SeeBeyond ICAN Suite

CORBA Client eWay Intelligent Adapter User's Guide

Release 5.0.2

Version 20040604113126.

Contents

Chapter 1

Introducing the CORBA Client eWay

This guide explains how to install, set properties for, and operate the SeeBeyond® Integrated Composite Application Network Suite™ (ICAN) CORBA Client eWay Intelligent Adapter, referred to as the CORBA Client eWay throughout this guide.

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

Chapter Topics

1.1 Overview

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

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

1.1.1 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 (Java).

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.

1.1.2 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 31**). 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

1.1.3 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 Integrator 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.

1.1.4 Setting Properties

The properties for the CORBA Client eWay, for convenience, allow you to set necessary parameters of operation using the eWay **Properties** dialog box in the eGate Enterprise Designer. See **Chapter 3** for details on how to set the CORBA Client eWay properties.

1.1.5 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, Windows 2000, and Windows Server 2003
- IBM AIX 5.1L and 5.2
- Sun Solaris 8 and 9
- HP NonStop Server G06.22

1.3 System Requirements

To use the CORBA Client eWay, you need:

- eGate Logical Host.
- TCP/IP network connection.

Logical Host Requirements

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

1.4 External System Requirements

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

Chapter 2

Installing the CORBA Client eWay

This chapter explains how to install the CORBA Client eWay.

Chapter Topics

2.1 Installation Procedures

During the ICAN Suite installation process, the Enterprise Manager, a Web-based application, is used to select and upload eWay and add-on files (**.sar** files) from the ICAN installation CD-ROM to the Repository.

When the Repository is running on a UNIX operating system, eWays are loaded using the Enterprise Manager on a Windows computer connected to the Repository server, using Internet Explorer.

Before installing the eWay

Open and review the **Readme.txt** file (located in the root directory of the ICAN installation's Repository CD-ROM) for the latest information, before installing the eWay.

Installing the CORBA Client eWay on an eGate-supported system

The CORBA Client eWay can be installed during the installation of eGate. The eGate installation process includes the following operations:

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

Follow the instructions for installing the ICAN Suite found in the *SeeBeyond ICAN Suite Installation Guide*, and include the following steps:

1. After the eGate or eInsight core products are uploaded to the Repository using the Enterprise Manager, select and upload the **FileeWay.sar**. The File eWay is used to by the eWay's Project sample. You must upload the File eWay (**FileeWay.sar**.) before uploading the CORBA Client eWay (**CorbaClienteWay.sar**).
2. After the File eWay is uploaded, upload **CorbaClienteWay.sar** to install the CORBA Client eWay.
3. Next, upload the **CorbaClienteWayDocs.sar**. This file contains the eWay user's guide and sample Project files.

   To obtain these files, follow the instructions provided by the user interface.
4. If needed, continue installing eGate as instructed in the *SeeBeyond ICAN Suite Installation Guide*.

2.2 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

Setting CORBA Client eWay Properties

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

Chapter Topics

3.1 CORBA Client eWay Properties Dialog Box

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.

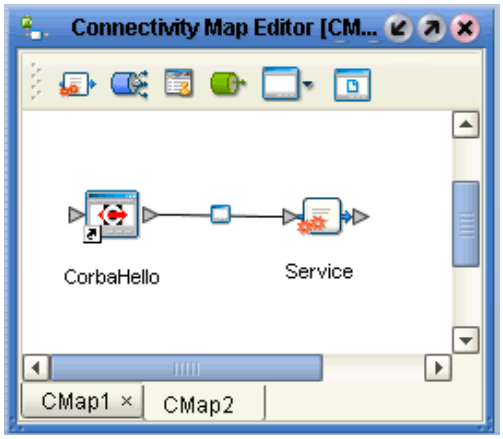
You can set properties for each individual eWay using the Enterprise Designer's eWay **Properties** dialog box. This section describes general procedures on how to change these default properties for the eWay. For details on these steps, see the *eGate Integrator User's Guide*.

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** dialog box 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** dialog box *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** dialog box 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 dialog box

- The CORBA Client eWay's properties are set using the eGate Enterprise Designer's eWay **Properties** dialog box. 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** dialog box 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:

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 dialog box 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 dialog box and explained later in this section.

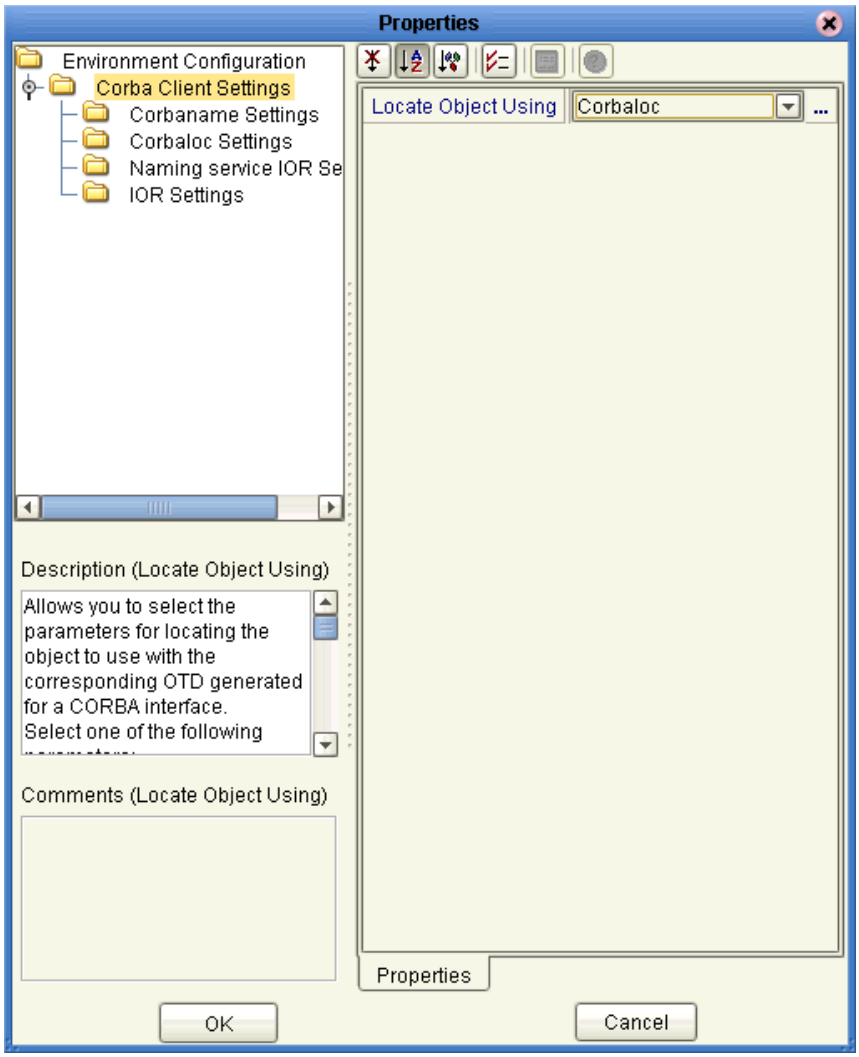
Table 1 Supported Bootstrapping Mechanisms (Continued)

Reference Format	Description	Correct Entry
Naming Service IOR reference	The CORBA Naming Service can write its own IOR to a URL.	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: ▪ Name_Path/NS_Ref (file URL) ▪ ftp://Server_Name/NS_Ref (FTP URL) (separate properties are provided for the user name and password) ▪ http://Server_Name:Port/NS_Ref (HTTP URL) Where: NS_Ref is the reference to the IOR of a CORBA Naming Service.
IOR file, HTTP URL, and FTP URL reference	These formats are supported by this eWay, allowing you to save an IOR or URI in a file, then specify that file via: ▪ File and path name, if it is in a reachable system. ▪ HTTP URL if it is published on a Web server. ▪ FTP URL if it is published in an FTP server.	Enter the appropriate character string, as follows: ▪ For a file, enter the complete path and file name. ▪ For HTTP, enter the complete URL. ▪ For FTP, enter the complete URL (separate properties are provided for the user name and password).
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** dialog box. See Figure 2.

Figure 2 eWay Properties Dialog Box: 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*** *dialog box.*

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	*None*	*None*
Corbaname Settings: See **"Corbaname Settings" on page 18**.	Corbaname	Corbaname host Corbaname service Corbaname port Corbaname stringified name
Corbaloc Settings: **"Corbaloc Settings" on page 19**	Corbaloc	Corbaloc host Corbaloc object key Corbaloc port
Naming Service IOR Settings: **"Naming Service IOR Settings" on page 20**	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
	For all selections: Naming service file, ftp, and http URL	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 strin**g (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

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:000000000000001B49444C3A64656D6F2F68656C6C6F2F476F6F644461793A312E3000000000000100000000000000060000102000000000D31302E312E3139302E323436000004EE0000001C5374616E64617264496D706C4E616D652F001041403C0A1604013D0100000001000000010000001C0000000000010001000000010501000100010109000000105010001

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.

Chapter 4

Reviewing the Sample Projects

This chapter describes how to implement the CORBA Client eWay using a review of the sample Projects included with the eWay.

This chapter assumes that you are already familiar with eGate concepts and that you understand the basics of creating a Project using the eGate Enterprise Designer.

For a complete explanation of how to use the eGate Enterprise Designer to create and set properties for the components of an eGate Project, see the *eGate Integrator User's Guide*.

Chapter Topics

4.1 eGate Project Descriptions

This section provides an overview of the eGate sample Projects for the CORBA Client eWay and how to import and use them.

4.1.1 Projects and the Enterprise Designer

A Project contains all of the eGate components you designate to perform one or more desired processes in eGate. 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 Canvas**: Contains the eGate 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.
- **Collaboration Editor (Java)**: Allows you to create and/or modify Business Rules to implement the business logic of a Project's Java-enabled Collaboration Definition.

4.1.2 Importing Sample Projects

Before you can view or work with a sample Project, you must first import it into eGate, using the Enterprise Designer.

Note: *The sample .**zip** file you first download 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. For details on how to download this file, see the **SeeBeyond ICAN Suite Installation Guide**.*

The container file you are looking for is **CORBA_Client_eWay_Sample.zip**. The Project file names are:

- **CORBAHello_Sample_Project.zip**
- **CORBAGrid_Sample_Project.zip**

You can name the imported Projects as desired.

To import a sample Project

1. From the Enterprise Designer's **Project Explorer** pane, right-click the desired Repository and select **Import**.
2. On the **Import Manager** window, browse to the directory that contains the sample Project **.zip** file.
3. Select the sample Project file and click **Open**.
4. Click **Import**. If the import was successful, click **OK** on the **Import Status** dialog box.
5. Close the **Import Manager** window and select **Refresh All from Repository** from the shortcut menu.

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

You must check out the major eGate components before you can change them.

4.1.3 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** dialog box in the eGate Enterprise Designer. For more information about CORBA Client eWay properties and the eWay **Properties** dialog box, see **Chapter 3**.

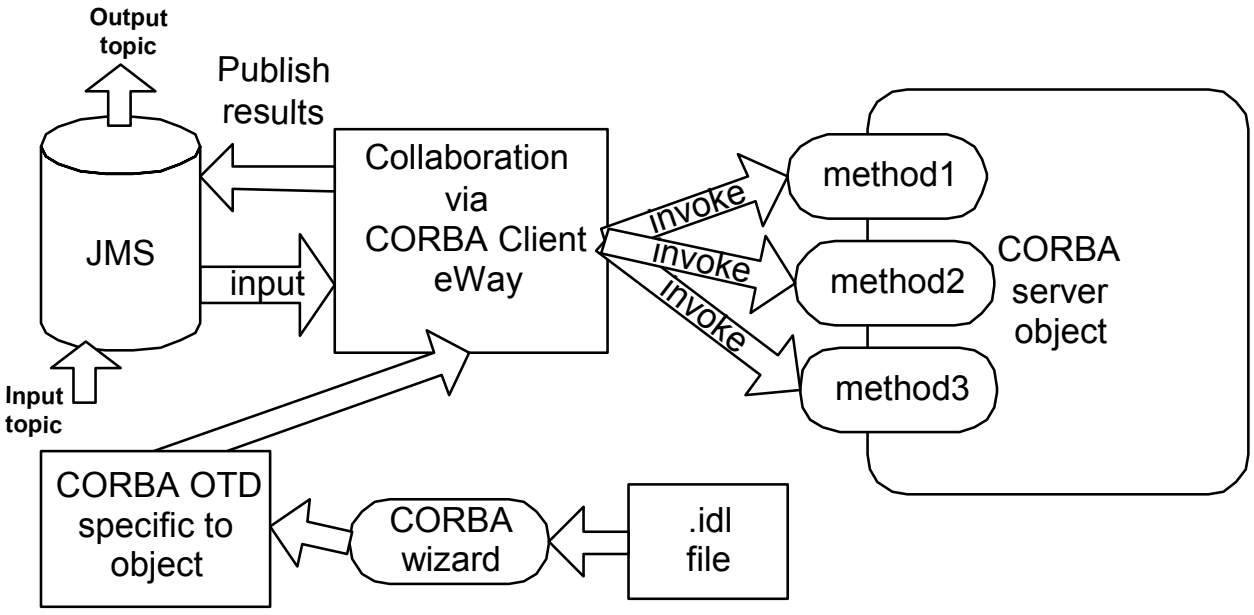
CORBA Client OTDs

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 31** for details on how to use this wizard.

4.1.4 eWay Operation 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.

4.1.5 Overview of Sample Projects

This section provides an overview of the sample Projects included with this eWay, including descriptions of how the Projects operate.

CORBAHello Project

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.

Project Components

CORBAHello_Sample_Project has the following components:

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

Project Operation and Input/output

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** dialog box), 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.

CORBAGrid Project

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.

Project Components

CORBAGrid_Sample_Project has the following components:

- External file application (inbound): **File1In**
- Inbound File eWay
- Collaboration (Java) 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** dialog box).
- 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 defintion 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.

4.2 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****.*

Accessing .jar files

The CORBA Client OTD has a feature that allows you to easily access the Java classes and methods in the corresponding OTD's **.jar** files. If there is any **.jar** file in an OTD

you create via the wizard, such files display as **Project File** icons in the **Project Explorer**.

In the Collaboration Editor (Java), this feature allows you to use the Editor's **Call Java Method** and **Call New Constructor** toolbar icons to launch the corresponding Enterprise Designer menu. Using this menu, you can access the methods and constructors of a given class.

When you use such an OTD in the Collaboration Editor (Java), this feature gives you instant access to the all classes and methods in the **.jar** files for the OTD.

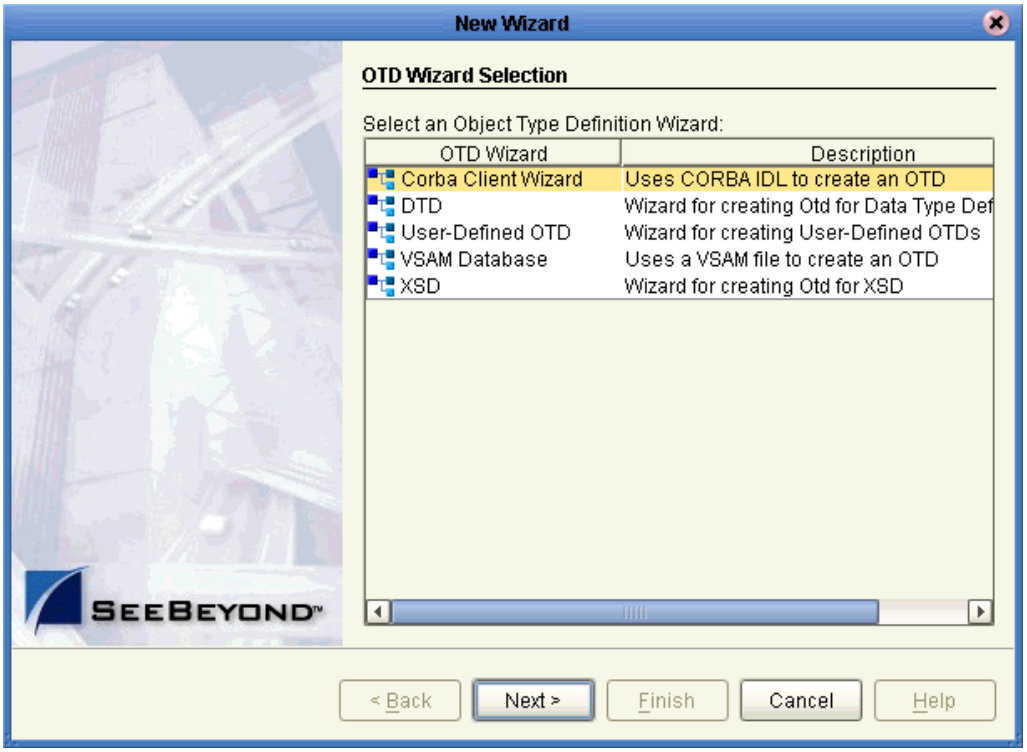
4.2.1 OTD Wizard Procedures

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 32**.

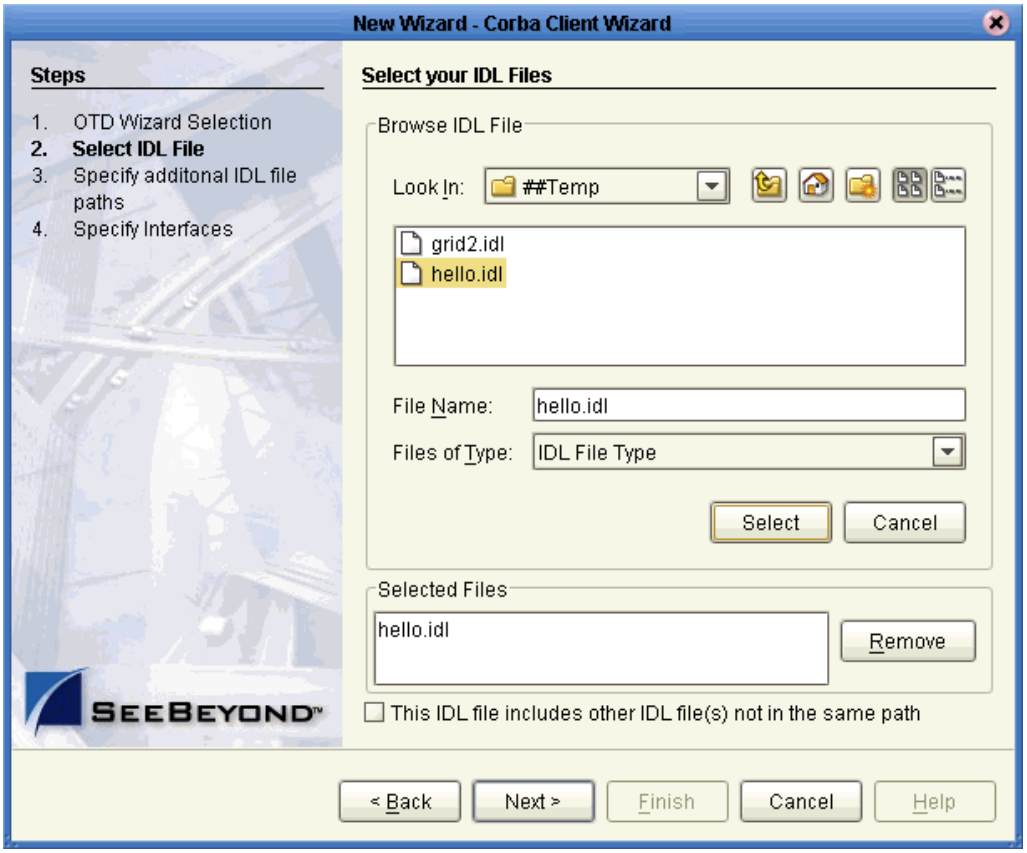
Figure 4 OTD Wizard Selection Window



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

   The **Select Your IDL Files** window appears. See **Figure 5 on page 33**.

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.*

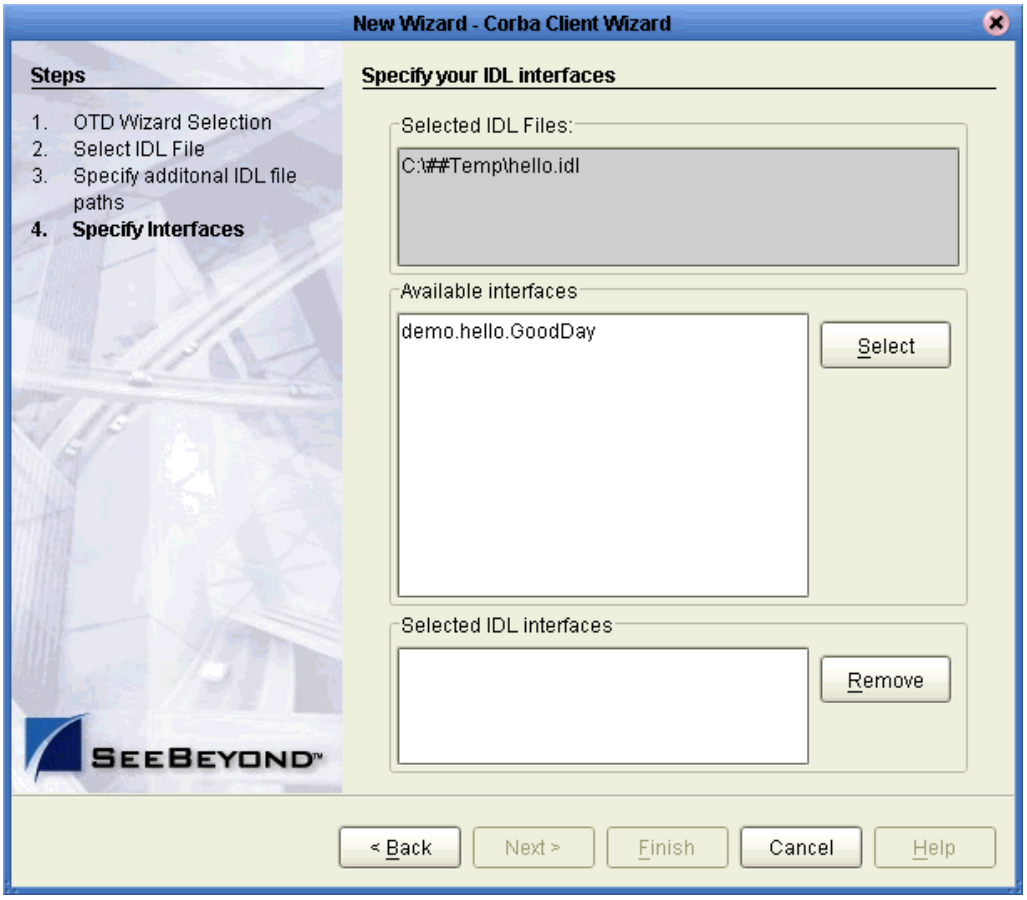
5 If the one or more file names are correct, click **Select**.

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 34**.

Figure 6 Specify Your IDL Interfaces Window 1

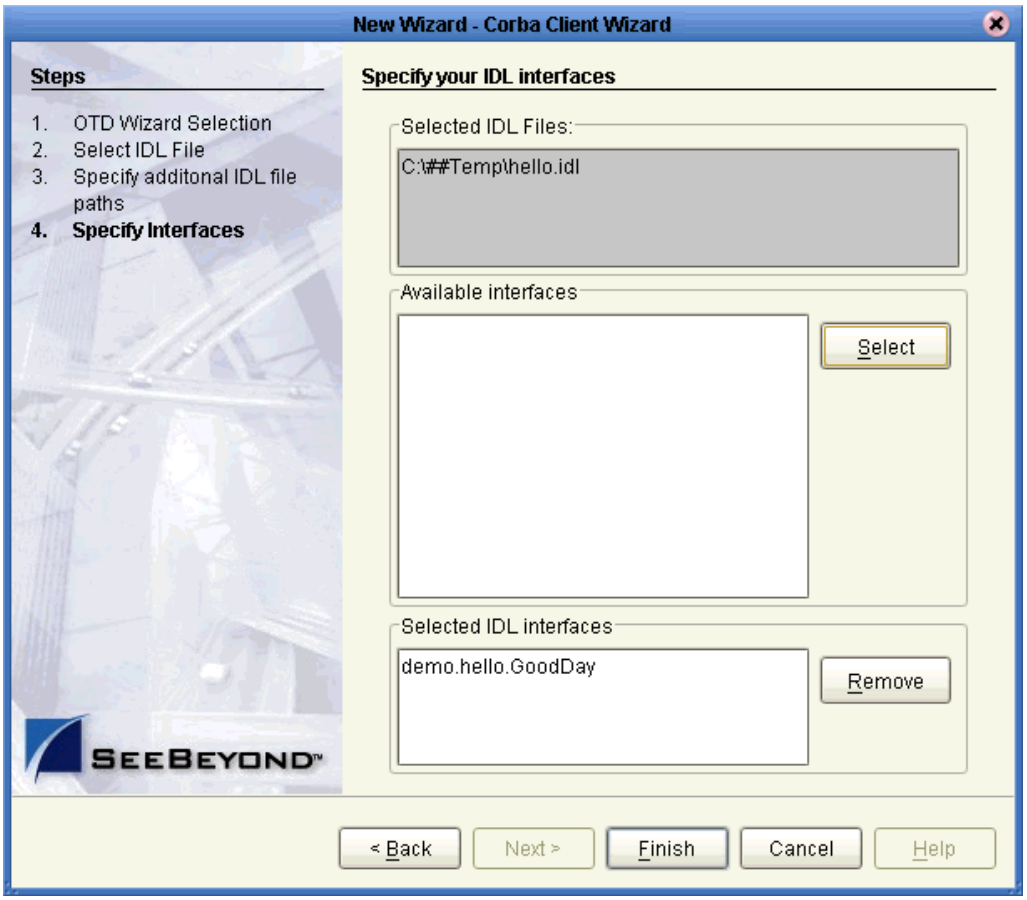


Note: *For more information on IDL interfaces, see* **"CORBA Interfaces" on page 36**.

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 35**.

Figure 7 Specify Your IDL Interfaces Window 2

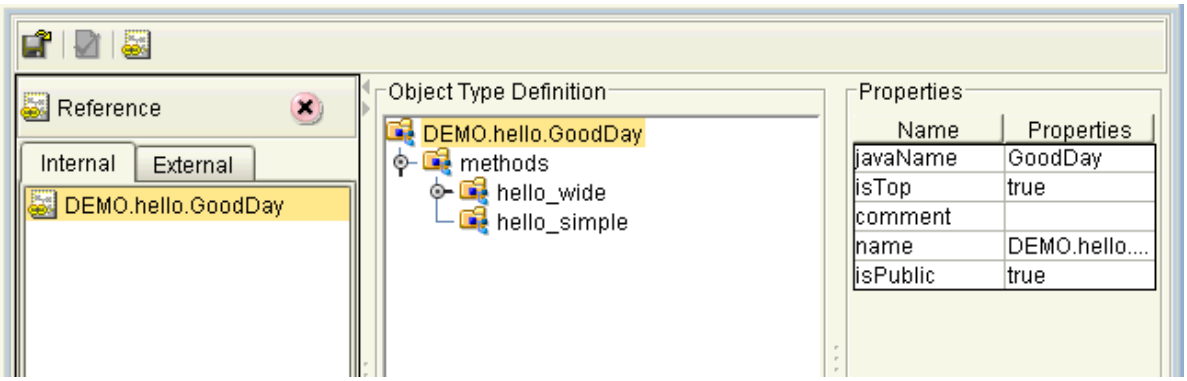


8 Review the information you have entered and be sure it is correct. You can click **Back** to change previously entered information, if necessary.

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****.*

4.2.2 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
```

4.3 Summary: Sample Projects Review

This section explains generally how to implement the CORBA Client eWay using the eGate Project samples. These samples contain Java-based Collaborations and are included on your installation CD-ROM.

The samples are named:

- **CORBAHello_Sample_Project.zip**
- **CORBAGrid_Sample_Project.zip**

These samples allow you to observe an end-to-end data-exchange scenario involving eGate and the CORBA Client eWay.

For instructions on how to import the sample Project, see the **procedure on page 44**. For an overview of the sample Project and what is does, see **"Sample Description" on page 42**.

4.3.1 Creating Collaboration Definitions

The eGate Enterprise Designer contains a Collaboration Definition wizard (Java) that allows you to create Java-based Collaborations. You must use the wizard to create a Collaboration Definition before implementing the Collaboration.

4.3.2 Using the Collaboration Editor (Java)

The Collaboration Editor (Java) window displays in the Enterprise Designer after you create a Collaboration Definition (Java). 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.

This user interface allows you to create the Business Rules that implement your business logic for a Java-enabled Collaboration. You can create the desired Business Rules for your Project by dragging and dropping values from a source OTD onto the nodes of a destination CORBA Client OTD and other OTDs. CORBA Client OTD nodes represent CORBA Client functions, which are in turn able to call CORBA Client methods.

See the *eGate Integrator User's Guide* for complete information on how to use the Collaboration Editor (Java).

4.3.3 Creating the Project's Environment

This section provides general procedures for creating an Environment for your Project. For a complete explanation, see the *eGate Integrator User's Guide.*

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 (inbound and outbound). 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** and return to the **Project Explorer** tab.

4.3.4 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** dialog box, 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 desired 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** dialog box 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.

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

3 For this sample, use the **C:\temp** as the property for **Directory**, and for **Input file name**, enter ***.txt**.

4 Click **OK** to save the settings and close the eWay **Properties** dialog box.

5 To change the default properties for the outbound File eWay, double-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.

6 Click **OK** to close the dialog box and save.

To set properties for the CORBA Client eWays

1 To begin changing the default properties for the CORBA Client eWay, double-click the **CORBAHello** or **CORBAGrid** external system's eWay icon.

The eWay **Properties** dialog box appears.

2 The properties settings appear in the **Properties** pane on the left.

3 Set the properties as desired then click **OK** to close the dialog box and save.

4 From the Enterprise Designer, click the **Environment Explorer** tab.

5 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** or **CORBAGrid**) created on the Connectivity Map in the **Project Explorer**.

For details on how to do these operations, see the *eGate Integrator User's Guide.*

6 In the left pane, right-click the **CORBAHello** or **CORBAGrid** external system icon and select **Properties** from the pop-up menu.

The eWay **Properties** dialog box appears. Use this dialog box to set additional properties settings.

7 The properties settings appear in the **Properties** pane on the left.

8 For the settings, enter the information appropriate to your system.

9 Click **OK** to close the dialog box and save.

4.4 Deploying a Project

This section provides general procedures for Project deployment.

4.4.1 Basic Steps

For a complete explanation of how to deploy and run an eGate Project, see the *eGate Integrator User's Guide* and *SeeBeyond ICAN Suite Deployment Guide.*

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 53** 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 window on the right.

5 Click **Save All** then **Activate**.

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

To run the Project

For instructions on how to run a Project, see the *eGate Integrator User's Guide*.

4.4.2 Alerting and Logging

eGate provides an alerting and logging feature. This feature allows the monitoring of messages and captures any adverse messages in the order of severity, based on a configured severity level and higher. See the *eGate Integrator User's Guide* for an explanation of how to enable the Logging feature.

Index