the Environment canvas to correspond to the systems you created using the **Project Explorer**.
- 5 Select the external system whose default eWay properties you want to change and right-click, selecting **Properties** from the pop-up menu.

The eWay **Properties** sheet appears. **Figure 2 on page 16** shows the eWay's default properties available from the **Environment Explorer**. You can use this dialog box to modify all of the eWay properties associated with the current external system.

- 6 Click **OK** then **Save All** to save your changes.

#### To use the eWay Properties sheet

- The CORBA Client eWay's properties are set using the eGate Enterprise Designer's eWay **Properties** sheet. The default properties are automatically provided.
- Clicking the **Corba Client Settings** folder in the left pane displays properties group subfolders in the right pane. Click any subfolder to display the eWay's editable properties.
- Many of the entries allow you to enter text. Click the desired text box, then click the ellipsis (...) that appears, to open a dialog box for this purpose.

**Note:** *Even if you do not change the eWay's properties, you must open each **Properties** sheet for every eWay and click **OK** to activate the eWay.*

The rest of this chapter explains all of the eWay's properties in detail under the following sections:

- [“CORBA Object Properties” on page 13](#)
- [“Setting Properties” on page 16](#)

---

## 3.2 CORBA Object Properties

You must tell the eWay where to access the CORBA object by providing a *stringified* Interoperable Object References (IOR) or an Interoperable Naming Service (INS), which provides a Uniform Resource Identifier (URI).

The INS is an extension to the CORBA **CosNaming Service**. An object reference contains at least three pieces of information: an address, the name of the Portable Object Adapter (POA) that created an object reference, and an Object ID. The URI identifies the object.

### 3.2.1 Interoperable Object Reference (IOR)

An IOR is an object reference understood by Object Reference Brokers (ORBs), which can interoperate using the OMG-defined protocols General Inter-ORB Protocol (GIOP) and Internet Inter-ORB Protocol (IIOP). A client can obtain an object reference using **orb.string\_to\_object(objRef)** or as a result of an invocation on another object reference.

The *stringified* IOR is the string representation of an object reference and can be obtained by calling:

```
orb.object_to_string(objRef)
```

### 3.2.2 URI Formats for CORBA Object References

The **corbaloc** and **corbaname** are readable formats that enable you to provide a URI to access CORBA objects.

#### **corbaloc**

The **corbaloc** format is typically used to resolve the reference using the GIOP **LocateRequest** or **Request** message. Use this format to resolve to a particular CORBA Service without going through a Naming Service.

The following **corbaloc** object reference example shows how to get an object reference for **TraderService** from the host **myBank.com** on port 2050:

```
corbaloc:iiop:1.2@MyBank.com:2050/TraderService
```

**Note:** The “1.2” in the sample URI refers to GIOP version 1.2 for the IOR that corresponds to that **corbaloc** URI. GIOP 1.2 is the default value for the default ORB. It is shown in this example in order to demonstrate how you can plug in a different version.

## corbaname

The **corbaname** format provides a mechanism for a client to bootstrap directly and is typically used to resolve the *stringified* name from the root naming context. Use this format to resolve to a *stringified* name from a specific naming context.

The following **corbaname** object reference example shows that **myBank.com** is the host and **2050** is the port:

**corbaname:myBank.com:2050#Personal/schedule**

In the previous example, the portion of the reference up to the hash mark (**corbaname:myBank.com:2050**) is the URI that returns the root naming context. The example provides the URI to use to:

- Locate the Naming Service
- Resolve the name **Personal/schedule** from the Naming Service

### 3.2.3 Locate Object Using Property

Use the **Locate Object Using** properties settings (see [“Locate Object Using” on page 17](#)) to specify whether to use a *stringified* IOR or INS URI for locating and obtaining an object reference for the CORBA interface. Observe the following rules:

- If you use an INS URI, you must specify whether it is in the **corbaname** or **corbaloc** format and enter the information required by the format.
- If you use an IOR, you must enter the IOR in full (**IOR reference** property) or specify a file containing it as a URL (**IOR http URL reference** or **IOR ftp URL reference** property) or via a path (**IOR file reference** property).

You must know how to correctly enter the property precisely according to the type of object reference format (also called bootstrapping mechanism) you use. The reference formats are listed in Table 1. The table also describes the eWay properties for each reference format.

**Table 1** Supported Bootstrapping Mechanisms

Reference Format	Description	Correct Entry
corbaloc	Contacts a published object reference on a server, for example, a Naming Service.	Enter properties as required by the eWay Properties sheet and explained later in this section.
corbaname	Makes a query against a Naming Service. The corbaname: (an extension of corbaloc) contains the stringified name that identifies the right binding in a naming context.	Enter properties as required by the eWay Properties sheet and explained later in this section.

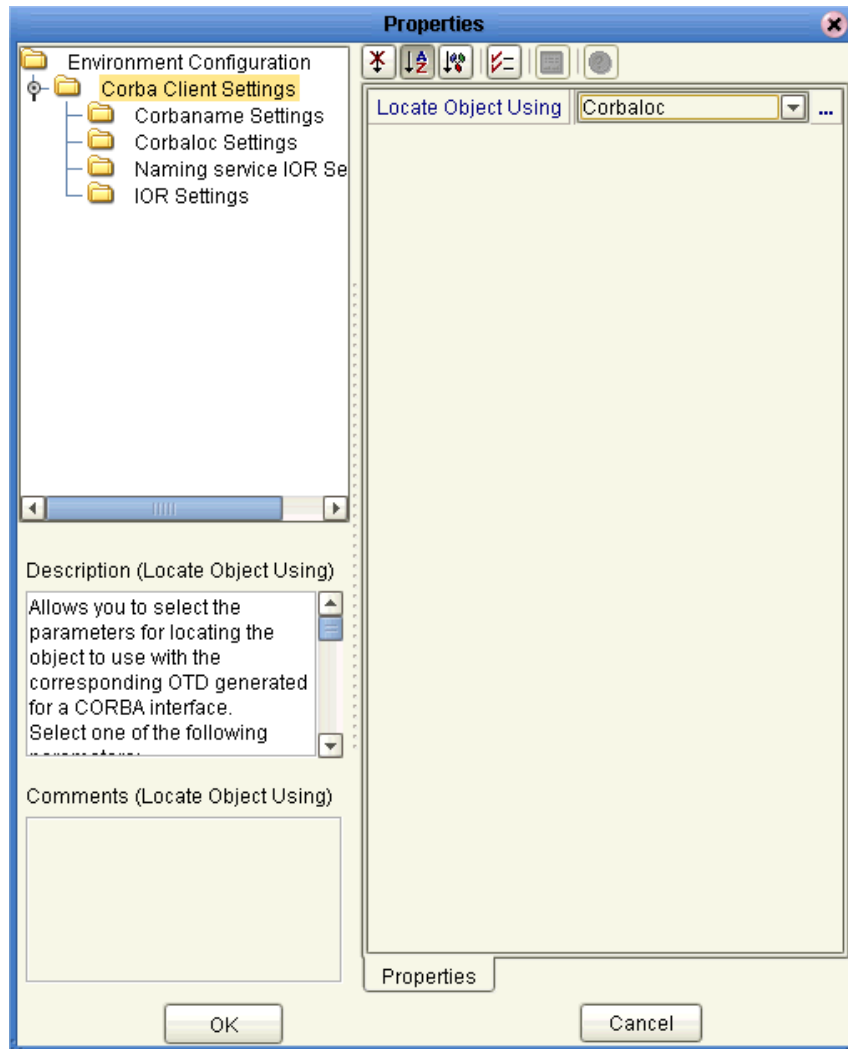
**Table 1** Supported Bootstrapping Mechanisms (Continued)

Reference Format	Description	Correct Entry
Naming Service IOR reference	The CORBA Naming Service can write its own IOR to a URL.	<p>A client-side ORB does not access a URL through a local or shared file system. Instead, the URL is read using the following syntax:</p> <ul style="list-style-type: none"> <li>▪ Name_Path/NS_Ref (file URL)</li> <li>▪ ftp://Server_Name/NS_Ref (FTP URL) (separate properties are provided for the user name and password)</li> <li>▪ http://Server_Name:Port/NS_Ref (HTTP URL)</li> </ul> <p>Where: NS_Ref is the reference to the IOR of a CORBA Naming Service.</p>
IOR file, HTTP URL, and FTP URL reference	<p>These formats are supported by this eWay, allowing you to save an IOR or URI in a file, then specify that file via:</p> <ul style="list-style-type: none"> <li>▪ File and path name, if it is in a reachable system.</li> <li>▪ HTTP URL if it is published on a Web server.</li> <li>▪ FTP URL if it is published in an FTP server.</li> </ul>	<p>Enter the appropriate character string, as follows:</p> <ul style="list-style-type: none"> <li>▪ For a file, enter the complete path and file name.</li> <li>▪ For HTTP, enter the complete URL.</li> <li>▪ For FTP, enter the complete URL (separate properties are provided for the user name and password).</li> </ul>
IOR reference	An unreadable (by a person) unique identifier for an object. Each IOR contains all the information necessary to find and call the desired object.	You must specify the correct IOR identifier, for example, by using copy and paste operations via shortcut keys. Keep in mind that the IOR can change for each run of the server.

## 3.3 Setting Properties

This section explains each of the eWay’s editable properties in detail. You can set these properties using the eGate Enterprise Designer’s eWay **Properties** sheet. See Figure 2.

**Figure 2** eWay Properties Sheet: CORBA Client eWay



**Important:** You must set the property **“Locate Object Using”** on page 17 first before setting any of the others in the eWay **Properties** sheet.

### 3.3.1 Corba Client Settings

This subfolder contains only one property, **Locate Object Using**. You must set this property first. The setting you use here determines which group (subfolder) of settings applies to your current properties.



## Locate Object Using

### Description

Allows you to select the group of properties for locating the object to use with the OTD generated for a CORBA interface.

### Required Values

From the pull-down menu, select one of the following properties:

- **Corbaname:** Queries a Naming Service for the IOR of a registered object. The Naming Service has to support INS specifications.
- **Corbaloc:** Use if your object is published as a remote service.
- **IOR string:** Specifies the full *stringified* IOR; be sure to use the IOR for the exact object you want to call. Be aware that, with some ORBs, the IOR changes each time the server is started.
- **IOR file URL, IOR http URL, or IOR ftp URL:** Specify that the IOR is saved in a file; enter this information using the appropriate format.
- **Naming service IOR file URL, Naming service IOR ftp URL, or Naming service IOR http URL:** Indicate that you are providing the reference to a CORBA Naming Service.

The additional properties you need to set properties for depend upon your choice from the previous list, as noted under each property section. See Table 2 for details.

**Table 2** Choice of eWay Properties

Subfolder Name	Locate Object Using Selection	Applies to These Properties
Locate Object Using	<i>None</i>	<i>None</i>
Corbaname Settings: See <a href="#">“Corbaname Settings” on page 18</a> .	Corbaname	Corbaname host Corbaname service Corbaname port Corbaname stringified name
Corbaloc Settings: <a href="#">“Corbaloc Settings” on page 19</a>	Corbaloc	Corbaloc host Corbaloc object key Corbaloc port
Naming Service IOR Settings: <a href="#">“Naming Service IOR Settings” on page 20</a>	Naming service IOR file URL	Naming IOR file URL
	Naming service IOR ftp URL	Naming IOR ftp password Naming IOR ftp URL Naming IOR ftp user name
	Naming service IOR http URL	Naming IOR http URL
	<i>For all selections: Naming service file, ftp, and http URL</i>	Object key or stringified name

**Table 2** Choice of eWay Properties

Subfolder Name	Locate Object Using Selection	Applies to These Properties
IOR Settings: “IOR Settings” on page 23	IOR file URL	IOR file reference
	IOR ftp URL	IOR ftp password IOR ftp URL reference IOR ftp user name
	IOR http URL	IOR http URL reference
	IOR string	IOR reference

### 3.3.2 Corbaname Settings

Properties in this subfolder allow you to specify **corbaname** settings. Only use these properties if you selected **Corbaname** under “[Locate Object Using](#)” on page 17.

This section explains the following properties:

- “[Corbaname host](#)” on page 18
- “[Corbaname name service](#)” on page 18
- “[Corbaname port](#)” on page 19
- “[Corbaname stringified name](#)” on page 19

#### Corbaname host

##### Description

Specifies the host to use for **corbaname**. This property is used in combination with other fields in this properties to construct:

```
corbaname:corbaname_host:corbaname_port/  
corbaname_naming_service#corbaname_stringified_name
```

##### Required Values

A valid host name.

**Note:** See the appropriate vendor documentation of the server ORB regarding the port and object key for the INS, as well as for the procedures to run/enable the INS.

#### Corbaname name service

##### Description

Specifies the name of the object key for the Naming Service to use for **corbaname**. This property is used in combination with other fields in this properties to construct:

```
corbaname:corbaname_host:corbaname_port/  
corbaname_naming_service#corbaname_stringified_name
```

**Note:** See the appropriate vendor documentation of the server ORB regarding the port and object key for the INS, as well as for the procedures to run/enable the INS.

The object key selected has to be for the INS.

#### Required Values

The name of a valid object key for a CORBA Naming Service, for example, **NameService**.

## Corbaname port

#### Description

Specifies the port to use for corbaname. This property is used in combination with other fields in this properties to construct:

```
corbaname:corbaname_host:corbaname_port/  
corbaname_naming_service#corbaname_stringified_name
```

**Note:** See the appropriate vendor documentation of the server ORB regarding the port and object key for the INS, as well as for the procedures to run/enable the INS.

The port key selected has to be for the INS.

#### Required Values

A valid host port number.

## Corbaname stringified name

#### Description

Specifies the stringified name of the object to use for **corbaname**. This property is used in combination with other fields in this properties to construct:

```
corbaname:corbaname_host:corbaname_port/  
corbaname_naming_service#corbaname_stringified_name
```

The stringified name specified has to be the name the object was bound to in the INS, for example:

**/myApp/MyObject1**

#### Required Values

A valid stringified name of the object to use for **corbaname**.

**Note:** See the appropriate vendor documentation of the server ORB regarding the port and object key for the INS, as well as for the procedures to run/enable the INS. Consult the vendor documentation of the server ORB, as well as the server documentation/implementation regarding the stringified name the server object was bound to.

### 3.3.3 Corbaloc Settings

Properties in this subfolder allow you to specify **corbaloc** settings. Only use these properties if you selected **Corbaloc** under **“Locate Object Using” on page 17**.

This section explains the following properties:

- [“Corbaloc host” on page 20](#)
- [“Corbaloc object key” on page 20](#)
- [“Corbaloc port” on page 20](#)

## Corbaloc host

### Description

Specifies the host to use for **corbaloc**. This property is used in combination with other fields in this properties to construct:

```
corbaloc:corbaloc_host:corbaloc_port/corbaloc_object_key
```

### Required Values

A valid host name.

## Corbaloc object key

### Description

Specifies the object key to use for **corbaloc**. This property is used in combination with other fields in this properties to construct:

```
corbaloc:corbaloc_host:corbaloc_port/corbaloc_object_key
```

### Required Values

The name of a valid **corbaloc** object key.

## Corbaloc port

### Description

Specifies the port to use for **corbaloc**. This property is used in combination with other fields in this properties to construct:

```
corbaloc:corbaloc_host:corbaloc_port/corbaloc_object_key
```

### Required Values

A valid host port number.

## 3.3.4 Naming Service IOR Settings

Properties in this subfolder allow you to specify Naming Service IOR settings. Only use these properties if you specified one of the following IOR references under [“Locate Object Using” on page 17](#):

- Naming service IOR file URL (see [“Naming IOR file URL” on page 21](#))
- Naming service IOR ftp URL (see [“Naming IOR ftp URL” on page 22](#))
- Naming service IOR http URL (see [“Naming IOR http URL” on page 22](#))

In addition, you must enter the string bind name of the current CORBA server (see [“Object key or stringified name” on page 23](#)) in a separate property. For more information on these IOR references, see [Table 1 on page 14](#).

### Accessing CORBA Naming Services

The CORBA Naming Service can write its own IOR to a URL if it is given a reference string. This string points to the desired Naming Service. The URL is read using the following syntax:

- *Name\_Path/NS\_Ref* (file URL)
- *ftp://Server\_Name/NS\_Ref* (FTP URL)
- *http://Server\_Name:Port/NS\_Ref* (HTTP URL)

Where:

- *NS\_Ref* is the reference of a CORBA Naming Service.
- *Name\_Path* is the name and absolute path of a file.
- *Server\_Name* is the name of the HTTP or FTP server.
- *Port* is the HTTP server port number.

Use the appropriate property depending on whether you are using a text file, FTP, or HTTP URL for the IOR's location. Separate properties allow you to enter an FTP user name and password.

This section explains the following properties:

- [“Naming IOR file URL” on page 21](#)
- [“Naming IOR ftp password” on page 22](#)
- [“Naming IOR ftp URL” on page 22](#)
- [“Naming IOR ftp user name” on page 22](#)
- [“Naming IOR http URL” on page 22](#)
- [“Object key or stringified name” on page 23](#)

## Naming IOR file URL

### Description

The URL (complete path location) for a text file containing the desired Naming Service IOR.

### Required Values

A valid URL.

Use this setting only if you chose **Naming Service IOR file URL** under [“Locate Object Using” on page 17](#).

## Naming IOR ftp password

### Description

Allows you to enter the current required FTP log-on password. If the FTP server does not allow for an anonymous log-on, you must enter a password for this property and a user name for the **Naming IOR ftp User Name** property.

### Required Values

A valid FTP password.

Use this setting only if you chose **Naming Service IOR ftp URL** under **“Locate Object Using” on page 17**.

## Naming IOR ftp URL

### Description

The FTP URL location containing the desired Naming Service IOR.

### Required Values

A valid URL.

Use this setting only if you chose **Naming Service IOR ftp URL** under **“Locate Object Using” on page 17**.

## Naming IOR ftp user name

### Description

Allows you to enter the current required FTP log-on user name. If the FTP server does not allow for an anonymous log-on, you must enter a user name for this property and a password for the **Naming IOR ftp Password** property.

### Required Values

A valid FTP user name.

Use this setting only if you chose **Naming Service IOR ftp URL** under **“Locate Object Using” on page 17**.

## Naming IOR http URL

### Description

The HTTP URL location containing the desired Naming Service IOR.

### Required Values

A valid URL.

Use this setting only if you chose **Naming Service IOR http URL** under **“Locate Object Using” on page 17**.

## Object key or stringified name

### Description

The string bind name of the current CORBA server.

Only use this property if you specified one of the following IOR references under [“Locate Object Using” on page 17](#):

- Naming service IOR file URL
- Naming service IOR ftp URL
- Naming service IOR http URL

### Required Values

A valid CORBA server string bind name.

## 3.3.5 IOR Settings

Properties in this subfolder allow you to specify stringified IOR settings. Only use these properties if you specified one of the following IOR references under [“Locate Object Using” on page 17](#):

- IOR file URL (see [“IOR file reference” on page 23](#))
- IOR http URL (see [“IOR ftp URL reference” on page 24](#))
- IOR ftp URL (see [“IOR http URL reference” on page 25](#))
- IOR string (see [“IOR reference” on page 25](#))

For more information on these references, see [Table 1 on page 14](#). Separate properties allow you to enter an FTP user name and password.

This section explains the following properties:

- [“IOR file reference” on page 23](#)
- [“IOR ftp URL reference” on page 24](#)
- [“IOR http URL reference” on page 25](#)
- [“IOR reference” on page 25](#)

## IOR file reference

### Description

Specifies the fully qualified file path containing the stringified IOR. The file must be reachable via the file system of an eGate Logical Host, using the specified file name, for example:

**C:/tmp/myiorfile.txt**

You must ensure that the file is reachable via the specified file name from all Logical Hosts this eWay properties is used from. It is recommended that you use one of the other bootstrapping options, for example, HTTP or FTP instead, if possible.

Use this setting only if you chose **IOR file URL** under “**Locate Object Using**” on [page 17](#).

#### Required Values

A valid path and file name reachable by a Logical Host.

*Note:* This option is not supported by all client ORBs.

## IOR ftp password

#### Description

Allows you to enter the current required FTP log-on password. If the FTP server does not allow for an anonymous log-on, you must enter a password for this property and a user name for the **IOR ftp User Name** property. See “**IOR ftp user name**” on [page 24](#).

#### Required Values

A valid FTP password.

Use this setting only if you chose **IOR ftp URL** under “**Locate Object Using**” on [page 17](#).

## IOR ftp URL reference

#### Description

Specifies the FTP URL where the eWay can find the file containing the stringified IOR, for example:

**ftp://myhost.com:8080/myiorfile.txt**

The FTP server hosting the file must be reachable via FTP from an eGate Logical Host. If the FTP server does not allow for an anonymous log-in, you must enter a user name and password under the **IOR ftp user name** and **IOR ftp password** properties

Use this setting only if you chose **IOR ftp URL** under “**Locate Object Using**” on [page 17](#).

#### Required Values

A valid FTP URL reachable by a Logical Host.

*Note:* This option is not supported by all client ORBs.

## IOR ftp user name

#### Description

Allows you to enter the current required FTP log-on user name. If the FTP server does not allow for an anonymous log-on, you must enter a user name for this property and a password for the **IOR ftp Password** property. See “**IOR ftp password**” on [page 24](#).

#### Required Values

A valid FTP user name.



Use this setting only if you chose **IOR ftp URL** under “**Locate Object Using**” on [page 17](#).

## IOR http URL reference

### Description

Specifies the HTTP URL where the eWay can find the file containing the stringified IOR, for example:

**http://myhost.com:8080/myiorfile.txt**

The Web server hosting the file must be reachable via HTTP from an eGate Logical Host.

Use this setting only if you chose **IOR http URL** under “**Locate Object Using**” on [page 17](#).

### Required Values

A valid HTTP URL reachable by a Logical Host; the default is **http://localhost:180/myiorfile.txt**.

*Note:* This option is not supported by all client ORBs.

## IOR reference

### Description

Specifies an IOR stringified object reference used to locate the object for the CORBA interface, for example:

**IOR:0000000000000001B49444C3A64656D6F2F68656C6C6F2F476F6F644461793A312E300000000000010000000000006000010200000000D31302E312E3139302E32343600004EE0000001C5374616E64617264496D706C4E616D652F001041403C0A1604013D0100000001000000010000001C000000000010001000000010501000100010109000000105010001**

Be sure there are no extra characters such as carriage returns.

Use this setting only if you chose **IOR string** under “**Locate Object Using**” on [page 17](#).

### Required Values

A valid IOR stringified object reference; there is no default.

# Building a CORBA Client eWay Project

This chapter describes how to create an eGate Project. These procedures are the same as those used when implementing a CORBA Client eWay in a production environment.

See the *eGate Tutorial* for an explanation of how to use the eGate Enterprise Designer to create and set properties for the components of an eGate Project.

## Chapter Topics

- [“Project Canvas” on page 26](#)
- [“Setting Up the eWay” on page 27](#)
- [“Using the CORBA Client OTD Wizard” on page 29](#)
- [“Overview of CORBA Client eWay Sample Projects” on page 35](#)
- [“Building the Project: CORBAHello Sample” on page 36](#)
- [“Building the Project: CORBAGrid Sample” on page 48](#)
- [“Deploying a Project” on page 59](#)

---

## 4.1 Project Canvas

Each eGate Project is created using the Enterprise Designer’s Project canvas. The Project canvas contains windows that represent the various stages of your Project. The types of windows in your Project canvas area include:

- **Connectivity Map:** Contains the business logic components, such as Collaborations, topics, queues, and eWays, that you include in the structure of the Project.
- **OTD Editor:** Contains the source files used to create Object Type Definitions (OTDs) to use with a Project.
- **Java Collaboration Definition:** Contains a Java-based Collaboration Definition that you want to add to the Project. This feature makes it possible to develop external Collaboration definitions.

## 4.2 Setting Up the eWay

In general, you set up the CORBA Client eWay as follows:

- **Creating a Connectivity Map:** When you create the Connectivity Map, the CORBA server object appears in a choice list in the Enterprise Designer. You can make your first selection of the object from this list. After your first selection, the Enterprise Designer places the object on a toolbar where it can be dragged onto the canvas of the Connectivity Map, as needed.
- **Selecting an IDL:** Using the Enterprise Designer, you can browse your file system for a file that contains the IDL for the object you want to interface in a Collaboration. You can select a file or select from a choice list any of the standard CORBA service interfaces (for example, Transaction).
- **Creating an OTD:** Using the Enterprise Designer's OTD wizard, you can select which module and interface from the `.idl` file or service you want to create an OTD for. An OTD is then created, which has a node for each function, return object, and property regardless of what IDL type is specified. When it is saved, the OTD and its IDL are stored into the eGate Repository.
- **Creating a Collaboration:** A wizard allows you to create a Java Collaboration using a CORBA Client OTD, as well as other types of OTDs. A Java or XSLT Collaboration Definition Editor allows you to create the Collaboration's business rules. You can create the desired business rules by dragging and dropping values from a source OTD onto the nodes of a destination CORBA Client OTD and other OTDs. These nodes represent CORBA Client functions, which are in turn able to call CORBA Client methods. Another dialog box also allows you to create Collaboration Bindings that control the Collaboration's inputs and outputs.
- **Setting eWay Properties:** Once a data flow is created between the current Collaboration and the external application, an eWay object is created. The properties of this eWay object include the location of the Interoperable Object Reference (IOR), for example, a file or CORBA naming service, for the server object to connect to. You can set properties for these eWay objects using the eWay Properties sheet in the Enterprise Designer.

See the *eGate Integrator User's Guide* for details on how to use these features.

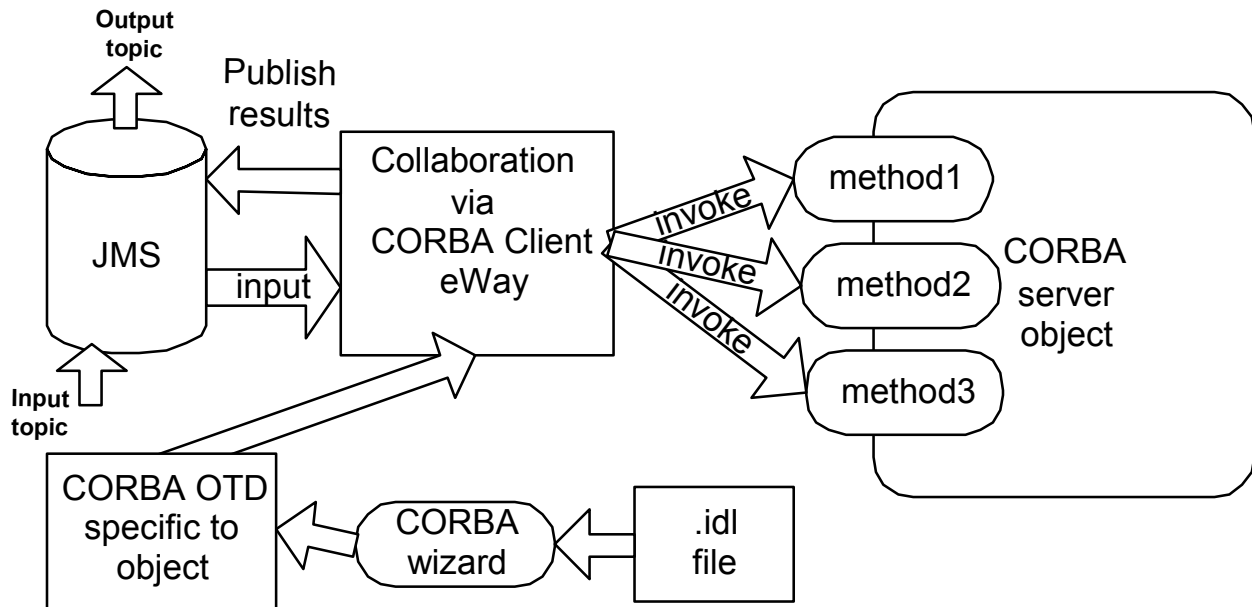
### 4.2.1 eWay Operation

The CORBA client OTD contains Java methods that allow you to use services on the corresponding CORBA server. The eGate Enterprise Designer contains a CORBA wizard feature (see [“Using the CORBA Client OTD Wizard” on page 29](#)) that provides an easy way to create an OTD from your `.idl` file, as described under [“Setting Up the eWay”](#) in the previous list.

## 4.2.2 eWay Design Overview

Figure 3 shows a general diagram of how the CORBA Client eWay operates. In this example, the data input and output is in the form of **.xml** (Extensible Markup Language) files via the eGate Java Messaging Service (JMS). Use of the JMS is optional.

**Figure 3** CORBA Client eWay Operation Example



The following list provides a general operational overview of the eWay:

- 1 Use the CORBA wizard in the eGate Enterprise Designer to convert one or more desired **.idl** files to CORBA OTDs specific to corresponding CORBA objects.
- 2 Input **.xml** files via a JMS Topic component. This data can in turn be input to an eGate Collaboration via the CORBA Client eWay.
- 3 The Collaboration uses eGate DTD (Document Type Definition) OTDs as its Source and Destination components.
- 4 In addition, the CORBA OTD can also be a Destination component of the Collaboration, allowing interface with the CORBA server object.
- 5 Using the CORBA OTD, the Collaboration is able to invoke methods on the CORBA server object.
- 6 Resulting data can be returned to the JMS Topic via the Collaboration and its OTDs.
- 7 You can retrieve data from the JMS Topic, again, as **.xml** data.

---

## 4.3 Using the CORBA Client OTD Wizard

Using the eGate Enterprise Designer's OTD wizard, you select one or more **.idl** files containing the interface that represents the desired service. The wizard guides you through the steps of creating the OTD and handles all necessary operations required for creating the OTD.

The OTD generated by the wizard eliminates the need for manual coding. Using the Collaboration Definition Editor, you can drag and drop OTD nodes to transfer the desired business logic into the appropriate eGate Collaboration Definitions.

*Note:* For complete information on how to use the eGate Enterprise Designer interface and Collaboration Definitions, see the *eGate Integrator User's Guide*.

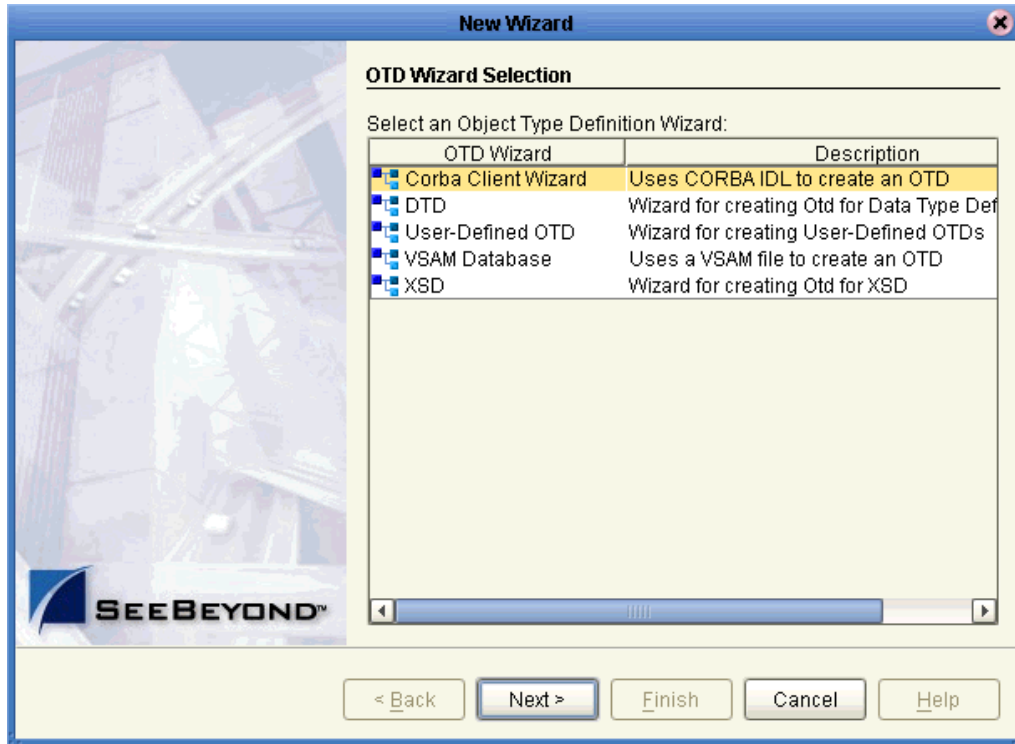
### 4.3.1 OTD Wizard Procedures

This section explains how to use the CORBA Client OTD wizard to create an eGate OTD from an **.idl** file.

To create an OTD from a CORBA **.idl** file

- 1 From the eGate Enterprise Designer's **Enterprise Explorer** pane, right-click the desired Project folder and select **Create an Object Type Definition** from the pop-up menu.
- 2 The **OTD Wizard Selection** window appears, displaying all installed OTD wizards. See [Figure 4 on page 30](#).

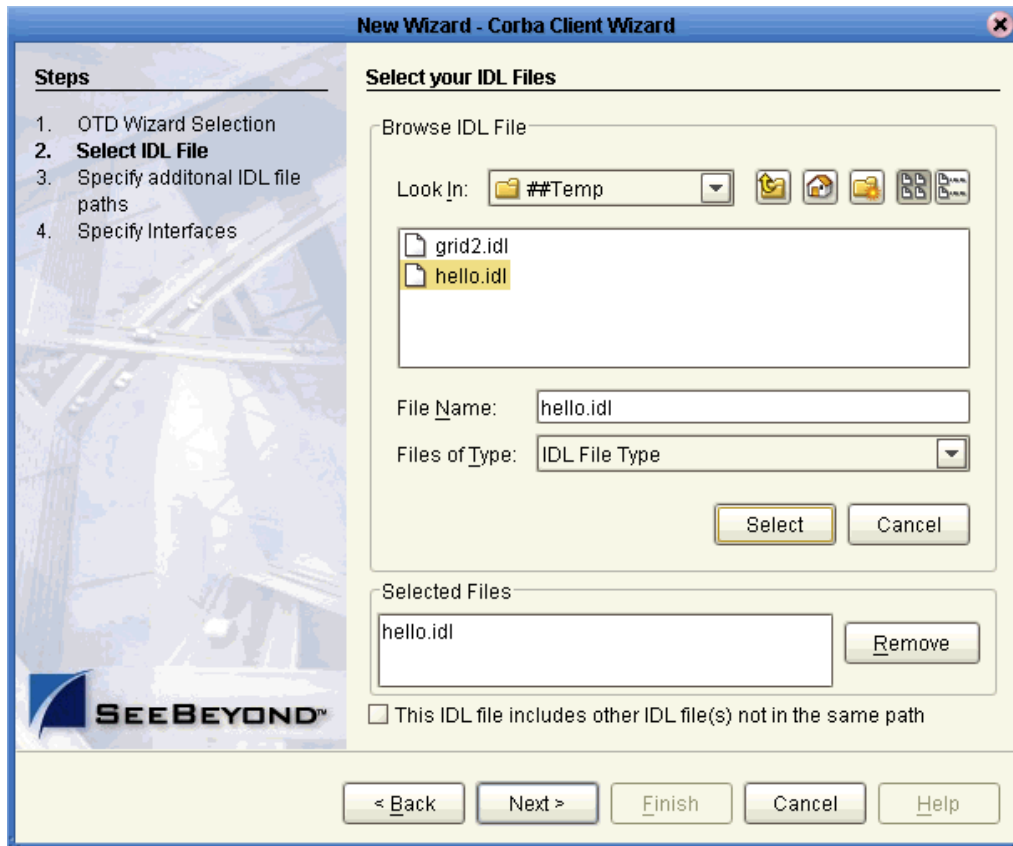
Figure 4 OTD Wizard Selection Window



- 3 Select the **Corba Client Wizard** from the list and click **Next**.

The **Select Your IDL Files** window appears. See [Figure 5 on page 31](#).

Figure 5 Select Your IDL Files Window



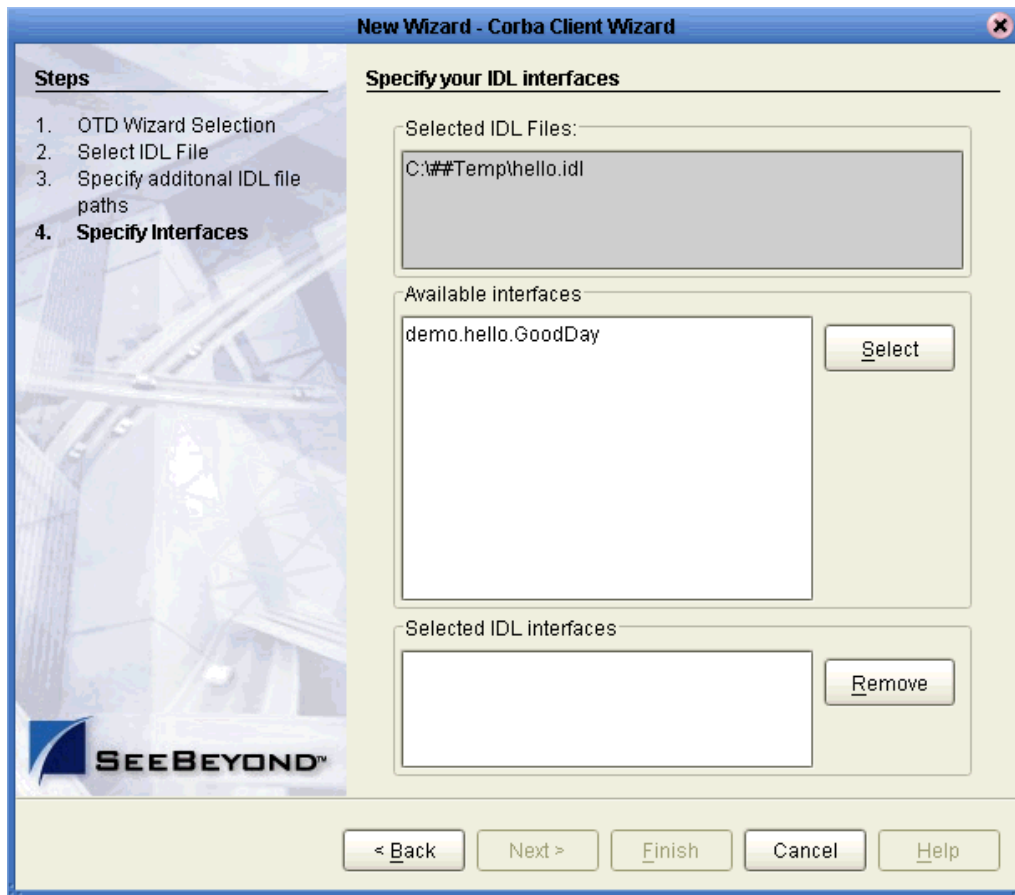
- 4 Use the browse feature to navigate to one or more desired **.idl** files or enter the name of any file. When you select an **.idl** file, its name then appears in the **File Name** text box.

*Note:* If you create an OTD from an **.idl** (Interface Definition Language or IDL) file that references one or more additional **.idl** files, keep in mind that, if any of these **.idl** files is moved or changed, the OTD based on these files may also change.

- 5 If the one or more file names are correct, click **Select**.  
Any selected file name appears in the **Selected Files** text box.
- 6 Click **Next**.

The **Specify Your IDL Interfaces** window appears, as shown in [Figure 6 on page 32](#).

Figure 6 Specify Your IDL Interfaces Window 1

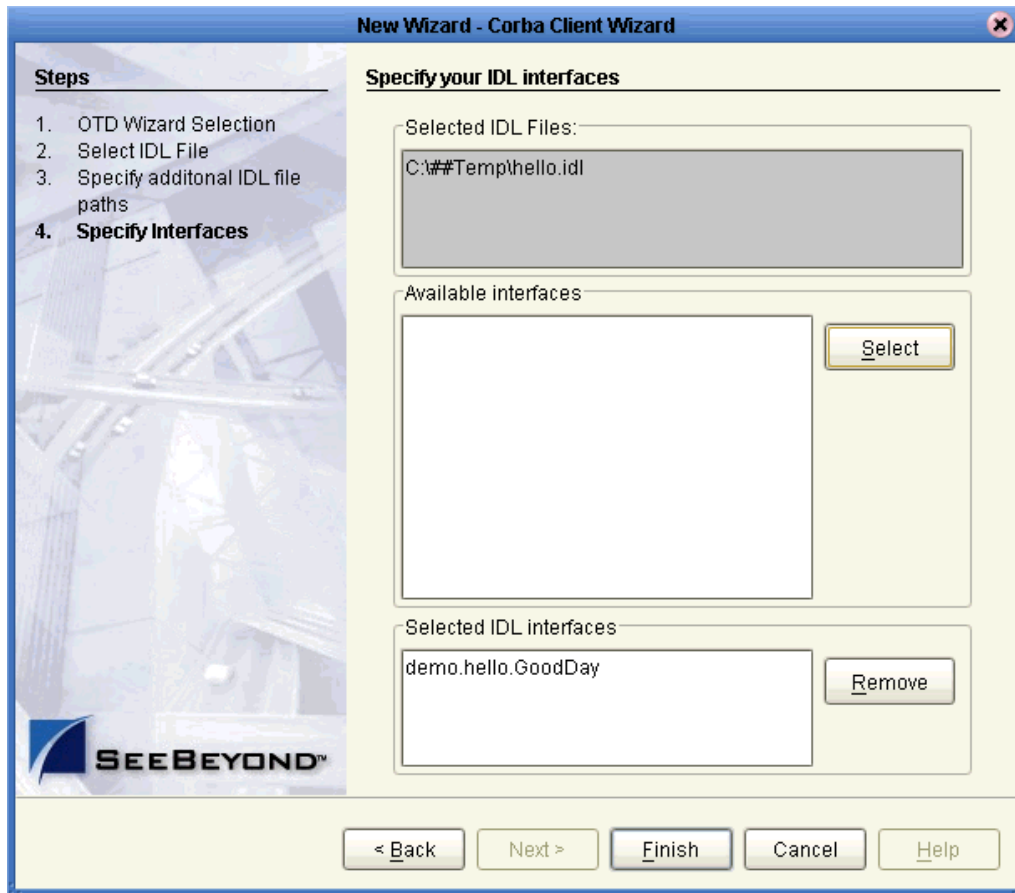


*Note:* For more information on IDL interfaces, see [“CORBA Interfaces” on page 34](#).

- 7 Select the desired interface from any in the list of **Available Interfaces**.  
The IDL interface name then appears in the **Selected IDL Interfaces** text box, as shown in [Figure 7 on page 33](#).



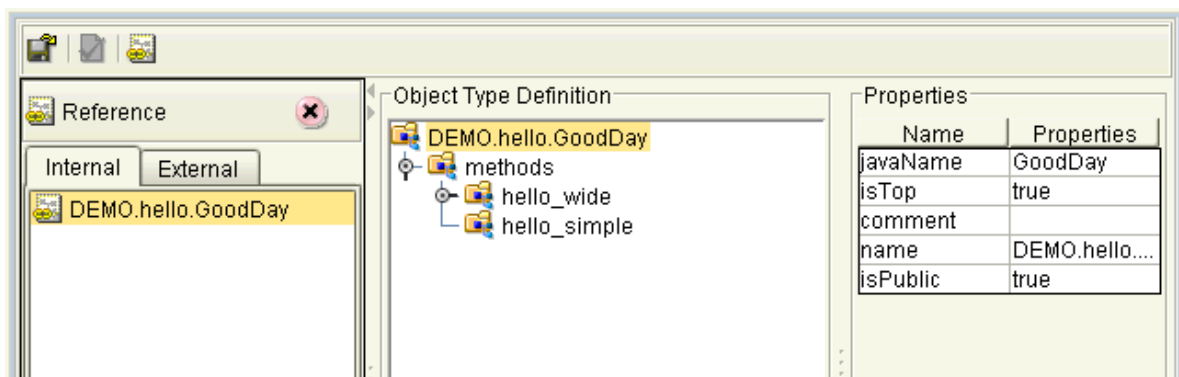
Figure 7 Specify Your IDL Interfaces Window 2



- 8 Review the information you have entered and be sure it is correct. You can click **Back** to change previously entered information, if necessary.
- 9 Click **Finish**.

The structure of the OTD you have created appears, as shown in Figure 8, in the **OTD Editor** window of the Enterprise Designer. The example shows the new CORBA Client OTD with all of its nodes expanded.

Figure 8 Enterprise Designer OTD Editor



**Note:** For more information on the OTD Editor feature and how to use it, see the *eGate Integrator User's Guide*.

## 4.3.2 CORBA Interfaces

This section provides additional details on CORBA interfaces. The following information is helpful in using the CORBA Client OTD wizard to create an OTD:

- A single **.idl** file can contain the description of one or more CORBA interfaces. These interfaces may or may not be contained within a module declaration in the **.idl** file.
- Modules are sometimes used to convey the *scope* of interfaces. For example, there could be a module for **bank** with bank-related interfaces and there could be a **customer** module with customer-related interfaces. In this example, the bank-related interfaces can be considered *scoped* within the **bank** module, and the customer-related interfaces are *scoped* within the **customer** module.
- Additionally, different modules can contain similarly named interfaces. For example, both the **bank** and **customer** modules could contain a **payment** interface. It is important to realize that each of the **payment** interfaces is totally unique. That is, the OTD for one interface cannot be used to access the other.
- The wizard's drop-down list displays the interfaces with *scope* information, which are found in the **.idl** file, if any exists. Using the previous bank and customer example, the interfaces can then be presented as **bank.payment** and **customer.payment**. You must be sure to choose the correct interface.

See the following **.idl** file example:

```
// File banking.idl
module bank {
    interface teller {
        ...
    };
    interface payment {
        ...
    };
}; // end of bank module

module customer {
    interface account {
        ...
    };
    interface payment {
        ...
    };
}; // end of customer module

// end of banking.idl
```

## 4.4 Overview of CORBA Client eWay Sample Projects

This section explains how to implement the CORBA Client eWay using the eGate Project samples included on your installation CD-ROM. These samples allow you to observe end-to-end data-exchange scenarios involving eGate and the CORBA Client eWay.

Both sample Projects are based on a scenario similar to the operations shown in [Figure 3 on page 28](#)

This section explains how to implement this sample Projects, including the CORBA Client eWay. You can also use the procedures given in this chapter to create your own Projects based on the samples provided.

### To import the sample Project

- 1 From the eGate Enterprise Designer, right-click the desired Repository in the **Project Explorer**.
- 2 From the pop-up menu, choose **Import Project**.  
The **Select File to Import** dialog box appears.
- 3 Browse to the directory where you downloaded the sample **.zip** file while you were installing the eWay. For details on how to download this file, see the *eGate Integrator Installation Guide*.

**Note:** *The .zip file you first locate may contain more than one Project and/or additional files. If this is the case, you must unzip this file first, find the desired Project file, then import the Project file.*

You are looking for the container file **CORBA\_Client\_eWay\_Sample.zip**. The file names of the sample files are:

- ♦ **CORBAHello\_Sample\_Project.zip:**  
Name this Project **CORBAHello\_Sample\_Project**.
  - ♦ **CORBAGrid\_Sample\_Project.zip:**  
Name this Project **CORBAGrid\_Sample\_Project**.
- 4 From the **File Destination** dialog box, select **Import to a New Project** and enter a name for the Project, as shown in the previous list.  
The Enterprise Designer imports the selected Project, and its name appears in the **Project Explorer**.
  - 5 Before opening the new Project, click **Save All** then refresh the Enterprise Designer.

**Important:** *An imported Project does not contain an environment or a deployment profile. After importing a Project, you must use the Enterprise Designer to create these functions for the Project. See [“Creating the Project’s Environment” on page 46](#) and [“Deploying a Project” on page 59](#). For additional information, see the *eGate Integrator User’s Guide* and *SeeBeyond ICAN Suite Deployment Guide*.*

## 4.4.1 Basic eWay Components

To use the CORBA Client eWay, you must:

- Set the desired properties settings.
- Create one or more CORBA Client OTDs.

The sections that describe how to build the Project explain more about how to do these operations.

### CORBA Client eWay Properties

The properties for the CORBA Client eWay allow you to enter properties used to connect with a specific external application. You can set these properties using the eWay **Properties** sheet in the eGate Enterprise Designer. For more information about CORBA Client eWay properties and the eWay **Properties** sheet, see [Chapter 3](#).

### CORBA Client OTDs

You create the CORBA Client OTD using the provided **.idl** file and the CORBA Client OTD wizard feature. See [“Using the CORBA Client OTD Wizard” on page 29](#) for details on how to use this wizard.

---

## 4.5 Building the Project: CORBAHello Sample

This section explains the general steps and operations you need to follow to build the sample Project **CORBAHello\_Sample\_Project** for the CORBA Client eWay.

### 4.5.1 Sample Description

This CORBA Client eWay sample Project demonstrates how the eWay processes information from a CORBA system. The resulting information is then written to a text file.

#### Importing the Project

- For an explanation of how to import this sample Project, see the [procedure on page 35](#).
- The Project's **.zip** file name is **CORBAHello\_Sample\_Project.zip**; name this Project **CORBAHello\_Sample\_Project**.

## Project Components

**CORBAHello\_Sample\_Project** has the following components:

- ◆ External file application (inbound): **FileIn**
- ◆ Inbound File eWay
- ◆ Java Collaboration for processing data: **FileIn2CorbaHello2FileOut1**
- ◆ CORBA Client eWay
- ◆ External CORBA application: **CorbaHello**
- ◆ Outbound File eWay
- ◆ External file application (outbound): **FileOut**

## Project Operation

**CORBAHello\_Sample\_Project** does the following operations:

- On instructions from an external text file, takes a file from an external CORBA system (as set in the CORBA Client eWay **Properties** sheet), with the text “My Output string is Hello World, from Somewhere.”
- Outputs the file to an external directory; the file, named **CorbaHellooutput1.dat**, contains the same text.

This Project uses the following **.idl** (**server.idl**) file:

```
module demo
{
    module hello {
        interface GoodDay {
            string hello_simple();
        };
    };
};
```

Be sure to set up and communicate with a CORBA server that is compatible with this **.idl** file.

### 4.5.2 Creating a New Project

To create and name a new Project in the eGate Enterprise Designer

- 1 Start the Enterprise Designer.
- 2 Select the Enterprise Explorer’s **Project Explorer** tab, to show the **Project Explorer** pane (left pane).
- 3 Select the **Repository** icon on the Project Explorer tree.
- 4 Right-click the **Repository** and select **New** then **Project** on the pop-up menus.

A new Project appears on the **Project Explorer** tree, named **Project1**.

- 5 Double-click the Project name and rename the Project, for this sample, **CORBAHello\_Sample\_Project**.

### 4.5.3 Creating a Connectivity Map

A Connectivity Map provides a canvas for assembling and setting properties for a Project's components.

To create a Connectivity Map

- 1 In the eGate Enterprise Designer's **Project Explorer** pane, right-click the new Project's name and select **New** then **Connectivity Map** on the pop-up menus.
- 2 The new Connectivity Map appears and adds a node for the Connectivity Map under the Project on the Project Explorer tree labeled **CMap1**.

The icons in the toolbar represent the components used to populate the Connectivity Map workspace. When linked together, these components define the Connectivity Map for your Project.

### 4.5.4 Selecting External Applications

When creating a Connectivity Map, you must associate the inbound and outbound Collaborations with an External Application. For example, to establish a connection to CORBA Client, you must first select CORBA Client as the external application to use in your Connectivity Map.

To select external applications

- 1 Click the **External Application** icon on the Connectivity Map toolbar.
- 2 Select the external applications necessary for your Project. For this sample, select the File and CORBA Client external applications. Icons representing the File and CORBA Client external applications are then added to the Connectivity Map toolbar.

### 4.5.5 Populating the Connectivity Map

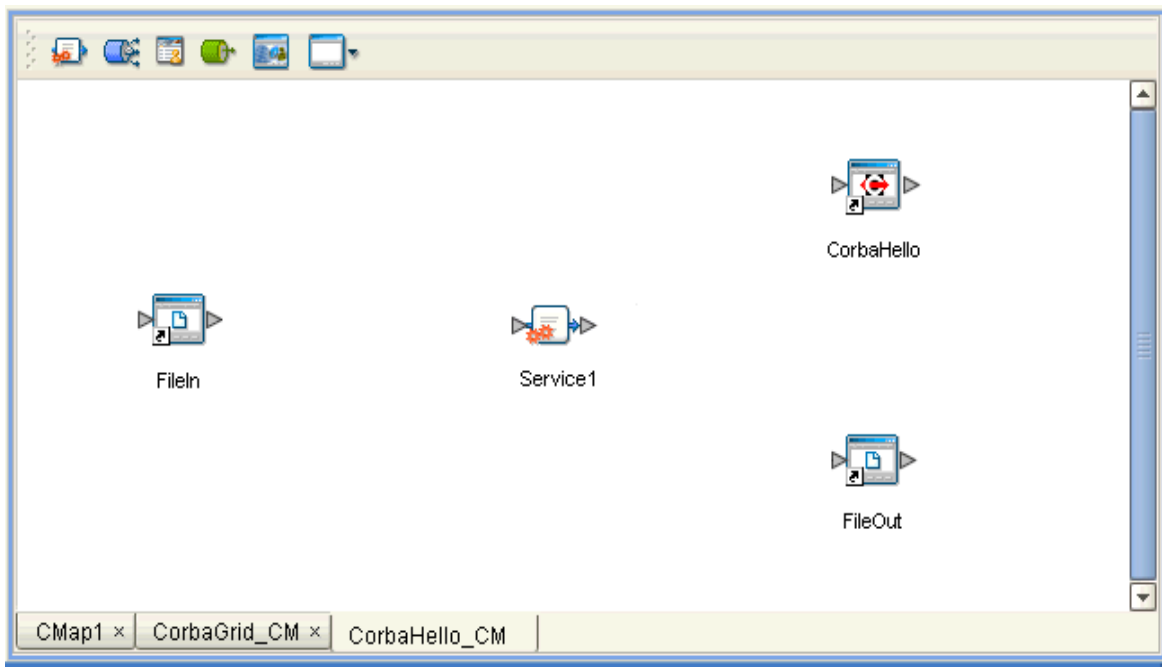
Add the Project components to the Connectivity Map by dragging the icons from the toolbar to the canvas. This operation creates the components for you.

For this sample Project, drag and drop the following components onto the Connectivity Map canvas.

- Two File eWays/external applications
- One Service
- CORBA Client eWay/external application

**Figure 9 on page 39** shows the components on the Connectivity Map.

**Figure 9** Connectivity Map With Components: CORBAHello\_Sample\_Project



Rename the **Service1** component to **FileIn2CorbaHello2FileOut1**. Name the other components as shown in the previous figure. Be sure to save the new Connectivity Map before you proceed. You can click **Save** on the Enterprise Designer toolbar for this purpose.

#### 4.5.6 Creating an OTD

An OTD contains a set of rules that define an object, which encodes data as it travels through eGate. OTDs are used as the basis for creating Collaboration Definitions for a Project. The CORBA Client eWay allows you to create a CORBA Client OTD, using the eGate Enterprise Designer’s CORBA Client OTD wizard.

Using the **server.idl** file for this Project, create a CORBA Client OTD using the OTD wizard. See [“Using the CORBA Client OTD Wizard” on page 29](#) for complete information on how to use this wizard.

Be sure to name the new CORBA Client OTD **CORBACLIENTdemohelloGoodDay**.

#### 4.5.7 Creating the Java Collaboration Definitions

The next step in the sample Project is to create the Java Collaboration Definitions. These components contain rules that define the processing and transport of data as it travels between eGate components.

You create Java Collaboration Definitions to assign to the Collaborations in your Project. This operation enables eGate to recognize the source and destination data relationships in your Project.

You must use the Java Collaboration Definition wizard to create a Java Collaboration Definition. After a Collaboration Definition is created, you can use the Java Collaboration Editor to create the Business Rules logic of the Collaboration.

*Note:* See the *eGate Integrator User's Guide* for complete information on editing Collaborations.

#### To create the Java Collaboration Definition

- 1 From the **Project Explorer** pane, right-click the sample Project and select **New** from the shortcut menu, and **Java Collaboration Definition** from the submenu.

The **Java Collaboration Definition** wizard appears.

- 2 Enter the Collaboration Definition name, **FileIn2CorbaHello2FileOut**, and click **Next**.
- 3 For the wizard's **Step 2**, you must select a Web Services Interface. From the main selection pane, click **Existing Web Service** then double-click in succession the icons for **SeeBeyond > eWays > File > FileClient > receive**.

The **Name** text box now displays **receive**.

- 4 Click **Next**.
- 5 For the wizard's **Step 3**, to select the OTD, double-click in succession the icons for **SeeBeyond > eWays > File > FileClient**.

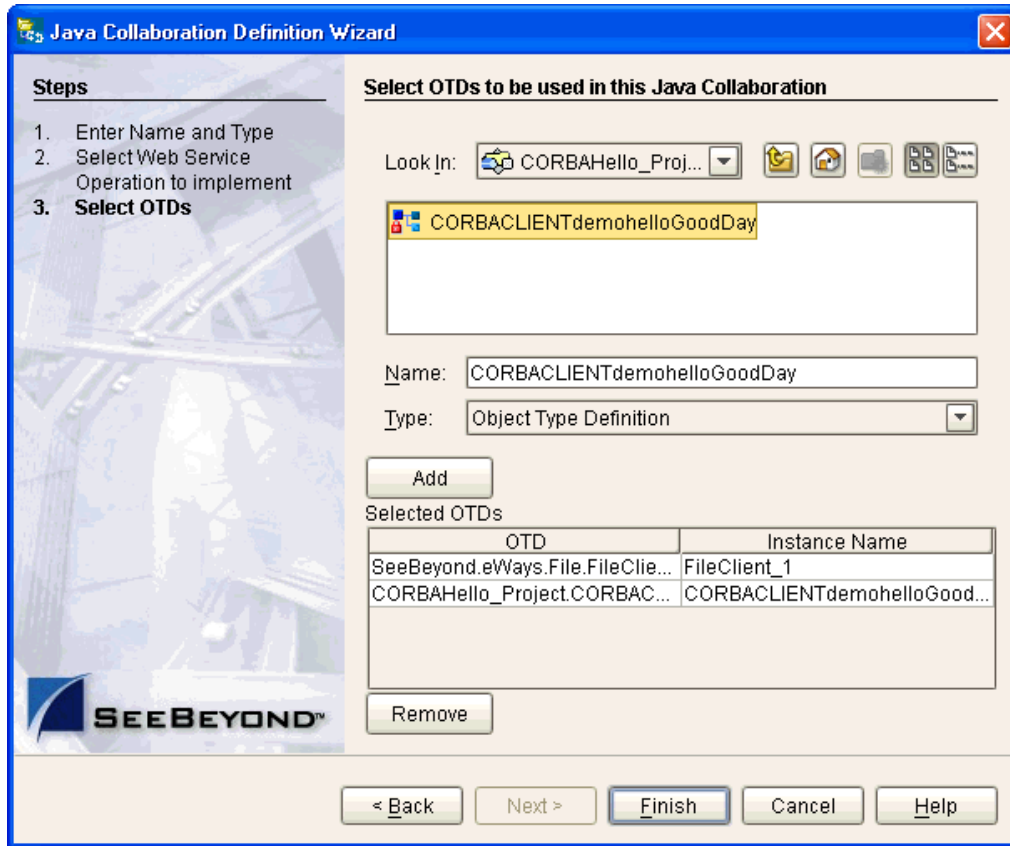
The **Selected OTDs** pane now lists **FileClient**.

- 6 At the main selection pane, return to the top of the selection tree.
- 7 From the main selection pane, double-click in succession the icons for **CORBAHello\_Sample\_Project > CORBACLIENTdemohelloGoodDay**.

See [Figure 10 on page 41](#).



**Figure 10** Java Collaboration Definition Wizard With OTDs: CORBAHello\_Sample\_Project



- 8 Click **Finish** to create the Java Collaboration Definition **FileIn2CorbaHello2FileOut**.

The Java Collaboration Editor with the new **FileIn2CorbaHello2FileOut** Collaboration Definition icon appears in the **Project Explorer** pane.

## 4.5.8 Creating Business Rules Within a Collaboration Definition

You can customize Collaboration Definitions by mapping Business Rules using the Java Collaboration Editor window in the Enterprise Designer. By clicking and dragging, you map items in the Transformation Designer areas with other desired items via their icons.

### Using the Java Collaboration Editor

This section explains generally how to create the Business Rules used in the sample Project.

The Java Collaboration Editor window displays in the Enterprise Designer after you create a new Java Collaboration Definition. You can also open this Editor by right-clicking on the name of the desired Collaboration Definition in the **Project Explorer** and choosing **Open** from the pop-up menu.

Use the Java Collaboration Editor to create the Business Rules that make up a Collaboration Definition. See the *eGate Integrator User's Guide* for details on how to use this Editor.

### To create Business Rules within FileIn2CorbaHello2FileOut

- 1 From the **Project Explorer** tree, double-click the **FileIn2CorbaHello2FileOut** Collaboration Definition.

The Java Collaboration Editor displays **FileIn2CorbaHello2FileOut**.

- 2 In both panes of the Transformation Designer, double-click and expand all of the OTD nodes. Also, expand all nodes in the Business Rules pane.

There are two Business Rules under the **receive()** method in the **Business Rules** pane.

- 3 For the first Business Rule, from the left pane, right-click on the **CORBACLIENTdemohelloGoodDay** node, choosing **Select a method to call** from the pop-up menu.

A selection box appears with a list of methods.

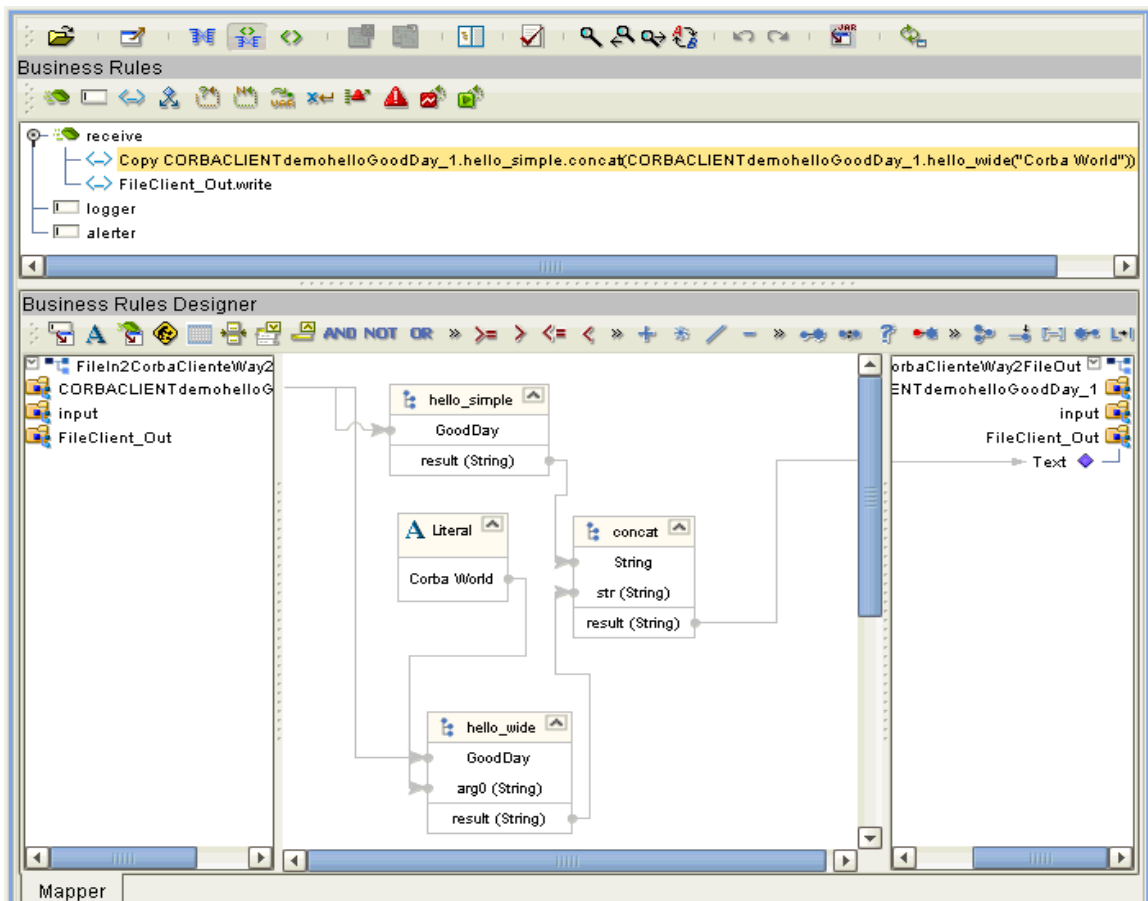
- 4 Choose the **hello\_simple** method.

A **Methods** box appears in the middle pane.

- 5 Create a **Literal** by clicking the icon in the **Business Rules Designer** toolbar and name it **Corba World**.
- 6 In the same way as you did previously, create a **hello\_wide** method from the **CORBACLIENTdemohelloGoodDay** node.
- 7 Create a **concat** method in the middle pane of the **Business Rules Designer**.
- 8 By dragging and dropping, connect the **Literal** text to **arg0 (String)** under **hello\_wide**.
- 9 In the same way, connect **result (String)** under **hello\_simple** to **String** under **concat**, and connect **result (String)** under **hello\_wide** to **str (String)** under **concat**.
- 10 Drag **result (String)** under the method **concat** to the **Text** node under **FileClient\_Out** in the right pane.

These steps create the first Business Rule. See Figure 11.

**Figure 11** FileIn2CorbaHello2FileOut Collaboration Definition: First Business Rule



- 11 Click **Save** on the Enterprise Designer toolbar to save your changes.

**Note:** *Saving your Java Collaboration Definition automatically validates it. You can ignore any error messages until you are finished with the Collaboration.*

- 12 For the second Business Rule, from the left pane, right-click on the **FileClient\_1** node, choosing **Select a method to call** from the pop-up menu.

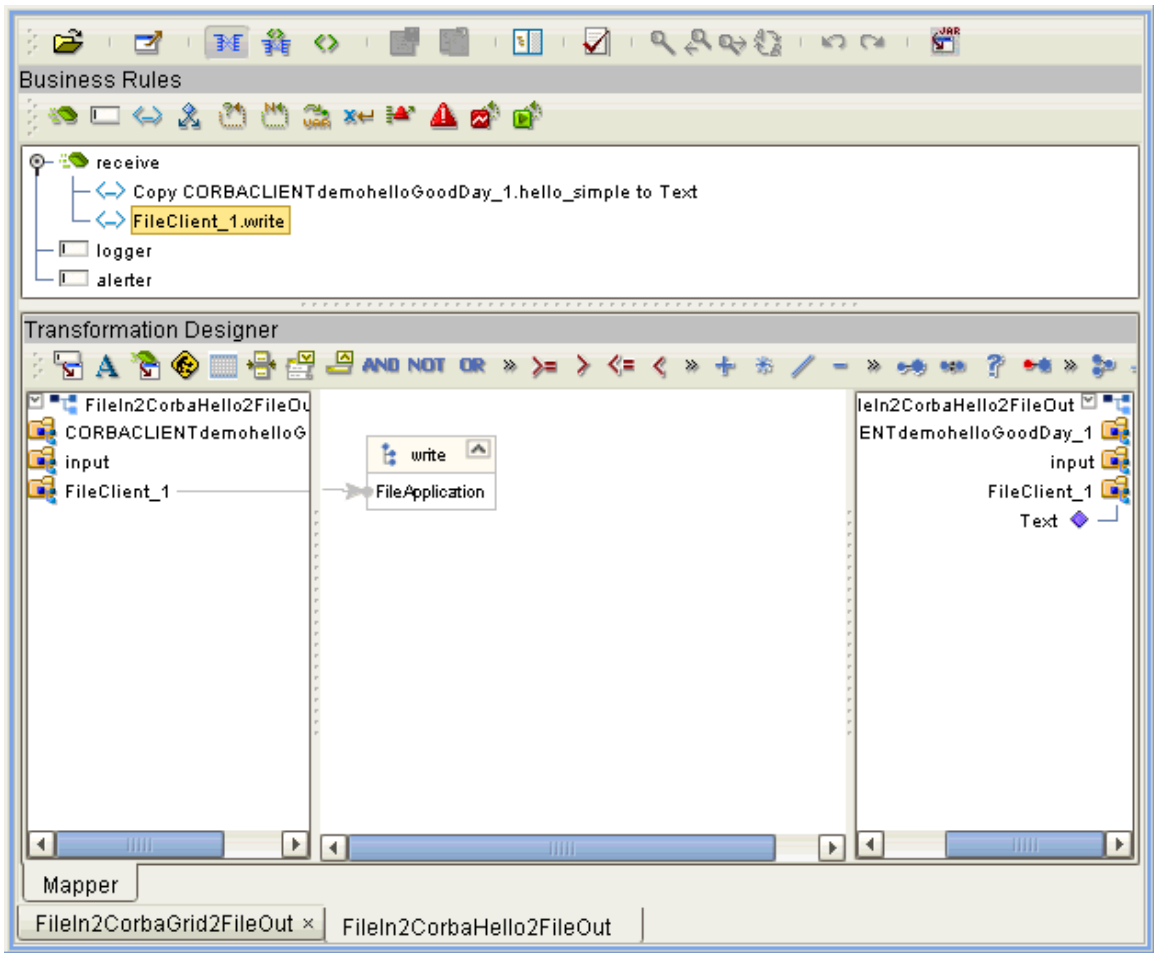
A selection box appears with a list of methods.

- 13 Choose the **write** method.

A **Methods** box appears in the middle pane.

See Figure 12.

**Figure 12** FileIn2CorbaHello2FileOut Collaboration Definition: Second Business Rule



- 14 Click **Save** to save your new Java Collaboration Definition. If the validation shows any errors, use the exception information in the Validation pane to troubleshoot the errors.

## 4.5.9 Defining the Collaboration

After you have finished with the Collaboration Definition, you need to define the Collaboration in the Connectivity Map.

To do this operation, you can drag and drop the Collaboration Definition from the **Project Explorer** tree onto its corresponding Service component in the Connectivity Map.

If the Collaboration is successfully defined, the “gears” icon changes from red to green.

*Note:* You can also drag the desired Collaboration Definition onto the Connectivity Map by itself and create a Collaboration without first creating the Service component.

#### 4.5.10 Using the Collaboration Binding Window

After you have defined the Collaborations, you need to bind, that is, associate each OTD with its corresponding eGate component.

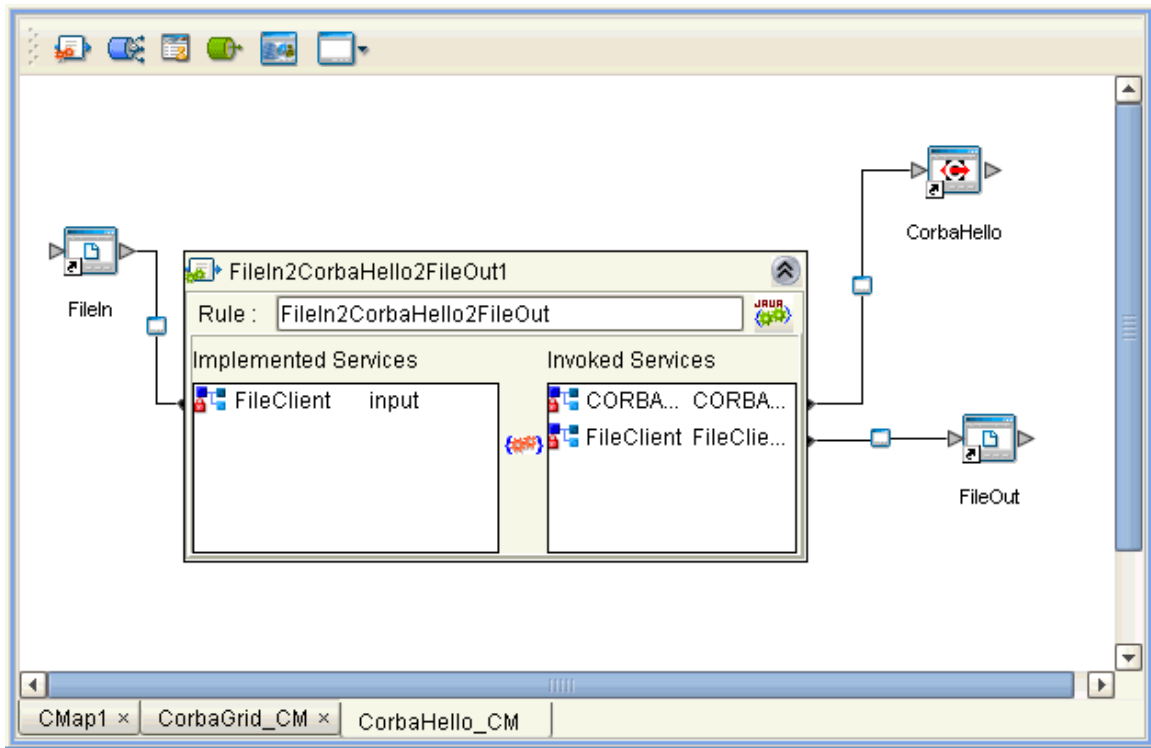
Use the eGate Enterprise Designer’s Collaboration Binding window to associate OTDs and their corresponding eWay or Topic components.

##### To bind the OTDs in the Collaboration

- 1 From the Connectivity Map, double-click the Collaboration component to open the Collaboration Binding window.
- 2 Locate the **FileClient input** OTD in the **Implemented Services** pane on the **FileIn2CorbaHello2FileOut1** window.
- 3 Click the **FileClient input** OTD and with the left mouse button depressed, drag it onto the Connectivity Map canvas and drop it onto the **FileIn** external application.
- 4 Following this same procedure, drag and drop the **CORBACLIENTdemohelloGoodDay\_1** OTD onto the **CorbaHello** external application.
- 5 Following this same procedure, drag and drop the **FileClient\_1** OTD onto the **FileOut** external application.

The Collaboration bindings appear with the components bound to the appropriate OTDs. If the binding is done correctly, the lines leading out from each OTD definition in the **Rule** text box line up with each associated component. See Figure 13.

**Figure 13** Connectivity Map: Binding the Collaboration, CORBAHello\_Sample\_Project



**Note:** For more information about assigning Collaboration Definitions and Collaboration Binding, see the *eGate Integrator User's Guide*.

- 6 To save all of your bindings, click **Save** on the Enterprise Designer toolbar.

### 4.5.11. Creating the Project's Environment

This section provides general procedures for creating an Environment for your Project. For a complete explanation, see the *eGate Tutorial*.

#### To create an Environment

- 1 From the Enterprise Designer, click the **Environment Explorer** tab on the Enterprise Explorer.
- 2 Under the current **Repository** icon in the **Environment Explorer**, create a new environment for your Project and name it as desired.

- 3 In the **Environment Explorer**, right-click the **Environment** icon and select the desired external systems from the pop-up menu. Include external systems for the CORBA Client eWay and the File eWays. Give them the same names as you did the corresponding external applications on the Connectivity Map.
- 4 Use the same pop-up menu to create a Logical Host for your Project, and name it as desired.
- 5 Click **Save All** and return to the **Project Explorer** tab.

## 4.5.12 Setting eWay Properties

You must set the eWay properties for your specific system and for the current Project, using the eGate Enterprise Designer. For directions on accessing and using the eWay **Properties** sheet, as well as a complete explanation of the CORBA Client eWay properties, see [Chapter 3](#).

### To set properties for the File eWays

- 1 From the **Project Explorer**, open the Connectivity Map for the sample Project.
- 2 To change the default properties for the inbound File eWay, double-click the **FileIn** external application's **eWay** icon. For this eWay, select the **Inbound** properties.

The eWay **Properties** sheet appears. This eWay can use the current default properties settings for the sample Project. However, if you are using different file names and/or folders than the defaults, you must enter the names of the files/folders you are using.

Be sure to use the input file name shown under **"Sample Description"** on page 36.

**Note:** *Even if you do not change the eWay's properties, you must open each **Properties** sheet for every eWay and click **OK** to activate the eWay.*

- 3 Click **OK** to save the settings and close the eWay Properties sheet.
- 4 To change the default properties for the outbound File eWay, click the **FileOut** external application's **eWay** icon. For this eWay, select the **Outbound** properties.

Use your file and folder names if they differ from the defaults. For all other properties, you can use the defaults and file names.

Be sure to use the output file name shown under **"Sample Description"** on page 36.

- 5 Click **OK** to close the window and save.

### To set properties for the CORBA Client eWay

- 1 To change the default properties for the CORBA Client eWay, click the **Environment Explorer** tab.
- 2 In the **Environment Explorer**, right-click the new Environment you created for your Project, and select **New CORBA Client External System**. Give this system the same name as you gave the corresponding external application (**CorbaHello**) created on the Connectivity Map in the **Project Explorer**.

If you have already done these operations, you can skip this step.

- 3 In the left pane, right-click the **CorbaHello** external system icon and select **Properties** from the pop-up menu.  
The eWay **Properties** sheet appears.
- 4 From the upper left pane of the eWay **Properties** sheet, select the **Corba Client Settings** folder.
- 5 The properties settings appear in the **Properties** pane on the left.
- 6 For the CORBA-specific information, enter the settings appropriate to the IOR or other CORBA object reference you are using (see [“CORBA Object Properties” on page 13](#) for details). For all other settings, use the default properties.
- 7 Click **OK** to close the window and save.

### Project Deployment

For information on how to deploy your Project, see [“Deploying a Project” on page 59](#).

---

## 4.6 Building the Project: CORBAGrid Sample

This section explains the general steps and operations you need to follow to build the sample Project **CORBAGrid\_Sample\_Project** for the CORBA Client eWay.

### 4.6.1 Sample Description

The CORBA Client eWay sample Project demonstrates how the eWay processes more complex information from a CORBA system. The resulting information is also written to a text file.

#### Importing the Project

- For an explanation of how to import the sample Project, see the [procedure on page 35](#).
- The Project’s **.zip** file name is **CORBAGrid\_Sample\_Project.zip**; name this Project **CORBAGrid\_Sample\_Project**.

### Project Components

**CORBAGrid\_Sample\_Project** has the following components:

- ♦ External file application (inbound): **File1In**
- ♦ Inbound File eWay
- ♦ Java Collaboration for processing data: **FileIn2CorbaGrid2FileOut1**
- ♦ CORBA Client eWay
- ♦ External CORBA application: **CorbaGrid**
- ♦ Outbound File eWay
- ♦ External file application (outbound): **FileOut**



## Project Operation

CORBAGrid\_Sample\_Project does the following operations:

- On being triggered by an external text file, takes a file from an external CORBA system (as set in the CORBA Client eWay **Properties** sheet).
- Outputs a file to an external directory; the file, named **CorbaGridoutput1.dat**, contains a height and width formatting specification.

This Project uses the following **.idl** (**server.idl**) file:

```
// grid.idl
// IDL definition of a 2-D grid:

module demo
{
  module grid2
  {
    interface MyServer
    {
      //typedef fixed <5,2> fixedT;

      readonly attribute short height; // height of the grid
      readonly attribute short width; // width of the grid

      // set the element [n,m] of the grid, to value:
      void set(in short n, in short m, in double value);

      // return element [n,m] of the grid:
      double get(in short n, in short m);

      exception MyException
      {
        string why;
      };

      short opWithException() raises( MyException );
    };
  };
};
```

Be sure to set up and communicate with a CORBA server that is compatible with this **.idl** file.

### 4.6.2 Creating Project, Connectivity Map, and External Application

Follow the same steps as those provided under the following sections:

- [“Creating a New Project” on page 37](#)
- [“Creating a Connectivity Map” on page 38](#)
- [“Selecting External Applications” on page 38](#)

Name this Project **CORBAGrid\_Sample\_Project**.

### 4.6.3 Populating the Connectivity Map

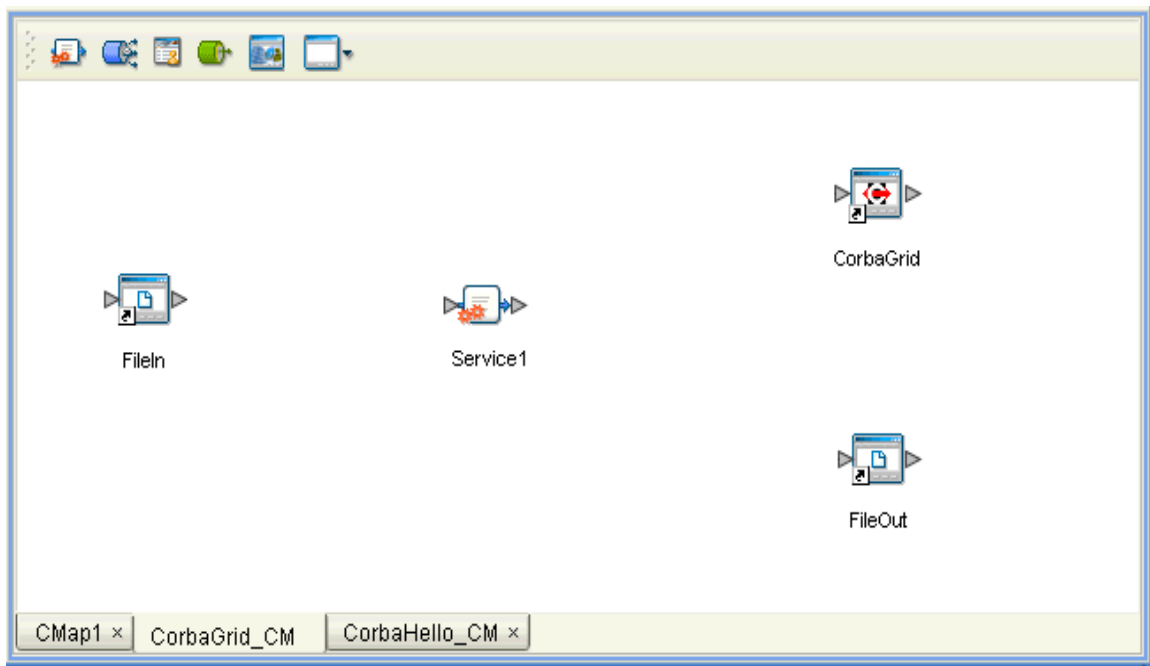
Add the Project components to the Connectivity Map by dragging the icons from the toolbar to the canvas. This operation creates the components for you.

For this sample Project, drag and drop the following components onto the Connectivity Map canvas.

- Two File eWays/external applications
- One Service
- CORBA Client eWay/external application

Figure 14 shows the components on the Connectivity Map.

**Figure 14** Connectivity Map With Components: CORBAGrid\_Sample\_Project



Rename the **Service1** component to **FileIn2CorbaGrid2FileOut1**. Name the other components as shown in the previous figure. Be sure to save the new Connectivity Map before you proceed. You can click **Save** on the Enterprise Designer toolbar for this purpose.

#### Creating an OTD

Using the **server.idl** file for this Project, create a CORBA Client OTD using the OTD wizard. See [“Using the CORBA Client OTD Wizard” on page 29](#) for complete information on how to use this wizard.

Be sure to name the new CORBA Client OTD **CORBACLIENTdemogrid2MyServer**.

## 4.6.4 Creating the Java Collaboration Definitions

You must use the Java Collaboration Definition wizard to create a Java Collaboration Definition. After a Collaboration Definition is created, you can use the Java Collaboration Editor to create the Business Rules logic of the Collaboration.

*Note:* See the *eGate Integrator User's Guide* for complete information on editing Collaborations.

### To create the Java Collaboration Definitions

- 1 From the **Project Explorer** pane, right-click the sample Project and select **New** from the shortcut menu, and **Java Collaboration Definition** from the submenu.

The **Java Collaboration Definition** wizard appears.

- 2 Enter the Collaboration Definition name, **FileIn2CorbaGrid2FileOut**, and click **Next**.

- 3 For the wizard's **Step 2**, you must select a Web Services Interface. From the main selection pane, click **Existing Web Service** then double-click in succession the icons for **SeeBeyond > eWays > File > FileClient > receive**.

The **Name** text box now displays **receive**.

- 4 Click **Next**.

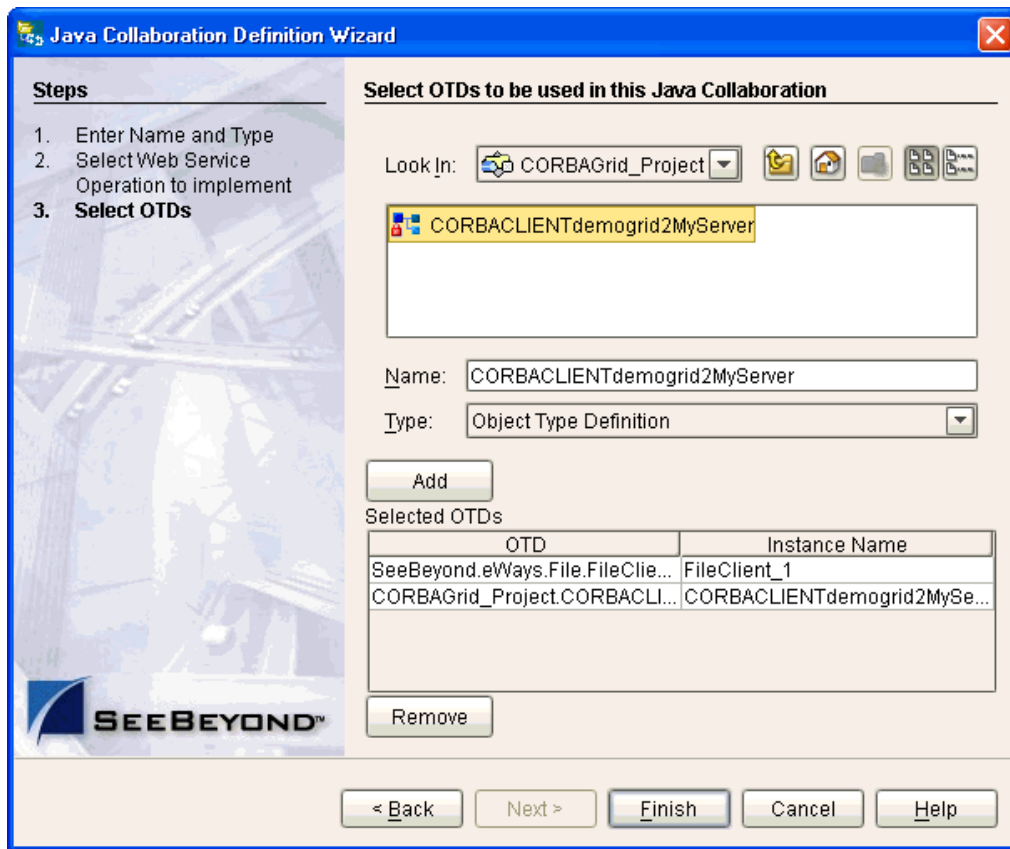
- 5 For the wizard's **Step 3**, to select the OTD, double-click in succession the icons for **SeeBeyond > eWays > File > FileClient**.

The **Selected OTDs** pane now lists **FileClient**.

- 6 From the main selection pane, double-click in succession the icons for **CORBAGrid\_Sample\_Project > CORBACLIENTdemogrid2MyServer**.

See [Figure 15 on page 52](#).

Figure 15 Java Collaboration Definition Wizard With OTDs: CORBA Grid\_Sample\_Project



- 7 Click **Finish** to create the Java Collaboration Definition **FileIn2CorbaGrid2FileOut**.  
The Java Collaboration Editor with the new **FileIn2CorbaGrid2FileOut** Collaboration Definition icon appears in the **Project Explorer** pane.

#### 4.6.5 Creating Business Rules Within a Collaboration Definition

This section explains generally how to create the Business Rules used in the sample Project.

##### Using the Java Collaboration Editor

The Java Collaboration Editor window displays in the Enterprise Designer after you create a new Java Collaboration Definition. You can also open this Editor by right-clicking on the name of the desired Collaboration Definition in the **Project Explorer** and choosing **Open** from the pop-up menu.

See the *eGate Integrator User's Guide* for details on how to use this Editor.

##### To create Business Rules within FileIn2CorbaGrid2FileOut

- 1 From the **Project Explorer** tree, double-click the **FileIn2CorbaGrid2FileOut** Collaboration Definition.

The Java Collaboration Editor displays **FileIn2CorbaGrid2FileOut**.

- 2 In both panes of the Transformation Designer, double-click and expand all of the OTD nodes. Also, expand all nodes in the Business Rules pane.

There are two Business Rules under the **receive()** method in the **Business Rules** pane.

- 3 For the first Business Rule, from the left pane, right-click on the **CORBACLIENTdemogrid2MyServer\_1** node, choosing **Select a method to call** from the pop-up menu.

A selection box appears with a list of methods.

- 4 Choose the **height()** method.

A **Methods** box appears in the middle pane.

- 5 From the pop-up box, choose the **width()** method.

A second **Methods** box appears in the middle pane.

- 6 Use the **Literal** icon to create a literal.

A **Literal** box appears in the middle pane.

- 7 Enter the following text in the new **Literal** text pop-up box:

Height =

- 8 Use the **Literal** icon to create a literal.

A second **Literal** box appears in the middle pane.

- 9 Enter the following text in the new **Literal** text pop-up box:

Width =

**Note:** Leave some spaces before "Width" to separate it from the "Height" text.

- 10 Use the **Concatenate** icon to create a concatenation.

A **concat** box appears in the middle pane.

- 11 Drag **Height =** from the literal to **String** under the new **concat**.

- 12 Drag **result (short)** under the method **height** to **str (String)** under **concat**.

- 13 Use the **Concatenate** icon to create a concatenation.

A second **concat** box appears in the middle pane.

- 14 Drag **Width =** from the literal to **String** under the second **concat**.

- 15 Drag **result (short)** under the method **width** to **str (String)** under the second **concat**.

- 16 Use the **Concatenate** icon to create a concatenation.

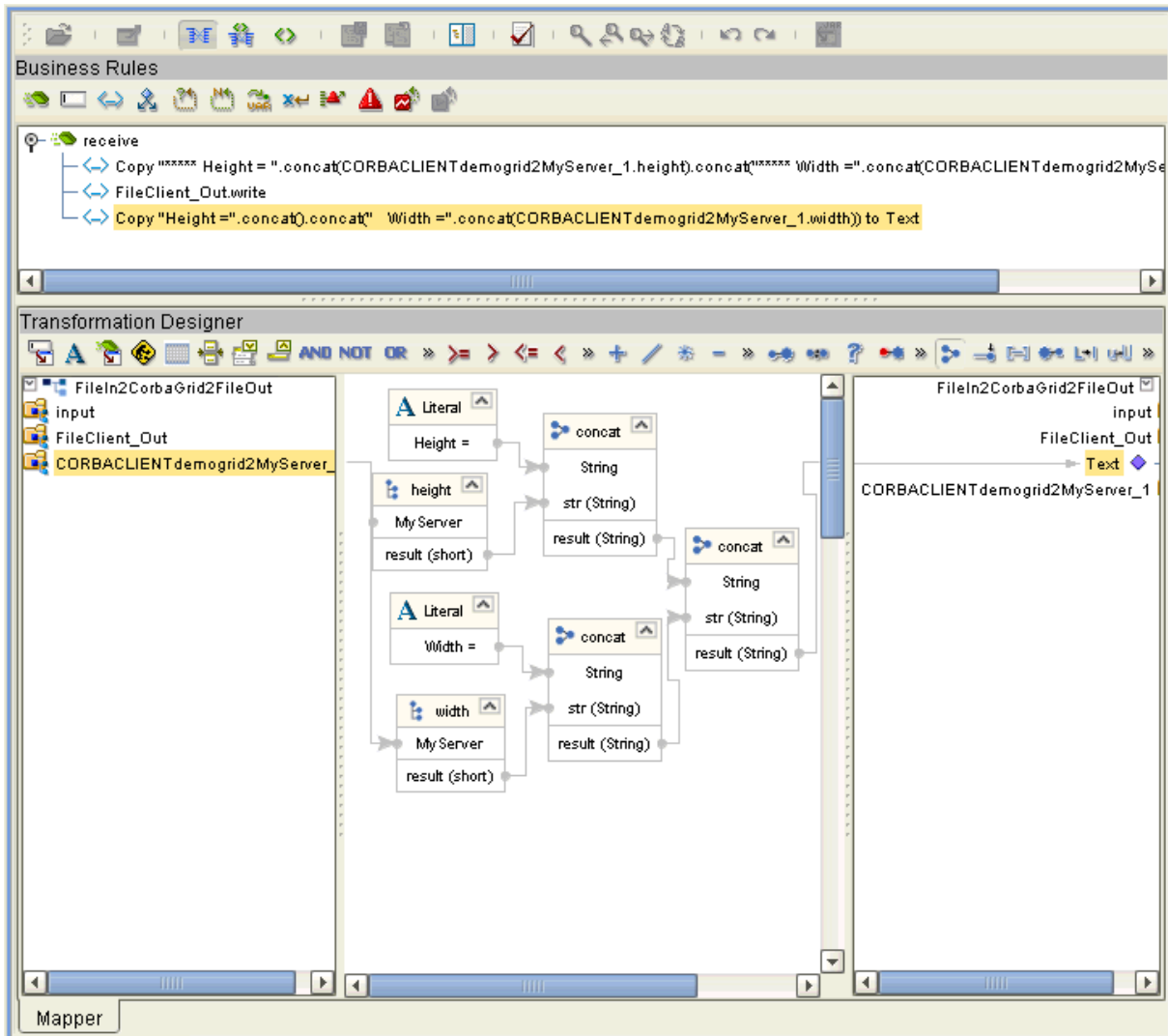
A third **concat** box appears in the middle pane.

- 17 Drag **result (String)** from the first **concat (height)** to **String** under the third **concat**.

- 18 Drag **result (String)** from the second **concat (width)** to **str (String)** under the third **concat**.

- 19 Drag **result (String)** under the third **concat** to the **Text** node under **FileClient\_Out** in the right pane. See [Figure 16 on page 54](#).

**Figure 16** FileIn2CorbaGrid2FileOut Collaboration Definition: First Business Rule, Started

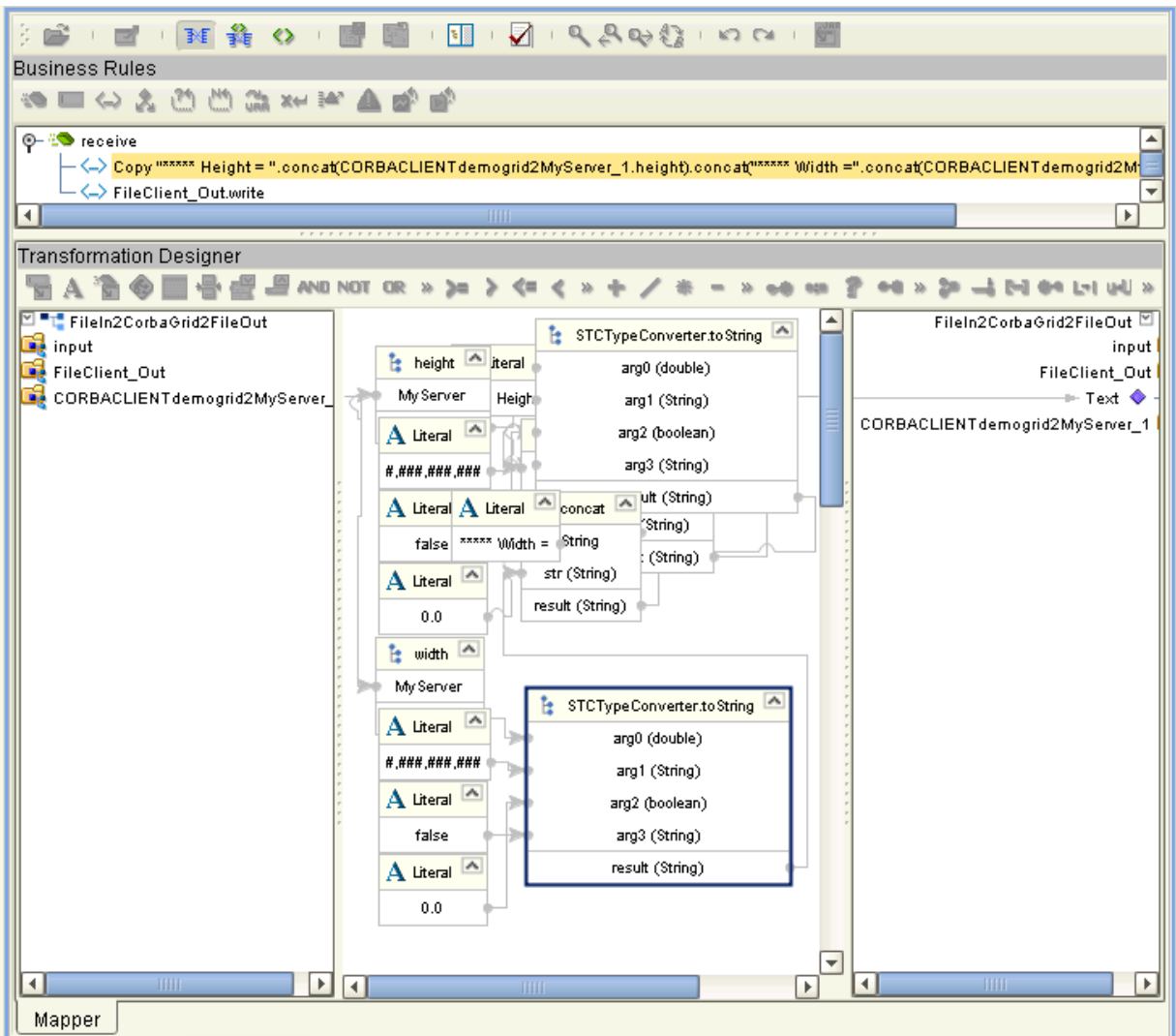


20 Click **Save** on the Enterprise Designer toolbar to save your changes.

*Note:* Saving your Java Collaboration Definition automatically validates it. You can ignore any error messages until you are finished with the Collaboration.

21 The middle pane of the Transformation Designer changes somewhat. Two **STCTypeConverter.toString** boxes appear in the middle pane. See [Figure 17 on page 55](#). These boxes complete the string conversion process.

Figure 17 FileIn2CorbaGrid2FileOut Collaboration Definition: First Business Rule, Finished



In Figure 17, one of the **STCTypeConverter.toString** boxes is selected.

These steps create the first Business Rule.

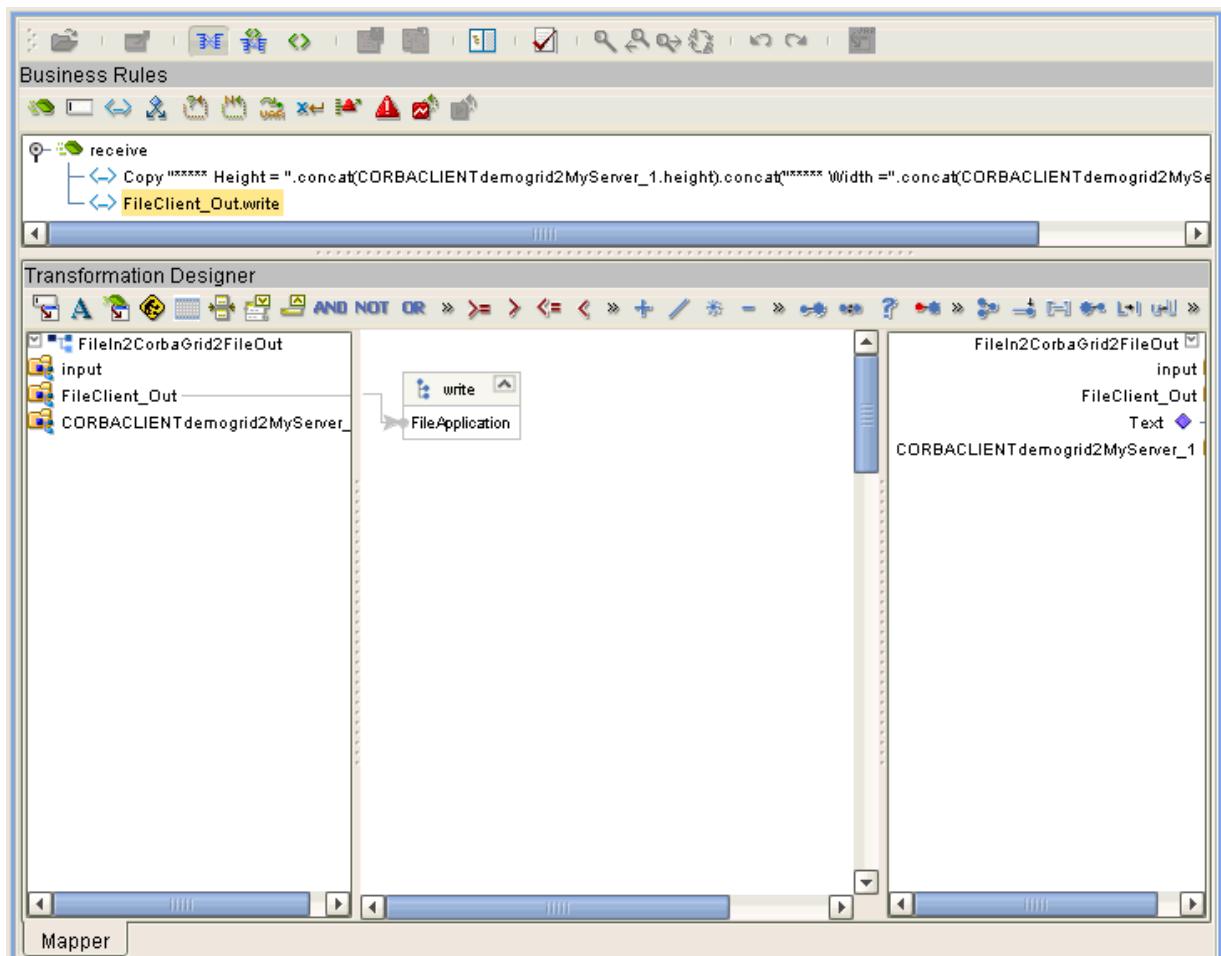
- 22 Click **Save** to save your changes.
- 23 For the second Business Rule, from the left pane, right-click on the **FileClient\_Out** node, choosing **Select a method to call** from the pop-up menu.

A selection box appears with a list of methods.

- 24 Choose the **write()** method.

A **Methods** box appears in the middle pane. See [Figure 18 on page 56](#).

**Figure 18** FileIn2CorbaGrid2FileOut Collaboration Definition: Second Business Rule



- 25 Click **Save** to save your new Java Collaboration Definition. If the validation shows any errors, use the exception information in the Validation pane to troubleshoot the errors.

## 4.6.6 Defining the Collaboration

After you have finished with the Collaboration Definition, you need to define the Collaboration in the Connectivity Map.

To do this operation, you can drag and drop the Collaboration Definition from the **Project Explorer** tree onto its corresponding Collaboration (Service) component in the Connectivity Map.

If the Collaboration is successfully defined, the **Service** icon changes to a **Java Collaboration** icon.

## 4.6.7 Using the Collaboration Binding Window

After you have defined the Collaborations, you need to bind, that is, associate each OTD with its corresponding eGate component.



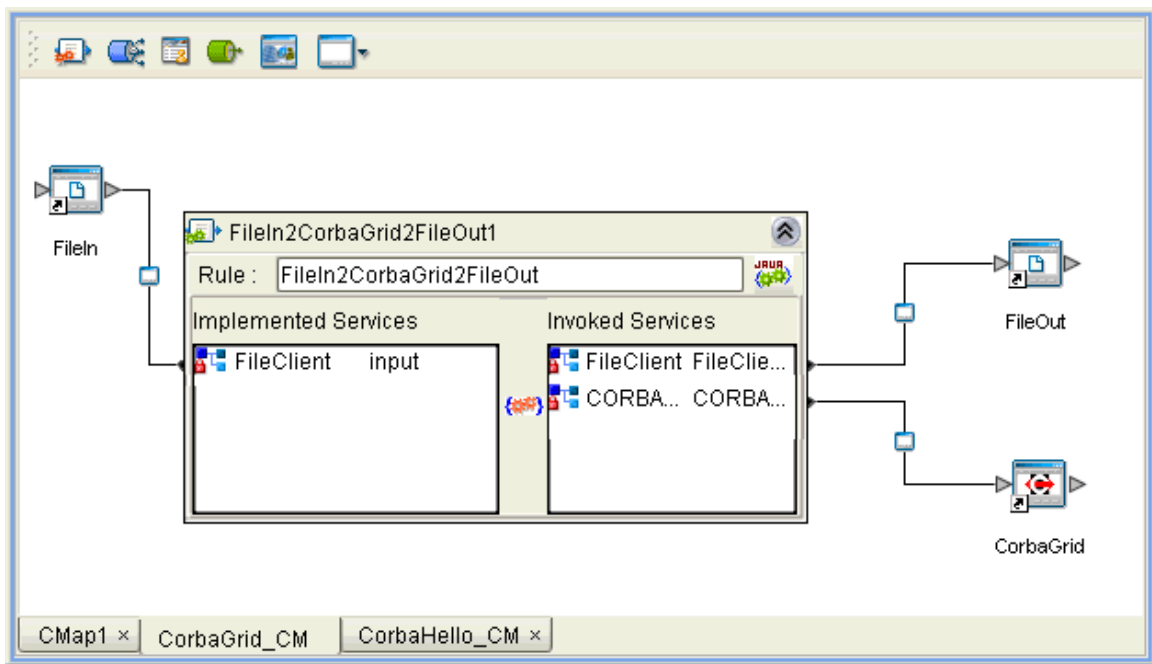
Use the eGate Enterprise Designer’s Collaboration Binding window to associate OTDs and their corresponding eWay or Topic components.

#### To bind the OTDs in the Collaboration

- 1 From the Connectivity Map, double-click the Collaboration component to open the Collaboration Binding window.
- 2 Locate the **FileClient input** OTD in the **Implemented Services** pane on the **Collaboration2** window.
- 3 Click the **FileClient input** OTD and with the left mouse button depressed, drag it onto the Connectivity Map canvas and drop it onto the **FileIn** external application.
- 4 Following this same procedure, drag and drop the **CORBACLIENTdemogrid2MyServer\_1** OTD onto the **CorbaGrid** external application.
- 5 Following this same procedure, drag and drop the **FileClient\_1** OTD onto the **FileOut** external application.

The Collaboration bindings appear with the components bound to the appropriate OTDs. If the binding is done correctly, the lines leading out from each OTD definition in the **Rule** text box line up with each associated component. See Figure 13.

**Figure 19** Connectivity Map: Binding the Collaboration, CORBAGrid\_Sample\_Project



- 6 To save all of your bindings, click **Save** on the Enterprise Designer toolbar.

### 4.6.8. Creating the Project’s Environment

See [“Creating the Project’s Environment” on page 46](#) for details.

## 4.6.9 Setting eWay Properties

For directions on accessing and using the eWay **Properties** sheet, as well as a complete explanation of the CORBA Client eWay properties, see [Chapter 3](#).

### To set properties for the File eWays

- 1 From the **Project Explorer**, open the Connectivity Map for the sample Project.
- 2 To change the default properties for the inbound File eWay, double-click the **FileIn** external system's eWay icon. For this eWay, select the **Inbound** properties.

The eWay **Properties** sheet appears. This eWay can use the current default properties settings for the sample Project. However, if you are using different file names and/or folders than the defaults, you must enter the names of the files/folders you are using.

Be sure to use the input file name shown under **"Sample Description"** on page 48.

**Note:** *Even if you do not change the eWay's properties, you must open each **Properties** sheet for every eWay and click **OK** to activate the eWay.*

- 3 Click **OK** to save the settings and close the eWay **Properties** sheet.
- 4 To change the default properties for the outbound File eWay, click the **FileOut** external system's eWay icon. For this eWay, select the **Outbound** properties.  
Use your file and folder names if they differ from the defaults. For all other properties, you can use the defaults and file names.  
Be sure to use the output file name shown under **"Sample Description"** on page 48.
- 5 Click **OK** to close the window and save.

### To set properties for the CORBA Client eWay

- 1 To change the default properties for the CORBA Client eWay, click the **Environment Explorer** tab.
- 2 In the **Environment Explorer**, right-click the new Environment you created for your Project, and select **New CORBA Client External System**. Give this system the same name as you gave the corresponding external application (**CorbaGrid**) created on the Connectivity Map in the **Project Explorer**.  
You can skip this step if you have already done this operation.
- 3 In the left pane, right-click the **CorbaGrid** external system icon and select **Properties** from the pop-up menu.  
The eWay **Properties** sheet appears.
- 4 From the upper left pane of the eWay **Properties** sheet, select the **Corba Client Settings** folder.
- 5 The properties settings appear in the **Properties** pane on the left.

- 6 For the CORBA-specific information, enter the settings appropriate to the IOR or other CORBA object reference you are using (see **“CORBA Object Properties” on page 13** for details). For all other settings, use the default properties.
- 7 Click **OK** to close the window and save.

### Project deployment

For information on how to deploy your Project, see the next section.

---

## 4.7 Deploying a Project

This section provides general procedures for Project deployment.

### 4.7.1 Basic Steps

For a complete explanation of how to deploy and run an eGate Project, see the *eGate Tutorial*.

#### To deploy the Project

- 1 From the **Project Explorer**, select the current Project and right-click, choosing **New > Deployment Profile** from the pop-up menus.
- 2 From the **Create a Deployment Profile** dialog box, enter the name of the current Project and select the Environment you created for this Project.
- 3 Click **OK**.

The Deployment Profile canvas appears as follows:

- ♦ The Project’s external applications and Services show up as icons on the left side of the canvas.
  - ♦ The external systems and Logical Host you created under **“Creating the Project’s Environment” on page 57** show up as windows on the right side of the canvas.
- 4 Set up your Deployment Profile by dragging the icons on the left into the corresponding windows on the right.
  - 5 Click **Save All** then **Activate**.

When the Project has been activated, a pop-up message window appears stating the activation was successful.

For additional information, see the *eGate Integrator User’s Guide* and *SeeBeyond ICAN Suite Deployment Guide*.

#### To run the Project

For instructions on how to run a Project, see the *eGate Tutorial*.

## 4.7.2 Alerting and Logging

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

# Index

## C

- Collaborations, overview 7
- Connectivity Map 26
- CORBA interfaces, description 34
- CORBA Services and ORBs 6
- CORBA types supported 8
- Corbaloc host 20
- Corbaloc object key 20
- Corbaloc port 20
- Corbaloc Properties Settings 19
- Corbaname host 18
- Corbaname name service 18
- Corbaname port 19
- Corbaname Properties Settings 18
- Corbaname stringified name 19

## E

- eWay components, basic 36
- eWay design overview 28
- eWay operation, general 6, 7
- eWay Properties sheet, using 12
- external system requirements 8

## G

- general operation, eWay 6

## I

- IOR file reference 23
- IOR ftp Password 24
- IOR ftp URL reference 24
- IOR ftp User Name 24
- IOR http URL reference 25
- IOR Properties Settings 23
- IOR reference 25

## L

- Locate Object Using 17
- Locate Object Using Property, overview 14
- Logical Host requirements 8

## N

- Naming IOR file URL 21
- Naming IOR ftp Password 22
- Naming IOR ftp URL 22
- Naming IOR ftp User Name 22
- Naming IOR http URL 22
- Naming Service IOR Properties Settings 20

## O

- Object key or stringified name 23
- Object Type Definitions, overview 7
- operating systems
  - supported 8
- operation overview, eWay 28
- ORB implementation 7
- OTD configuration and eWay operation 27
- OTD Editor 26
- OTD wizard, CORBA Client
  - overview 29
  - procedures 29

## P

- Project
  - canvas 26
  - deploying 59
  - External Environment 46
- Project sample
  - Building CORBAGrid\_Project 48
  - building CORBAHello\_Project 36
  - components 37, 48
  - importing 35
  - overview 35
- Project sample CORBAGrid\_Project
  - operation 49
  - overview 48
- Project sample CORBAHello\_Project
  - operation 37
  - overview 36
- properties template 11

## S

- Setting 11
- setting eWay properties
  - overview 11
  - procedures 11
- setting up the eWay, overview 27
- supported operating systems 8
- system requirements 8