SeeBeyond ICAN Suite

ADABAS eWay Intelligent Adapter User's Guide

Release 5.0

SEEBEYOND®

SeeBeyond Proprietary and Confidential

Version 20031222114742.

Contents

Chapter 6

Chapter 7

Chapter 1

Introducing the ADABAS eWay

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

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

Chapter Topics

1.1 Overview

The ADABAS eWay enables eGate Integrator to exchange data with an external ADABAS mainframe file system.

ADABAS system

The ADABAS system is a database product of Software AG. This database is designed around a two-process structure. The ADABAS nucleus executes in a separate partition of the mainframe operating system from the application or the ADABAS utility.

The ADABAS nucleus manages all database activities and serves all user processes centrally. The number of users who can be served in parallel is unlimited and typically can be many thousands.

To be able to support thousands of users in parallel and deliver fast response times, ADABAS has its own internal multi-threading architecture. This architecture allows the ADABAS nucleus to support an unlimited number of read-and-write tasks simultaneously.

The ADABAS system also provides the following features:

- Supports Entry-sequenced Data Set (ESDS) and Key-sequenced Data Set (KSDS) files
- Data and index compression
- Asynchronous buffer flushing
- Buffer-search algorithms based on hashing algorithms

The ADABAS eWay allows you to utilize and extend all of these features. For more information on ADABAS, see Software AG's Web site:

ADABAS eWay

This eWay is similar to other database eWays, such as IAM, VSAM, and IDMS. The eWay uses its own properties settings and eGate components to enable ADABAS data integration.

This eWay enables eGate to communicate with ADABAS files on a mainframe host system via TCP/IP. The eWay utilizes a CrossAccess Corp. software component that allows you to view and access ADABAS in the same way as a standard relational database system.

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

The ADABAS eWay's properties settings provide the necessary parameters for the operation of the ADABAS eWay. See **Chapter 3** for details on how to set the ADABAS eWay properties.

1.2 Supported Operating Systems

The ADABAS eWay is available for the following operating systems:

- Windows XP, Windows 2000, and Windows Server 2003
- HP-UX 11.0 and 11i (RISC)
- IBM AIX 5.1 and 5.2
- Red Hat Linux 8 (Intel)
- Red Hat Linux Advanced Server 2.1 (Intel)
- Sun Solaris 8 and 9

1.3 System Requirements

To use the ADABAS eWay, you need:

- eGate Logical Host
- TCP/IP network connection.

The eWay supports ADABAS version 5.3.

Logical Host requirements

The ADABAS eWay Installation program installs the CrossAccess JDBC driver, version 3.0.

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 *SeeBeyond ICAN Suite Installation Guide*.

eInsight interface

To enable the eWay's eInsight interface (see **Chapter 5** for details), you must also install and set properties for SeeBeyond's eInsight Business Process Manager. See the *eInsight Business Process Manager User's Guide* for details.

1.4 External System Requirements

This section lists the eWay's external system requirements.

Additional Windows Requirements

To enable eGate Enterprise Designer's editors to communicate with the ADABAS file system, the following CrossAccess item must be installed on any host machines running the editors:

- CrossAccess version 3.0, including:
    - JDBC driver (installed with the eWay)

You also need the Predict application.

Note: *Predict is a standard ADABAS software tool.*

For more information, including installation procedure and serial numbers, see **Chapter 2**.

Additional UNIX Requirements

If you are installing the ADABAS eWay on a UNIX system, you need:

- CrossAccess JDBC driver, version 3.0 (installed with the eWay)

z/OS External Requirements

- CrossAccess eXadas Data Integrator, version 3.0, installed via a 3480 cartridge tape
- The serial number for the installation, shipped with the tape

Note: *The eXadas Data Integrator is shipped with the eWay, but you must install it separately.*

All of these eXadas software products are also available from the CrossAccess Corp. For more information, you can visit the company's Web site at:

www.crossaccess.com

Chapter 2

Installing the ADABAS eWay

This chapter explains how to install the ADABAS eWay.

Chapter Topics

Note: *See the* ***SeeBeyond ICAN Suite Installation Guide*** *for complete eGate installation instructions.*

2.1 Complete Installation: Overview

This section provides a brief overview of the eWay's installation components and the order of their installation.

Note: *The CrossAccess eXadas Data Integrator is shipped with the eWay, but you must install it separately.*

You must install the ADABAS eWay and its components in the following order:

1. eXadas Data Integrator
2. ADABAS eWay

In addition, the eWay requires the Predict software tool. See the appropriate Software AG documentation for complete instructions on Predict. Also, see the CrossAccess documentation for instructions on how to use Predict with CrossAccess software.

Note: *Once the installation operation is complete, you may want to pass a small amount of data to ensure you have a working connection.*

Installing eXadas Data Integrator

For complete details on how to install eXadas Data Integrator on the mainframe, see the CrossAccess *eXadas Data Integrator Getting Started* manual, the chapter that details mainframe installation. All eXadas software products, as well as additional information on each product, are available from the CrossAccess Corp.

For more information on CrossAccess, you can visit their Web site.

Important: *When you install eXadas Data Integrator, be sure to set properties for it for ADABAS. See the appropriate CrossAccess documentation for details.*

2.2 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.

2.2.1 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

2.2.2. Installing the ADABAS eWay on an eGate-supported System

The ADABAS eWay is installed during the installation of eGate Integrator. Follow the instructions in the *SeeBeyond ICAN Suite 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:
    - **ADABASeWay.sar** (to install the ADABAS eWay)
    - **FileeWay.sar** (to install the File eWay, used in the sample Project)
2. Continue installing eGate Integrator as instructed in the *SeeBeyond ICAN Suite Installation Guide.*

2.2.3. After Installation

Once the eWay is installed and set up, 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 ADABAS eWay Properties

This chapter explains how to set the properties for the ADABAS eWay.

Chapter Topics

3.1 ADABAS eWay Properties Sheet

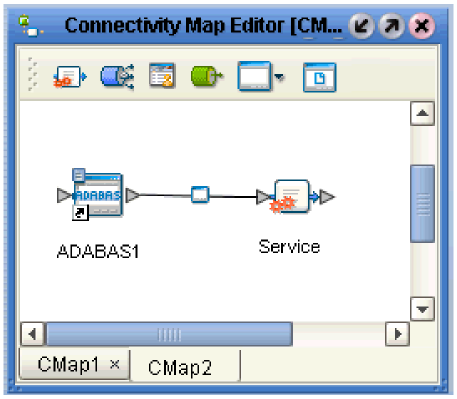
When you install the ADABAS eWay, a default properties template for the eWay is also installed. The template's default properties are accessible via the eGate Enterprise Designer. These default settings apply to all ADABAS eWays you use within your current Project or Business Process.

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 ADABAS eWay on the Connectivity Map

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.
3. Select the external application whose default eWay properties you want to change by 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 13**.

Figure 1 eWay Icon



The eWay **Properties** sheet appears. Figure 2 on page 14 shows the eWay's default properties available via the **Project Explorer** and Connectivity Map. You can use this window to modify the current eWay's properties settings.

4 Click **OK** then **Save All** to save your changes.

To set properties for the ADABAS eWay on the Project Environment

1 From the Enterprise Designer, create your external systems.

2 Click the **Environment Explorer** tab (at the bottom of the left pane).

3 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**.

4 Select the external system whose default eWay properties you want to change by right-clicking the desired system's icon in the **Environment Explorer**.

The eWay **Properties** sheet appears. Figure 3 on page 18 shows the eWay's default properties available from the **Environment Explorer**. You can use this dialog box to modify the eWay properties associated with the current external system.

5 Click **OK** then **Save All** to save your changes.

To use the eWay Properties sheet

- The ADABAS eWay's properties are set using the eGate Enterprise Designer's eWay **Properties** sheet. The default properties are automatically provided.
- Clicking the **Configuration** (Connectivity Map) or **Environment Configuration** folder in the left pane displays the properties group subfolder 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.*

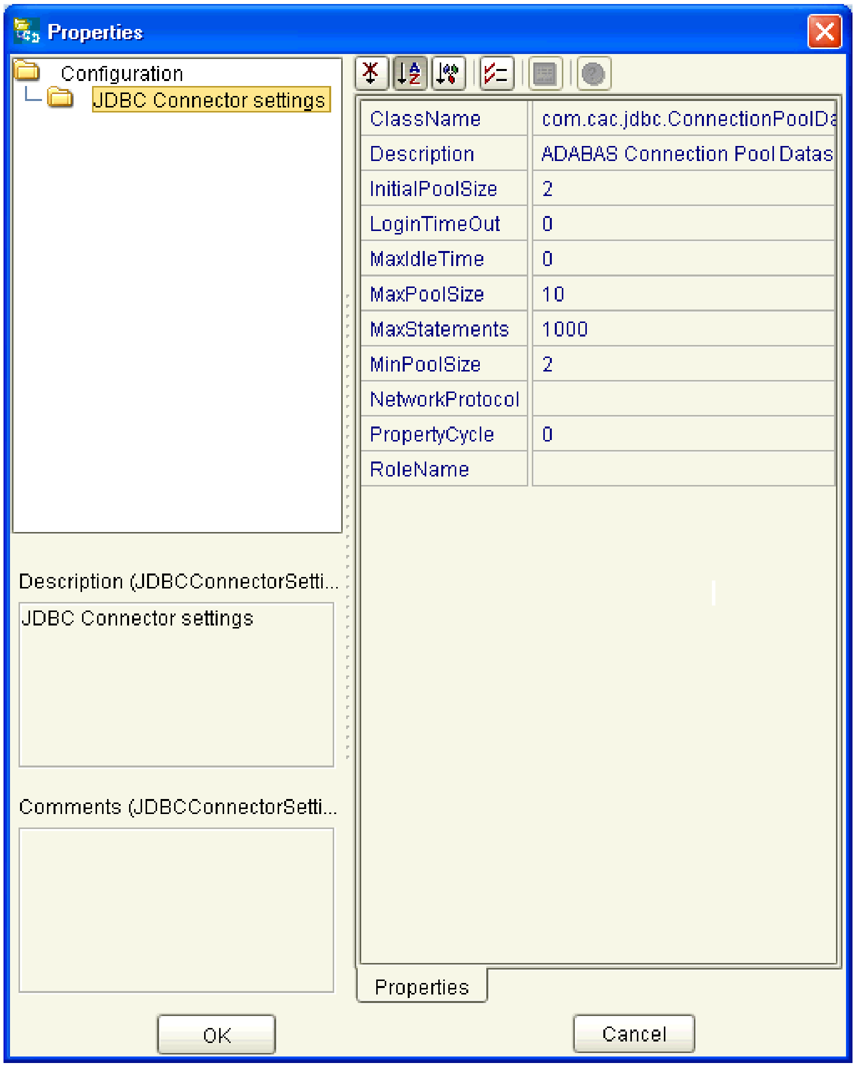
The rest of this chapter explains all of the eWay's properties in detail, under the following sections:

- **"Setting eWay Properties on the Connectivity Map" on page 14**
- **"Setting Properties on the Project Environment" on page 17**

3.2 Setting eWay Properties on the Connectivity Map

This section explains in detail the eWay's editable properties accessible on the Connectivity Map via the eGate Enterprise Designer's **Project Explorer**. You can set these properties using the eWay **Properties** sheet. See Figure 2.

Figure 2 eWay Properties Sheet: Settings on the Connectivity Map



Properties
Configuration
JDBC Connector settings
Description (JDBCConnectorSetti...
Comments (JDBCConnectorSetti...
Properties
OK
Cancel

JDBC Connector settings

ClassName	com.cac.jdbc.ConnectionPoolDa
Description	ADABAS Connection Pool Datas
InitialPoolSize	2
LoginTimeOut	0
MaxIdleTime	0
MaxPoolSize	10
MaxStatements	1000
MinPoolSize	2
NetworkProtocol	
PropertyCycle	0
RoleName	

The eWay's properties settings define the properties used to interact with the external database, under the **JDBC Connector settings** folder. These settings are:

- **"ClassName" on page 15**
- **"Description" on page 15**
- **"InitialPoolSize" on page 15**
- **"LoginTimeOut" on page 16**
- **"MaxIdleTime" on page 16**
- **"MaxPoolSize" on page 16**
- **"MaxStatements" on page 16**
- **"MinPoolSize" on page 16**
- **"NetworkProtocol" on page 17**
- **"PropertyCycle" on page 17**
- **"RoleName" on page 17**

ClassName

Description

Specifies the Java class in the JDBC driver, which is used to implement the **ConnectionPoolDataSource** interface.

Required Values

A valid class name.

The default is **ADABAS.jdbc.pool.ADABASConnectionPoolDataSource**.

Description

Description

Enter a description for the current database.

Required Values

A valid string. The default is **ADABAS Connection Pool Datasource**.

InitialPoolSize

Description

Enter a number for the physical connections the pool must contain when it is created.

Required Values

A valid numeric value.

LoginTimeOut

Description

Specifies the number of seconds the driver waits before attempting to log on to the database before timing out.

Required Values

A valid numeric value.

MaxIdleTime

Description

Specifies the maximum number of seconds that a physical connection can remain unused before it is closed. A 0 (zero) indicates there is no limit.

Required Values

A valid numeric value.

MaxPoolSize

Description

Specifies the maximum number of physical connections the pool must keep available at all times. A 0 (zero) indicates there is no maximum.

Required Values

A valid numeric value.

MaxStatements

Description

Specifies the maximum total number of statements the pool must keep open. A 0 (zero) indicates the caching of statements is disabled.

Required Values

A valid numeric value.

MinPoolSize

Description

Specifies the minimum number of physical connections the pool must keep available at all times. A 0 (zero) indicates there are no physical connections in the pool, and the new connections are created as needed.

Required Values

A valid numeric value.

NetworkProtocol

Description

Specifies the network protocol used to communicate with the server.

Required Values

Any valid string.

PropertyCycle

Description

Specifies the interval, in seconds, which the pool waits before enforcing the current policy defined by the values of the other connection pool properties specified in the eWay's properties.

Required Values

A valid numeric value.

RoleName

Description

Enter an initial SQL role name.

Required Values

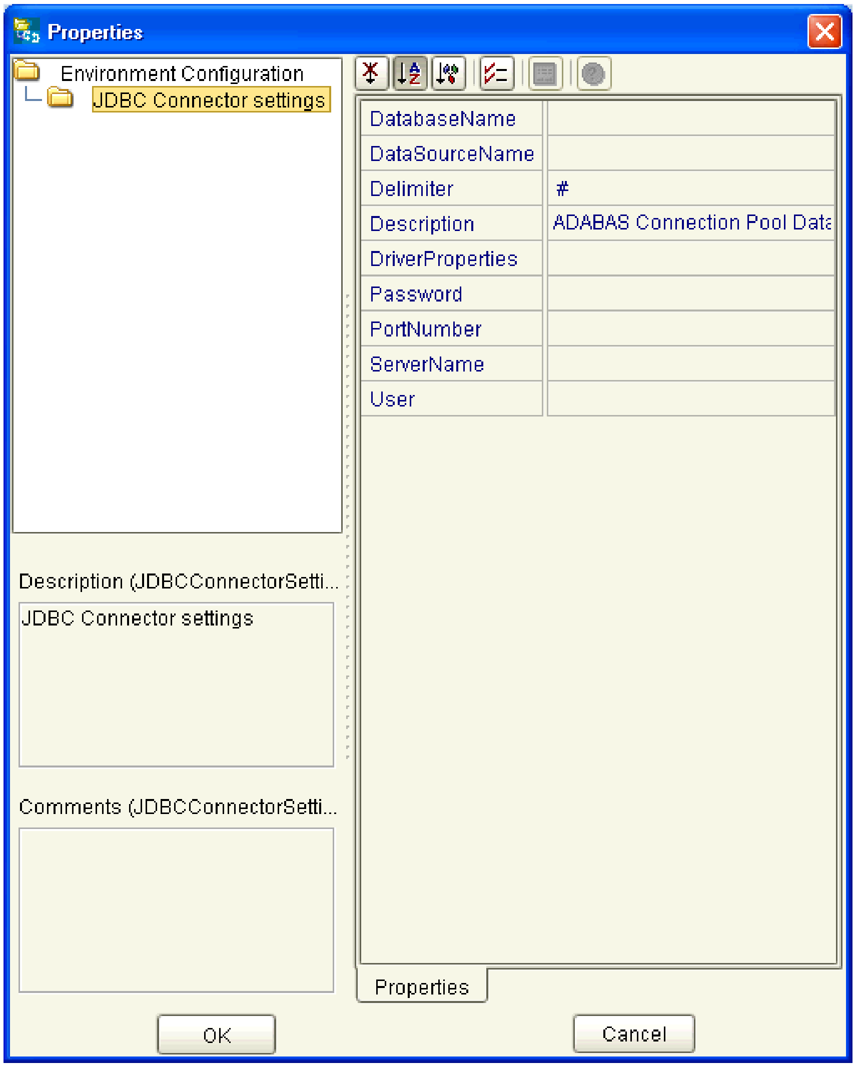
Any valid string.

3.3 Setting Properties on the Project Environment

This section explains in detail the eWay's editable properties accessible via the eGate Enterprise Designer's **Environment Explorer**. You can set these properties using the eWay **Properties** sheet.

See **Figure 3 on page 18**.

Figure 3 eWay Properties Sheet: Settings on the Project Environment



DatabaseName	
DataSourceName	
Delimiter	#
Description	ADABAS Connection Pool Data
DriverProperties	
Password	
PortNumber	
ServerName	
User	

Description (JDBCConnectorSetti...

JDBC Connector settings

Comments (JDBCConnectorSetti...

The eWay's properties, under the **JDBC Connector settings** folder, in the **Environment Explorer** are:

- **"DatabaseName" on page 19**
- **"DataSourceName" on page 19**
- **"Delimiter" on page 19**
- **"Description" on page 19**
- **"Password" on page 20**
- **"PortNumber" on page 20**
- **"ServerName" on page 20**
- **"User" on page 20**

DatabaseName

Description

Specifies the name of the database instance.

Required Values

Any valid string.

DataSourceName

Description

Provide the name of the **ConnectionPoolDataSource** object that the **DataSource** object delegates behind the scenes when connection pooling or distributed transaction management is being done.

Required Values

Optional. In most cases, leave this box empty.

Delimiter

Description

Specifies the delimiter character to be used in the **DriverProperties** prompt.

Required Value

The default is #.

Description

Description

Enter a description for the current database.

Required Values

A valid string. The default is **ADABAS Connection Pool Datasource**.

DriverProperties

Description

Allows you to enter the properties of the driver you are using; this property is optional.

You can also use this property to change some driver parameters “on the fly” such as turning on a driver trace (a description box that identifies a driver).

Required Values

Any valid delimiter and/or method as follows:

- **Delimiter**: Valid delimiters are:

```
“<method-name-1>#<param-1>#<param-2>##.........<param-n>##<method-
    name-2>#<param-1>#<param-2>#........<param-n>##......##”
```

- **Method**: For example: to execute the method **setURL()**, give the following method a string for the URL:

```
“setURL#<url>##”
```

Password

Description

Specifies the password the eWay must use to access the database.

Required Values

Any valid string.

PortNumber

Description

Specifies the input-output port number on which the server is listening for connection requests.

Required Values

A valid port number; there is no default.

ServerName

Description

Specifies the host name of the external database server.

Required Values

Any valid string.

User

Description

Specifies the user name the eWay must use to access the database.

Required Values

Any valid string.

Chapter 4

Using the ADABAS Database Wizard

This chapter describes how to use the ADABAS Database wizard to create ADABAS Object Type Definitions (OTDs) in eGate.

Chapter Topics

4.1 Overview

You can use the eGate Enterprise Designer's ADABAS Database wizard to generate an eGate OTD. For more information on Enterprise Designer features, see the *eGate Integrator User's Guide*.

The ADABAS Database wizard generates OTDs by connecting to an external ADABAS file system as if it were connecting to a database and creating corresponding OTDs. The OTD builder can then create corresponding OTDs based on any combination of tables.

Field nodes are added to the OTD based on the tables in the external data source. Java method and property nodes are added to provide the appropriate JDBC functionality. For more information on these Java methods, refer to your JDBC developer's reference.

4.2 Wizard Procedures

This section explains how to use the ADABAS Database wizard to create ADABAS OTDs.

4.2.1 Generated OTDs

The ADABAS Database wizard can create OTDs for the ADABAS eWay. These types of OTDs can also be combined with each other. The types of OTDs are:

- **Table**: Contains fields for each of the columns in the selected table, as well as the methods required to exchange data with an external data source.
- **View**: The same as a table, except for read-only purposes.
- **Prepared Statements**: Allow you to add prepared SQL statements to an OTD.

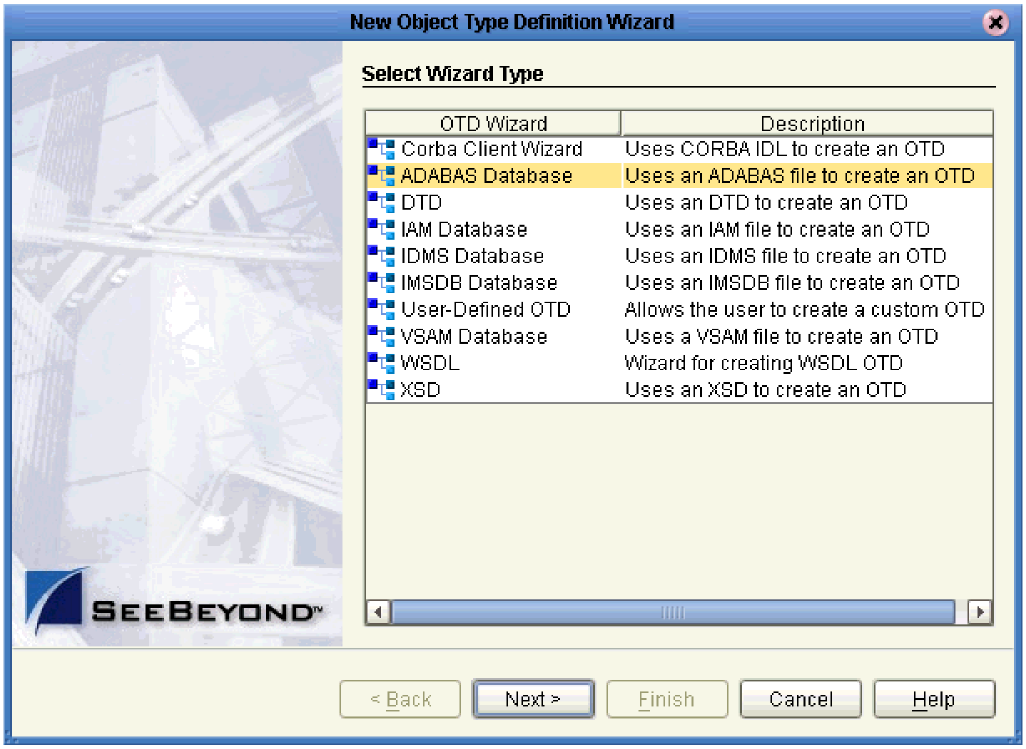
For more information on these types, see **Chapter 6**.

4.2.2 Creating an ADABAS OTD

To create an ADABAS OTD using the wizard

1. From the eGate Enterprise Designer's **Enterprise Explorer** pane (**Project** tab), right-click on the desired **Project** folder and select **New** then **Create an Object Type Definition** from the pop-up menu.
2. The **OTD Wizard Selection** window appears, displaying all installed OTD wizards as shown in Figure 4.

Figure 4 OTD Wizard Selection Window

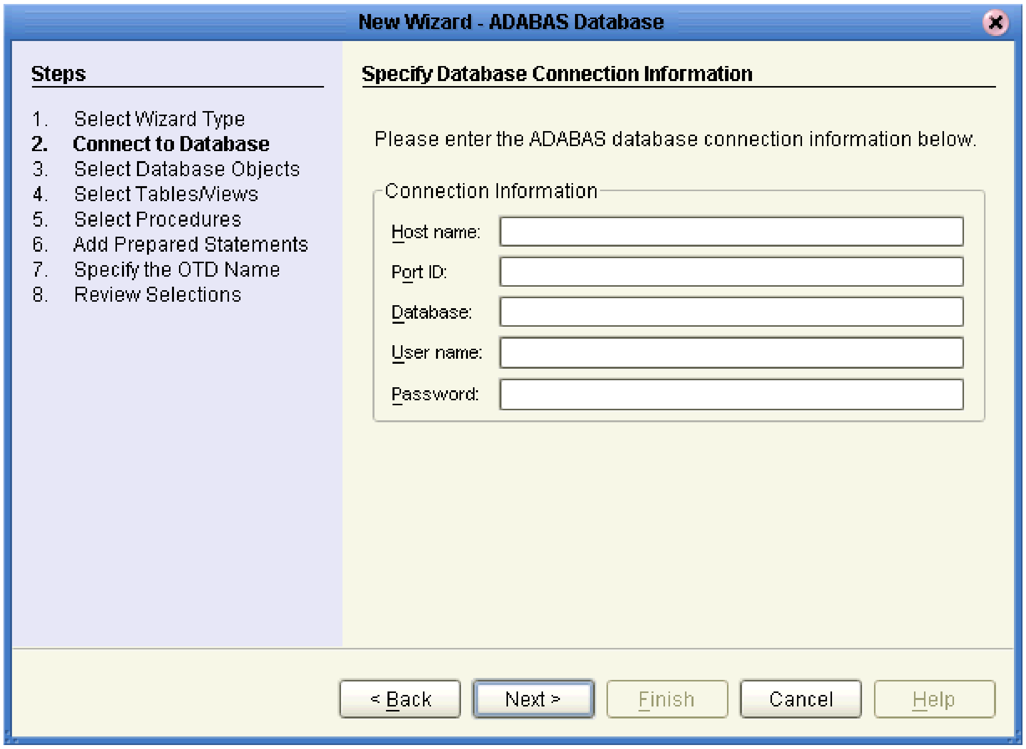


OTD Wizard	Description
Corba Client Wizard	Uses CORBA IDL to create an OTD
ADABAS Database	Uses an ADABAS file to create an OTD
DTD	Uses an DTD to create an OTD
IAM Database	Uses an IAM file to create an OTD
IDMS Database	Uses an IDMS file to create an OTD
IMSDB Database	Uses an IMSDB file to create an OTD
User-Defined OTD	Allows the user to create a custom OTD
VSAM Database	Uses a VSAM file to create an OTD
WSDL	Wizard for creating WSDL OTD
XSD	Uses an XSD to create an OTD

3. From the list, select the **ADABAS Database** OTD. Click **Next**.

   The **Specify Database Connection Information** window appears. See **Figure 5 on page 23**.

Figure 5 Specify Database Connection Information Window



New Wizard - ADABAS Database

Steps

1. Select Wizard Type
2. **Connect to Database**
3. Select Database Objects
4. Select Tables/Views
5. Select Procedures
6. Add Prepared Statements
7. Specify the OTD Name
8. Review Selections

Specify Database Connection Information

Please enter the ADABAS database connection information below.

Connection Information

Host name:

Port ID:

Database:

User name:

Password:

4 Specify the connection information for the current database as follows:

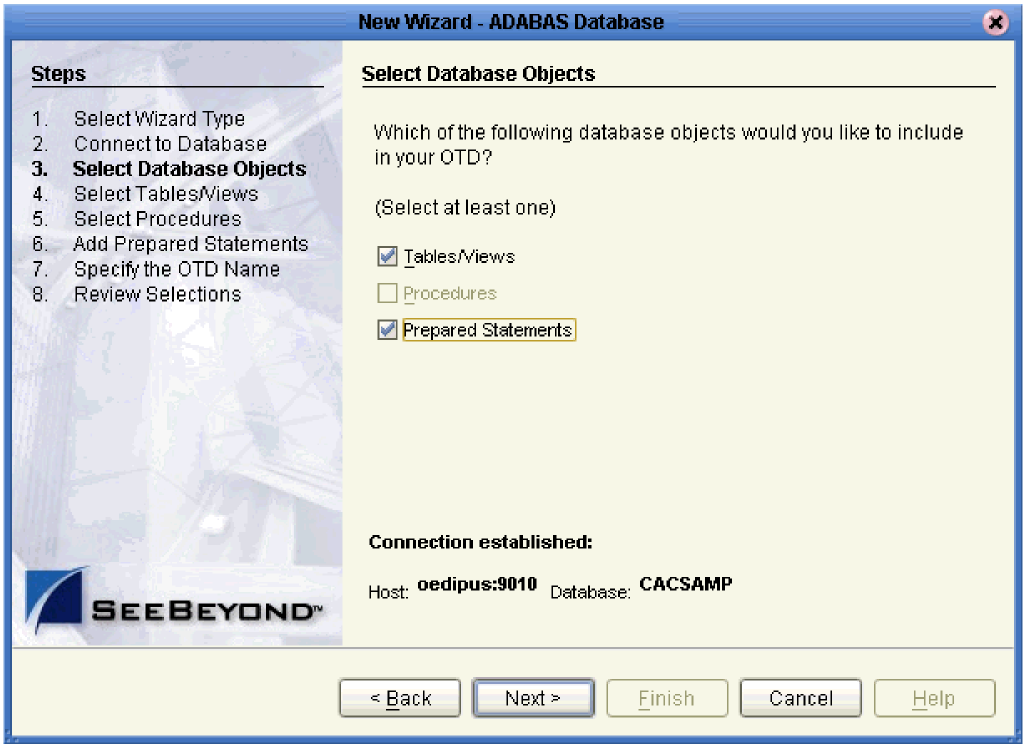
- **Host name**: The name of the host you are connecting to.
- **Port ID**: The host port number.
- **Database**: The name of the database you are connecting to.
- **User name**: Your user ID.
- **Password**: Your password.

5 Click **Next**.

6 When the **Connection Status** line says you are connected to the database, click **Next** to continue.

7 In the **Select Database Objects** window, select **Tables/Views**. Click **Next** to continue. See Figure 6.

Figure 6 Select Database Objects Window



Select Database Objects

Which of the following database objects would you like to include in your OTD?

(Select at least one)

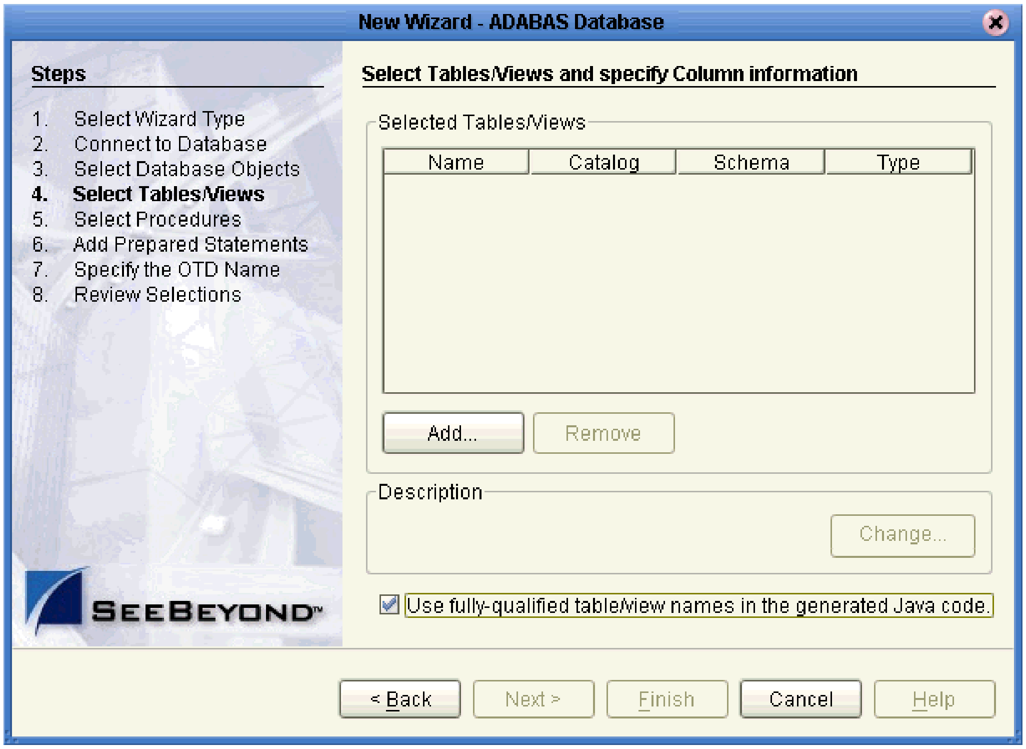
☑ Tables/Views
□ Procedures
☑ Prepared Statements

Connection established:

Host: oedipus:9010 Database: CACSAMP

The **Select Tables/Views** window appears. See **Figure 7 on page 25**.

Figure 7 Select Tables/Views Window



New Wizard - ADABAS Database

Steps

1. Select Wizard Type
2. Connect to Database
3. Select Database Objects
4. **Select Tables/Views**
5. Select Procedures
6. Add Prepared Statements
7. Specify the OTD Name
8. Review Selections

Select Tables/Views and specify Column information

Selected Tables/Views

Name	Catalog	Schema	Type

☑ Use fully-qualified table/view names in the generated Java code.

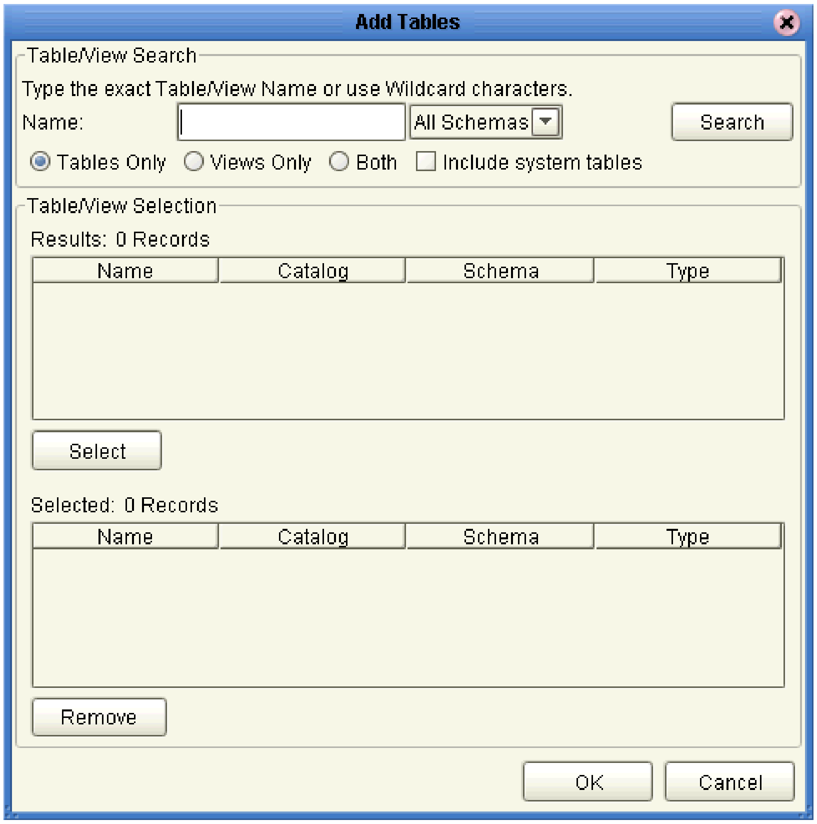
Note: *You must select the* ***Use fully-qualified table/view names in the generated Java code*** *check box.*

To select tables and views

1. Click **Add**.

   The **Add Tables** window appears. See **Figure 8 on page 26**.

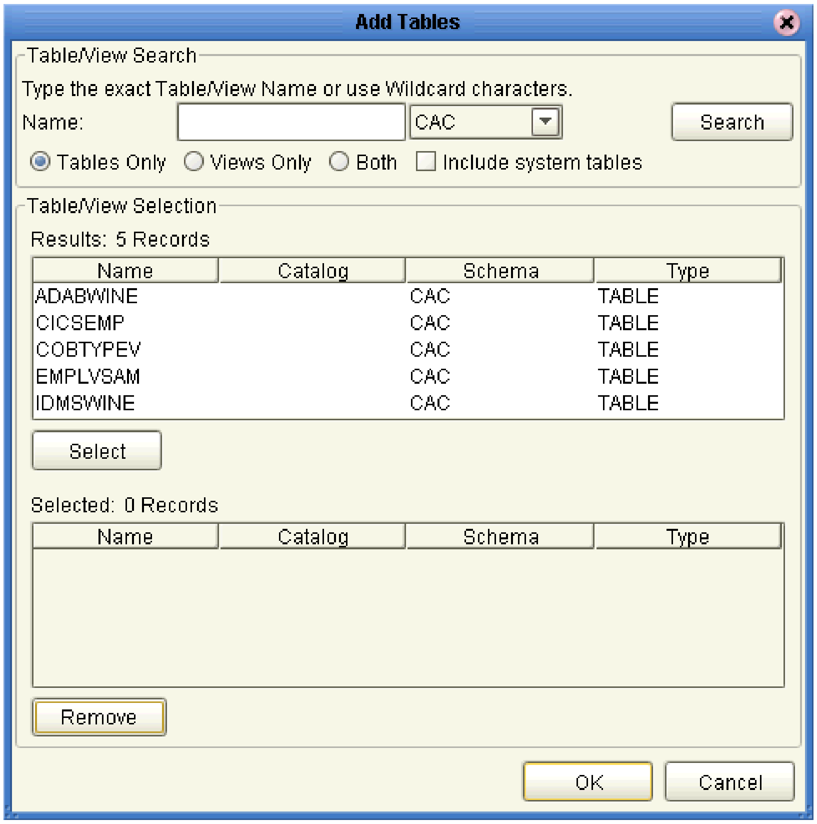
Figure 8 Add Tables Window: Beginning



Note: *Views are read-only and for informational purposes only.*

2 Select the schema you want to search in the **All Schemas** drop-down list and click **Search**. See **Figure 9 on page 27**.

Figure 9 Add Tables Window: With Schemas



Type the exact Table/View Name or use Wildcard characters.

Name: CAC

● Tables Only ○ Views Only ○ Both □ Include system tables

Results: 5 Records

Name	Catalog	Schema	Type
ADABWINE		CAC	TABLE
CICSEMP		CAC	TABLE
COBTYPEV		CAC	TABLE
EMPLVSAM		CAC	TABLE
IDMSWINE		CAC	TABLE

Selected: 0 Records

Name	Catalog	Schema	Type

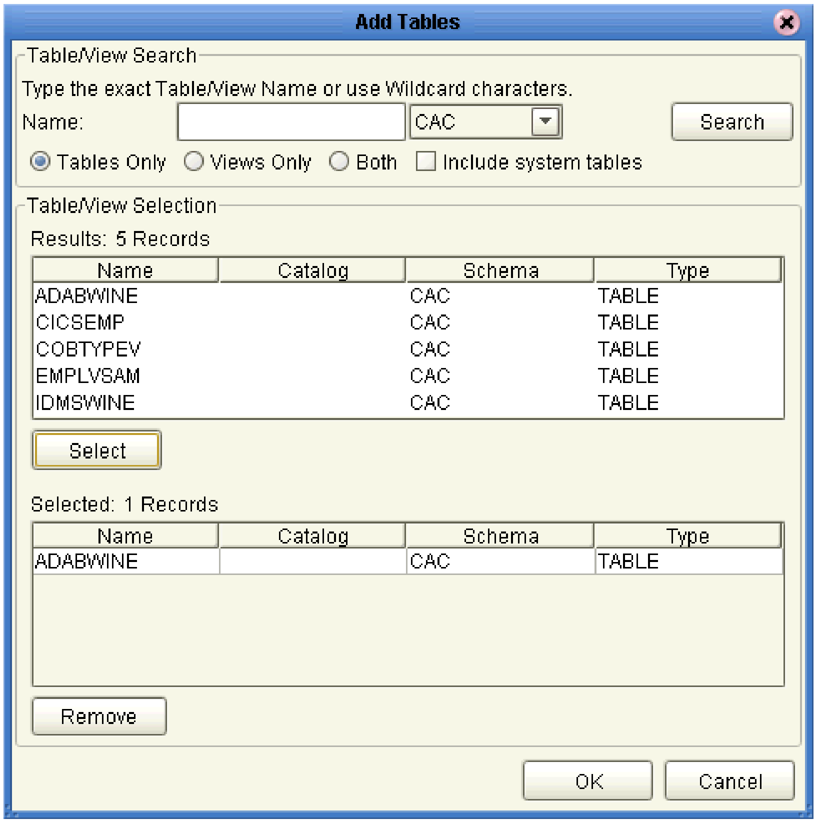
3 Select the desired table and click **Select**. You can also use adjacent or nonadjacent selections to select multiple table names.

Every name of the table or tables you have selected appear in the lower pane. See **Figure 10 on page 28**.

You can click **Search** to find the desired table or tables.

You can use wildcard characters to search for a table or view. If desired, use the eWay wildcard characters "?", "_", and "*". For example, you can use "AB?CD", "AB_CD," or "AB*CD". However, do not use "%". Using this character results in nothing being returned.

Figure 10 Add Tables Window: With Added Table



Table/View Search

Type the exact Table/View Name or use Wildcard characters.

Name: CAC

● Tables Only ○ Views Only ○ Both □ Include system tables

Table/View Selection

Results: 5 Records

Name	Catalog	Schema	Type
ADABWINE		CAC	TABLE
CICSEMP		CAC	TABLE
COBTYPEV		CAC	TABLE
EMPLVSAM		CAC	TABLE
IDMSWINE		CAC	TABLE

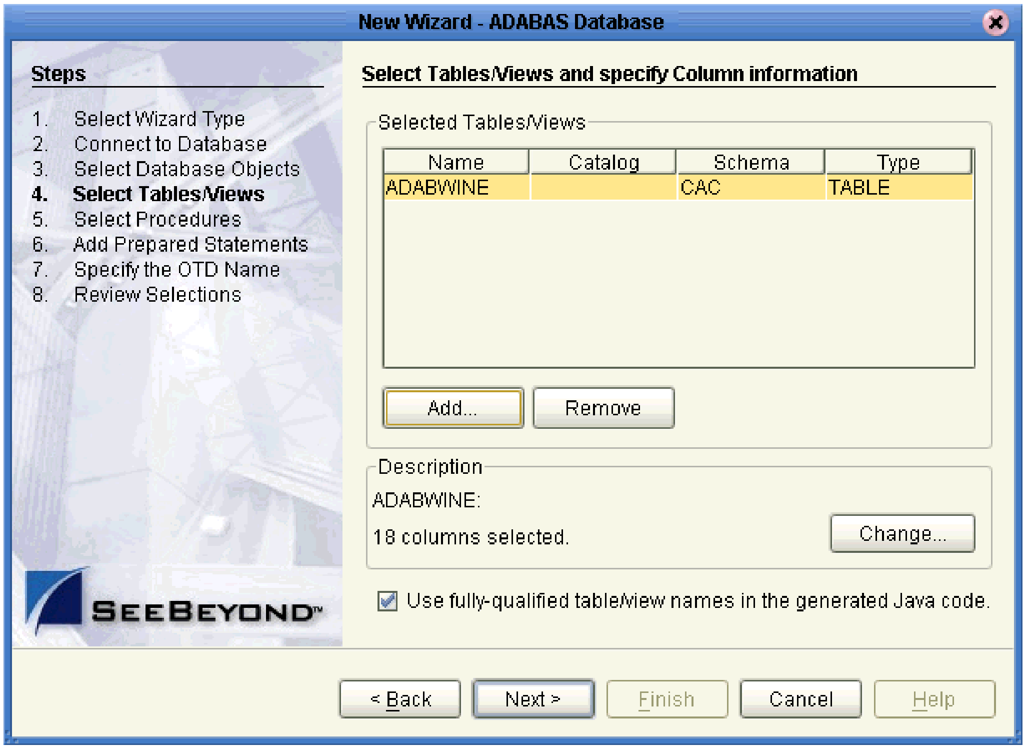
Selected: 1 Records

Name	Catalog	Schema	Type
ADABWINE		CAC	TABLE

4 When you are finished, click **OK**.

The **Select Tables/Views** window appears again, showing your one or more selected tables. See Figure 11 on page 29. The figure shows one table selected, **EMPLOYEES**.

Figure 11 Select Tables/Views Window With Table



New Wizard - ADABAS Database

Steps

1. Select Wizard Type
2. Connect to Database
3. Select Database Objects
4. **Select Tables/Views**
5. Select Procedures
6. Add Prepared Statements
7. Specify the OTD Name
8. Review Selections

Select Tables/Views and specify Column information

Selected Tables/Views

Name	Catalog	Schema	Type
ADABWINE		CAC	TABLE

Description

ADABWINE:

18 columns selected.

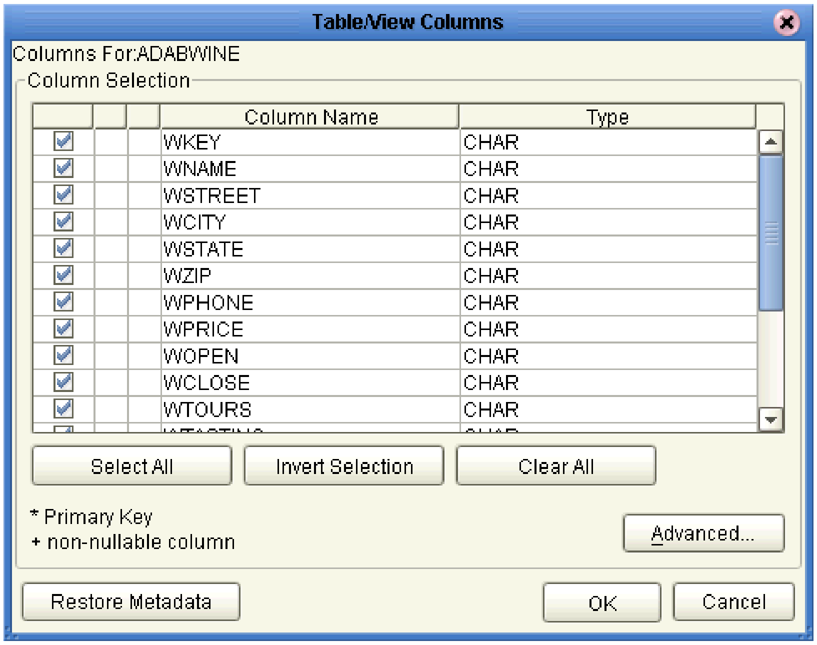
☑ Use fully-qualified table/view names in the generated Java code.

5 In the **Select Tables/Views** window, review the table or tables you have selected. You can make changes to the table columns, if desired, by clicking **Change** to open the **Table/View Columns** window.

See **"Using Tables" on page 51** for more information on how to use tables with the OTD.

6 If you click **Change** on the **Select Tables/Views** window, the **Table/View Columns** window appears. See **Figure 12 on page 30**.

Figure 12 Table/View Columns Window



			Column Name	Type
☑			WKEY	CHAR
☑			WNAME	CHAR
☑			WSTREET	CHAR
☑			WCITY	CHAR
☑			WSTATE	CHAR
☑			WZIP	CHAR
☑			WPHONE	CHAR
☑			WPRICE	CHAR
☑			WOPEN	CHAR
☑			WCLOSE	CHAR
☑			WTOURS	CHAR

7 In the **Table/View Columns** window, you can select or deselect any table columns. You can also change the data type for each column by highlighting the data type and selecting a different data type from the drop-down list.

The buttons in this window operate as follows:

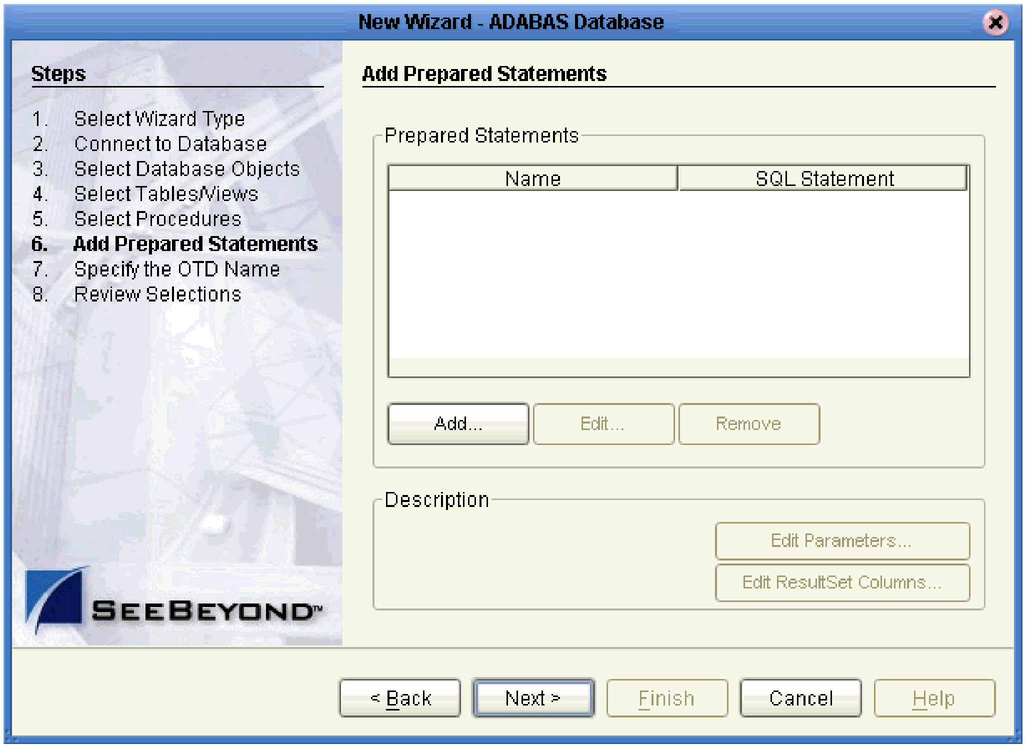
- **Select All**: Allows you to select all columns for inclusion in the OTD.
- **Invert Selection**: Allows you to invert the order of the selected columns.
- **Clear All**: Allows you to de-select all columns.
- **Advanced**: Opens a dialog box allowing you to change the current column properties of every column. You can edit the column's **SQL Type**, **Precision** or **Length**, and **Scale**.
- **Restore Metadata**: Allows you to restore the data to its original database state before you made any changes via the wizard.

When you are finished using this window, click **OK** to save your changes and return to the **Select Tables/Views** window.

8 When you are finished with the **Select Tables/Views** window (see **Figure 11 on page 29**), click **Next** to continue.

The **Add Prepared Statements** window appears. See **Figure 13 on page 31**.

Figure 13 Add Prepared Statements Window



New Wizard - ADABAS Database

Steps

1. Select Wizard Type
2. Connect to Database
3. Select Database Objects
4. Select Tables/Views
5. Select Procedures
6. **Add Prepared Statements**
7. Specify the OTD Name
8. Review Selections

Add Prepared Statements

Prepared Statements

Name	SQL Statement

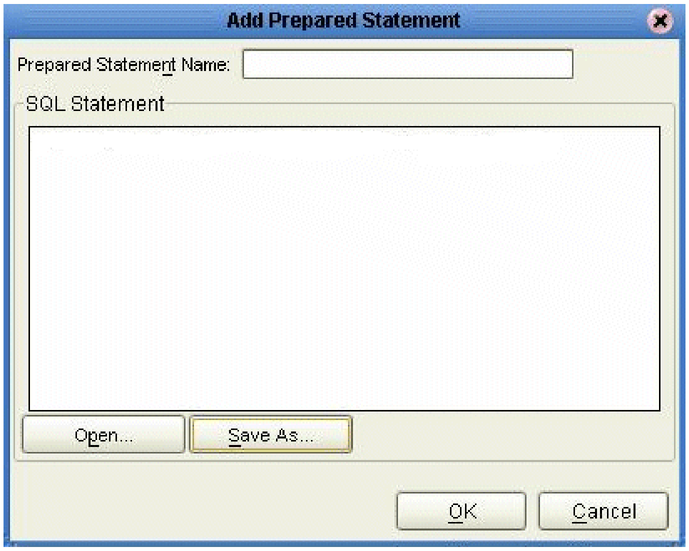
If you do not want to add any prepared statements, click **Next** to go to the next window in the wizard.

To add prepared statements

1 If you do want to add a prepared statement, click **Add** on the **Add Prepared Statements** window.

The **Add Prepared Statement** dialog box appears. See Figure 14.

Figure 14 Add Prepared Statement Dialog Box



2 Enter a name for your prepared statement.

3 Create an SQL statement by clicking in the **SQL Statement** pane and entering your statement, for example:

```
Select * from ADABWINE where WNAME like ?
```

4 If desired, when you are finished creating the statement, click **Save As** and give the statement a name. You can also click **Open** to access an existing prepared statement.

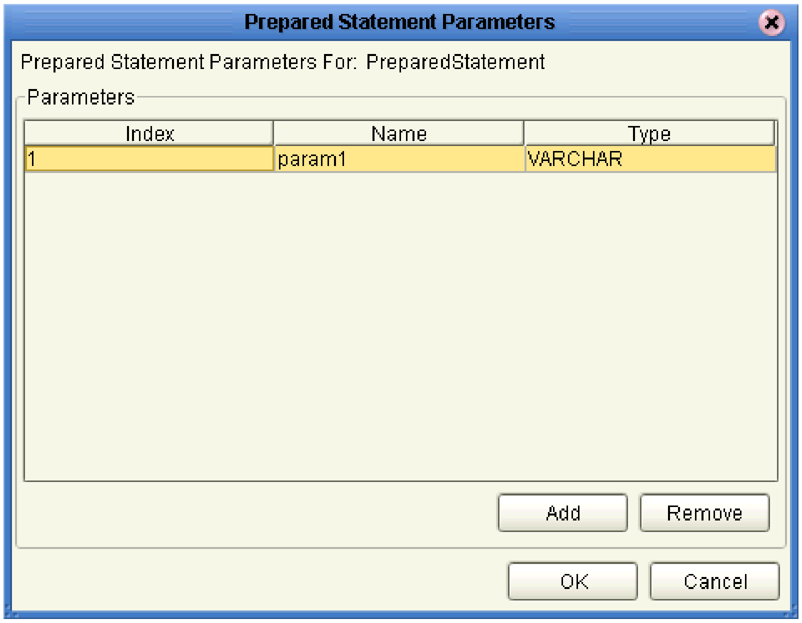
5 When you are finished, click **OK**.

The **Add Prepared Statements** window displays the name of your new prepared statement in the list pane.

6 To edit the parameters, click **Edit Parameters**. Using the resulting dialog box, you can change the data type by clicking the **Type** field and selecting a different type from the list.

7 Click **Add** if you want to add additional parameters to the statement, or highlight a row and click **Remove** to remove it then click **OK**. See Figure 15.

Figure 15 Edit Prepared Statement Parameters



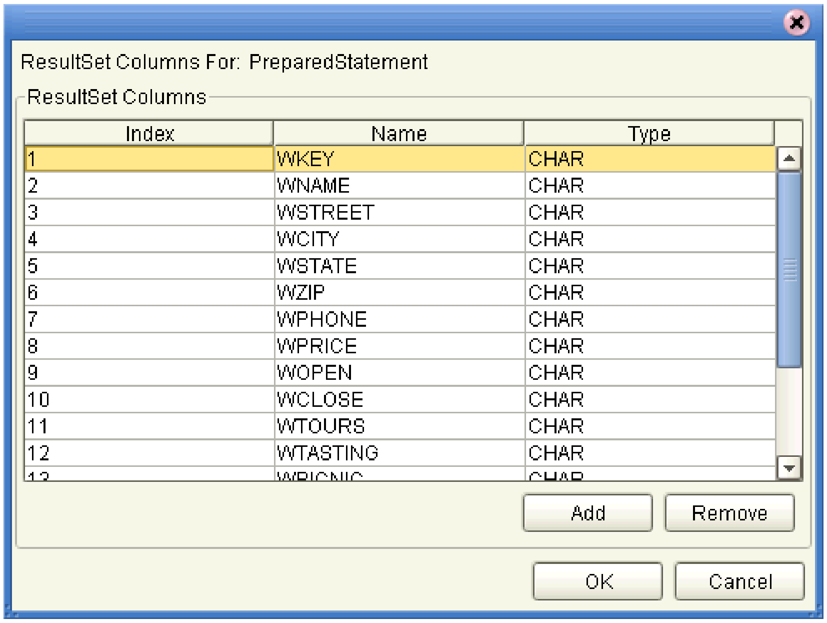
Prepared Statement Parameters For: PreparedStatement

Parameters

Index	Name	Type
1	param1	VARCHAR

8 To edit the result set columns, click **Edit ResultSet Columns**. In the resulting dialog box, both the **Name** and **Type** are editable, but it is recommend that you do not change the name. Doing so can cause a loss of integrity between the result set and the database. See Figure 16.

Figure 16 ResultSet Columns



ResultSet Columns For: PreparedStatement

ResultSet Columns

Index	Name	Type
1	WKEY	CHAR
2	WNAME	CHAR
3	WSTREET	CHAR
4	WCITY	CHAR
5	WSTATE	CHAR
6	WZIP	CHAR
7	WPHONE	CHAR
8	WPRICE	CHAR
9	WOPEN	CHAR
10	WCLOSE	CHAR
11	WTOURS	CHAR
12	WTASTING	CHAR
13	WPICNIC	CHAR

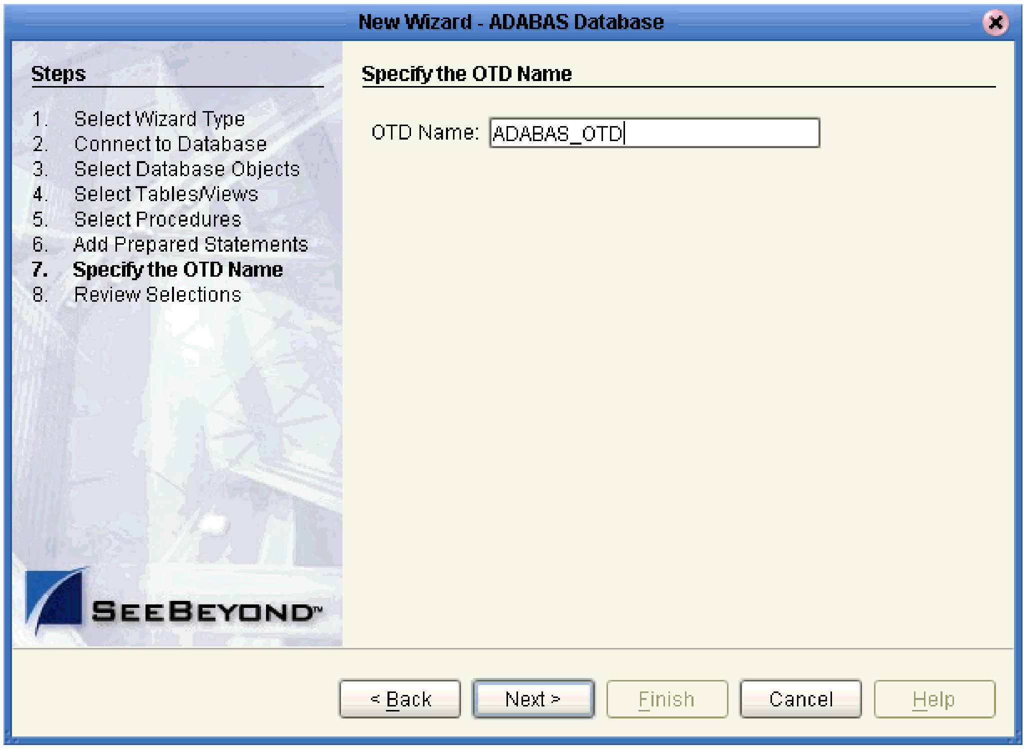
9 Whey you are finished with the dialog box, click **OK**.

10 On the **Add Prepared Statements** window, click **OK**.

11 When you are finished with the **Add Prepared Statements** window (see **Figure 13 on page 31**), click **Next** to continue.

The **Specify the OTD Name** window appears. See **Figure 17 on page 34**.

Figure 17 Specify the OTD Name Window



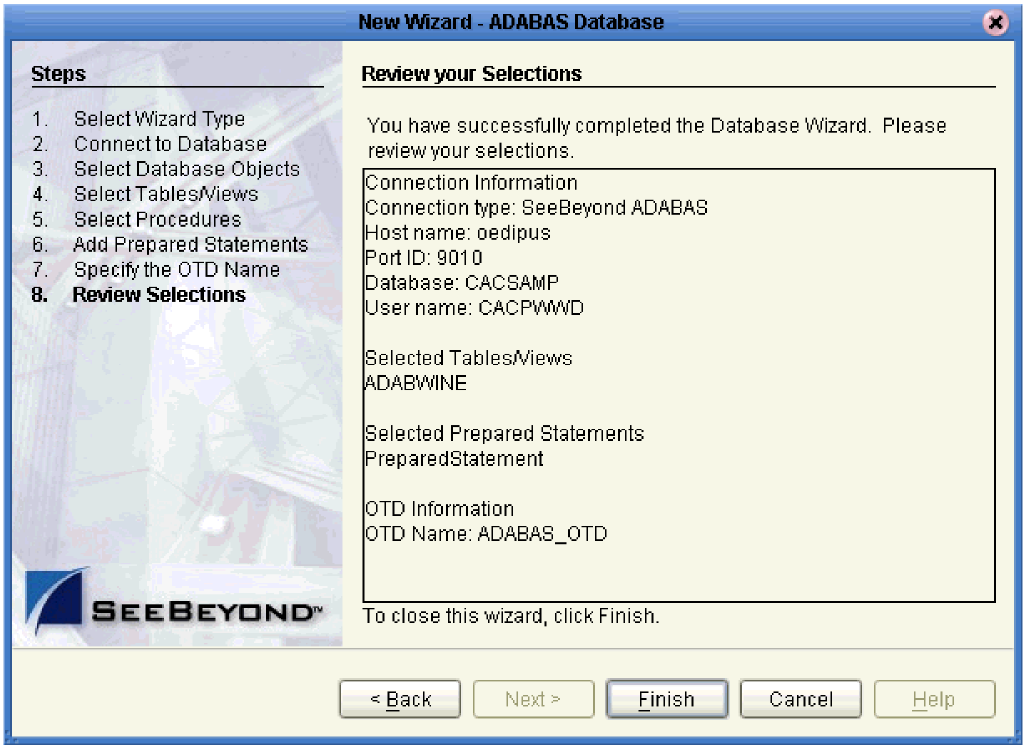
OTD Name: ADABAS_OTD

To create the OTD

1 Enter a name you want to specify for the current OTD, for example, **ADABAS_OTD**, and click **Next**.

The **Review your Selections** window appears. See **Figure 18 on page 35**.

Figure 18 Review Your Selections Window



Review your Selections

You have successfully completed the Database Wizard. Please review your selections.

Connection Information
Connection type: SeeBeyond ADABAS
Host name: oedipus
Port ID: 9010
Database: CACSAMP
User name: CACPWWD

Selected Tables/Views
ADABWINE

Selected Prepared Statements
PreparedStatement

OTD Information
OTD Name: ADABAS_OTD

To close this wizard, click Finish.

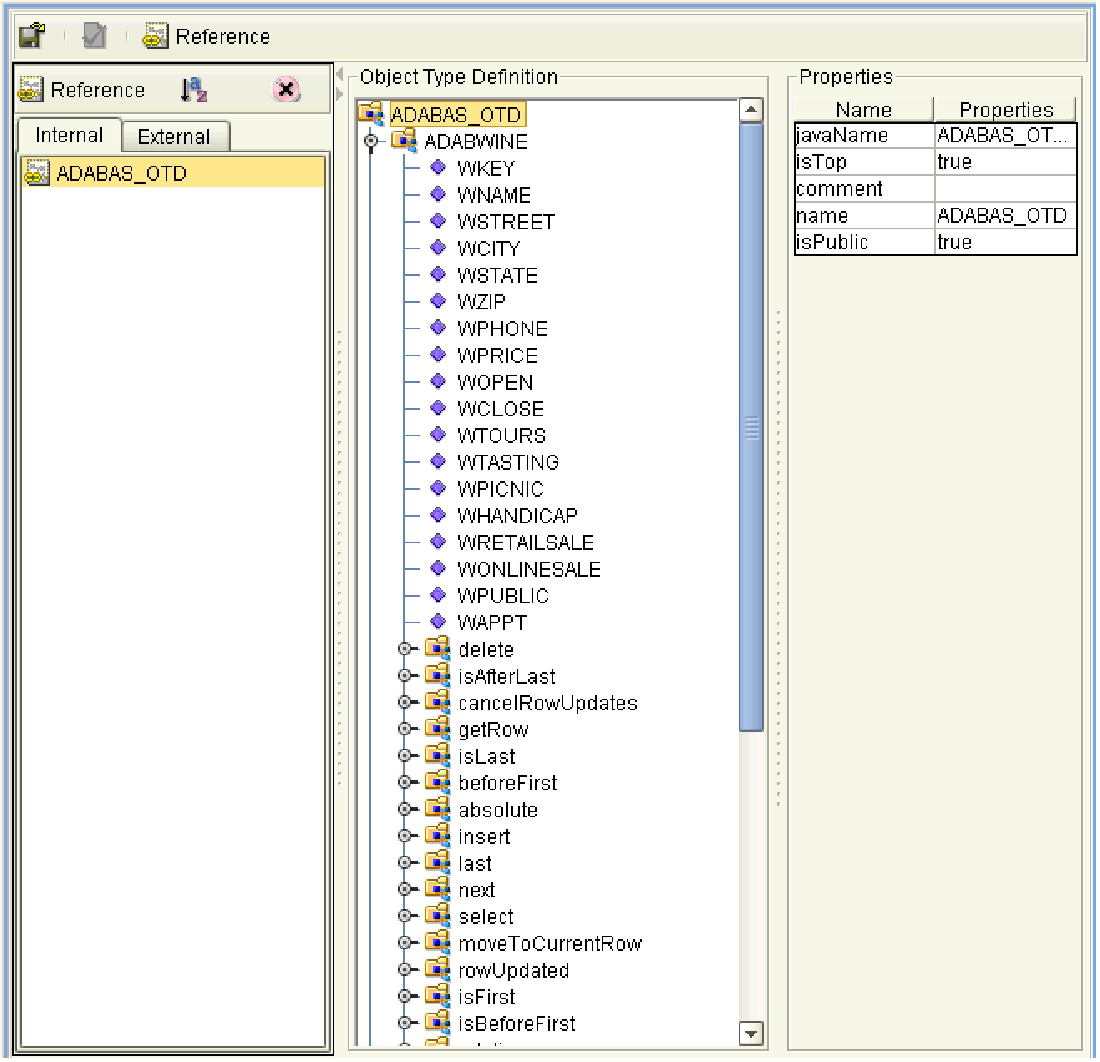
2 Review the information you have entered (see Figure 18). If any information is incorrect, click **Back** to return to the desired window and correct it. If all the information is correct, click **Finish** to create your OTD.

4.3 Resulting OTD

The resulting ADABAS OTD appears on the eGate Enterprise Designer's canvas, in the OTD Editor window. See **Figure 19 on page 36**.

In the example, a node in the OTD has been expanded, allowing you to view some of its structure.

Figure 19 OTD Editor in Enterprise Designer



Name	Properties
javaName	ADABAS_OT...
isTop	true
comment	
name	ADABAS_OTD
isPublic	true

Chapter 5

Reviewing the Sample Projects

This chapter describes how to implement the ADABAS eWay via a review of the sample Projects included with the eWay. The following Projects are included:

- Sample that employs SeeBeyond ICAN Suite's eInsight Business Process Manager and operates using eInsight Business Processes
- Sample that operates using Collaborations (Java) in eGate

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

For a complete explanation of eGate/eInsight terminology and concepts, see the *eGate Integrator User's Guide* and the *eInsight Business Process Manager User's Guide.* 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 Tutorial*.

Chapter Topics

5.1 eGate Project Description

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

5.1.1 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.
- **Business Process Canvas**: Allows you to use eInsight's Business Process management features.
- **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.

5.1.2 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 **ADABAS_eWay_Sample.zip**. The Project file names are:

- **ADABAS_Project_BPEL.zip**: eInsight Project
- **ADABAS_Project_JCE.zip**: Collaboration (Java) Project

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 48** *and* **"Deploying a Project" on page 49**. *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. For details, see the *eGate Tutorial*.

5.1.3 Basic eWay Components

Each Project allows you to set up eWays to perform the necessary operations. To use the ADABAS eWay in a Project, you must:

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

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

ADABAS eWay Properties

The properties for the ADABAS eWay contains the settings used to connect with a specific external system. You can change these settings using the eGate eWay **Properties** sheet. For more information about ADABAS eWay properties and this dialog box, see **Chapter 3**.

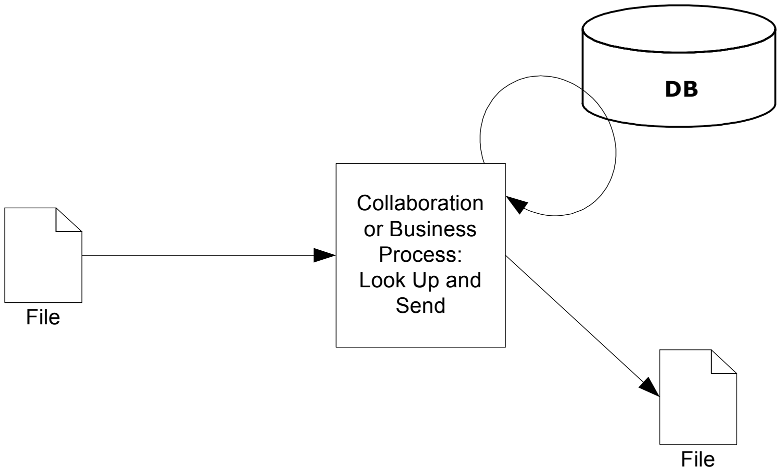
ADABAS OTDs

You create the ADABAS OTD using the ADABAS OTD wizard feature. See **Chapter 4** for details on how to use this wizard.

5.1.4 Overview: Sample Projects

The ADABAS eWay sample Projects demonstrate how the ADABAS eWay processes information from an ADABAS system. The resulting information is then written to a text file. Figure 20 shows a general diagram of how this scenario operates.

Figure 20 Sample Project Scenario



Project Components

Each Project has the following components:

- File external application (inbound): **FileIn**
- Inbound File eWay
- Business logic implementation employs one of the following components:
  - eInsight Business Process for processing data: **BP_Service**
  - Collaboration (Java) for processing data: **Collab_Service**
- ADABAS eWay
- External ADABAS database: **ADABAS1**
- Outbound File eWay
- File external application (outbound): **FileOut**

Project Operation

Each Project operates as follows:

- **FileIn**: The external file system that provides query and selection instructions to the inbound File eWay; this eWay gets a text file containing the instructions and passes them to a Business Process, **BP_Service**.
- Business logic implementation employs one of the following components:
  - **BP_Service**: Sends instructions, using a **SelectOne** Business Process operation, to the desired ADABAS database via the ADABAS eWay. **BP_Service** also receives the information looked up, via the ADABAS eWay, then sends it to a File eWay, **FileOut**.
  - **Collab_Service**: Sends instructions, using a **select** method, to the desired ADABAS database via the ADABAS eWay. **Collab_Service** also receives the information looked up, via the ADABAS eWay then sends it to a File eWay, **FileOut**.
- **ADABAS1**: The ADABAS database system; the ADABAS eWay handles inbound and outbound communication with this database.
- **FileOut**: The external file system that receives the query information; another File eWay writes the database information to a text file on this system.

Input and Output Data

The sample Project has the following input and output:

- **Input**: Use the file, **inputadabas.txt**, provided with the sample.
- **Output**: The information, in a text file, selected from the database.

5.2 Summary: Sample Business Process Project

This section explains how to implement the ADABAS eWay using the eGate Project sample with an eInsight Business Process. This sample is included on your installation CD-ROM. The sample is named **ADABAS__Project_BPEL**. It allows you to observe an end-to-end data-exchange scenario involving eGate and the ADABAS eWay.

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

5.2.1 Using the eWay With eInsight

You can set up and deploy an eGate component using eInsight. Once you have associated the desired component with a Business Process, the eInsight engine can automatically invoke that component during run time, using a Business Process Execution Language (BPEL) interface.

Note: *You must have the **eInsight.sar** file installed to use the features explained in this chapter. See the **SeeBeyond ICAN Suite Installation Guide** for complete installation procedures.*

Using eInsight With eGate Components

Examples of eGate components that can interface with eInsight in this way are:

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

Using the eGate Enterprise Designer and its eInsight canvas, you can add a desired operation to a Business Process, then associate that process with an eGate component, for example, a Service. In the Enterprise Designer, associate the Business Process and Service icons using drag-and-drop operations.

See the *eInsight Business Process Manager User's Guide* for details.

ADABAS eWay With eInsight

You can add ADABAS eWay objects to an eInsight Business Process during the system design phase. To make this association, select the desired operation, for example **Insert** or **Update,** under the eWay in the Enterprise Explorer and drag it onto the eInsight Business Process canvas. In turn, you can activate a Business Process in eGate by dragging it onto a Service or onto the Business Process canvas.

At run time, the eInsight Engine is able to invoke each of the steps in order as set up in the Business Process. Using the engine's BPEL interface, eInsight in turn invokes the ADABAS eWay operations, as well as any other eWays in the current Project.

Table 1 shows the eInsight Business Process operations available to the ADABAS eWay, as well as the inputs and outputs to each of these operations.

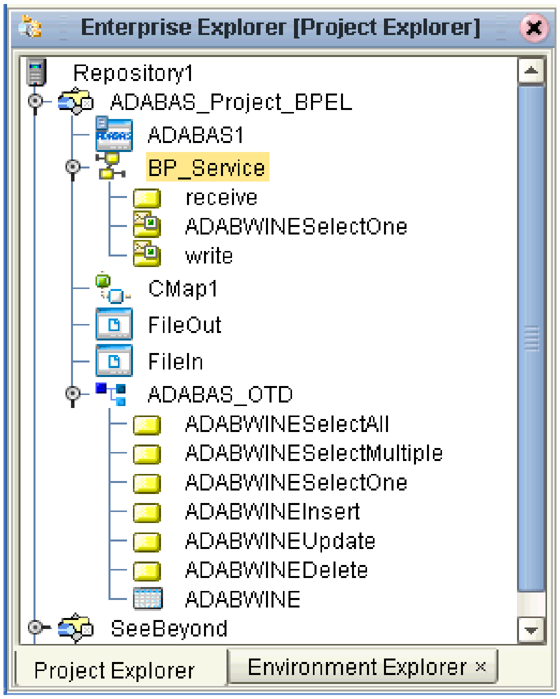
Table 1 Available eInsight ADABAS Business Process Operations

eInsight Business Process Operation	Input	Output
SelectAll	where() clause (optional)	Returns all rows that fit the condition of the where() clause.
SelectMultiple	Number of rows where() clause (optional)	Returns the number of rows specified that fit the condition of the where() clause.
SelectOne	where() clause (optional)	Returns the first row that fits the condition of the where() clause.
Insert	Definition of new item to be inserted	Returns status of the insert operation.
Update	where() clause	Returns status of the update operation.
Delete	where() clause	Returns status of the delete operation.

Note: *The **where** clause can be left blank for **SelectAll**, **SelectMultiple**, and **SelectOne**. If this clause is imbedded in an input file, the file can contain nothing (empty). If the clause is imbedded in a **Literal**, it can be a blank **Literal**.*

Figure 21 shows the Enterprise Designer's **Project Explorer** with all the ADABAS eWay Business Process operations exposed under the OTD icon.

Figure 21 Project Explorer With Business Process Operations



Before you can make these Business Process operations available, you must first create an ADABAS OTD using the Database wizard. See **Chapter 4** for details on this operation.

5.2.2 Using a Business Process

Once you have designed your Business Process for this sample, you can use the eGate Enterprise Designer to create it. See **"Overview: Sample Projects" on page 40** for a description of the sample's Business Process.

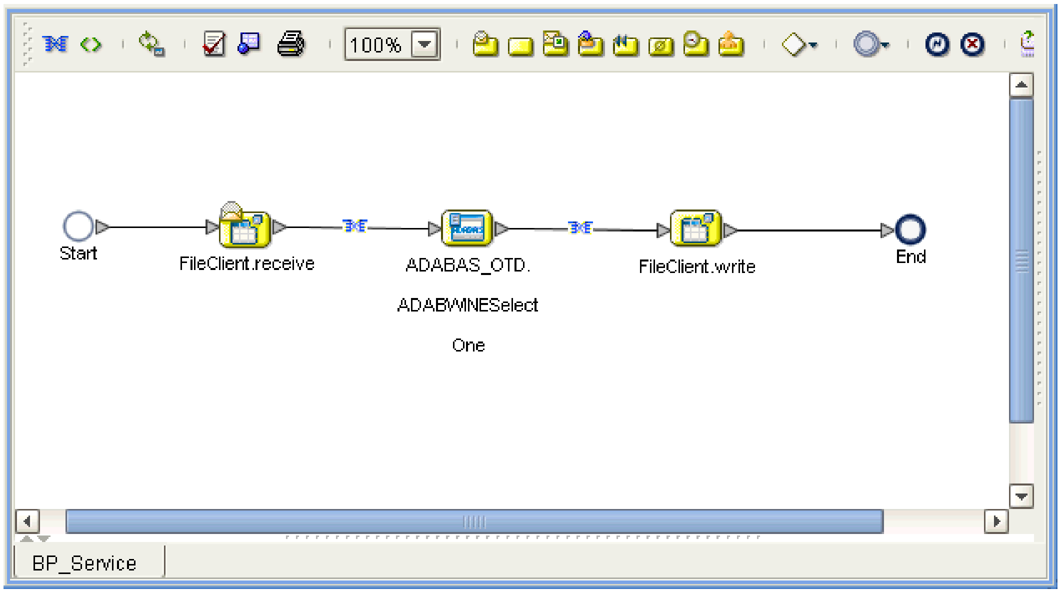
The eInsight sample Project Business Process includes the following steps:

1. An inbound File eWay subscribes to an external directory, and picks up a text request for specific data. The sample Project uses the text file **inputadabas.txt** to acquire data from an ADABAS database.
2. The File eWay sends the data to an eInsight Activity, which then sends a request to the ADABAS eWay to get and return the requested data from the ADABAS database.

3 The eInsight Activity converts the related data received from the ADABAS database to a text file and then uses an outbound File eWay to publish this information an external file.

Figure 22 demonstrates how these operations appear on the sample Project's Business Process canvas.

Figure 22 Sample Project Business Process



5.2.3 Configuring the Modeling Elements

Business Rules are defined and configured between the Business Process Activities located on the modeling canvas. The sample Project contains the Business Rules that govern each of the Activities listed in a Business Process flow.

Business Rules found in the sample Project include:

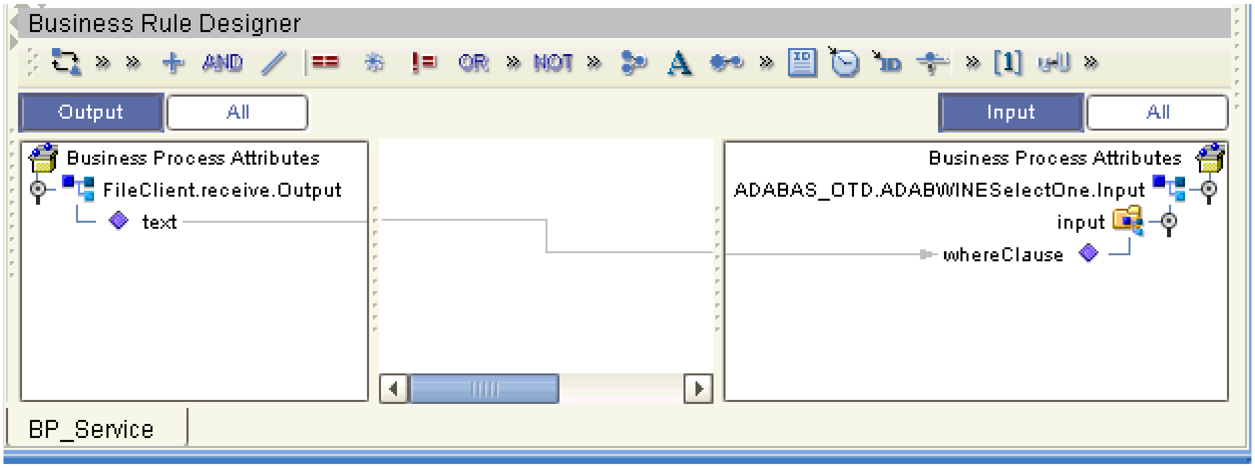
- **"Copying the Output File" on page 44**
- **"Returning the Value" on page 45**

Note: *A detailed description of the steps required to configure modeling elements is found in the* ***eInsight Business Process Manager User's Guide****.*

Copying the Output File

The **FileClient.receive.Output** container copies the output file containing the text requests a specific name and queries a **whereClause** in the OTD input container. See **Figure 23 on page 45**.

Figure 23 Copying the Output File

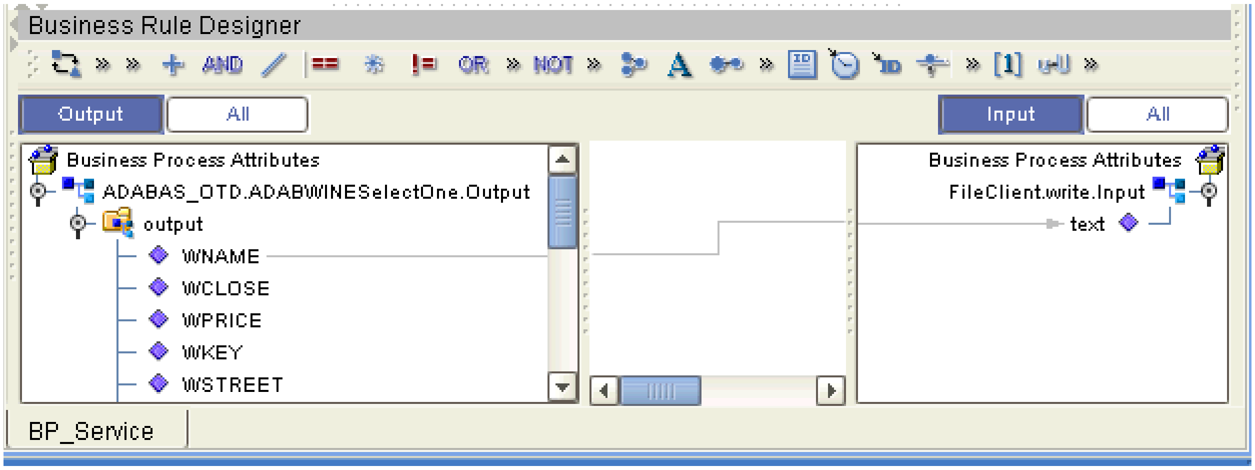


Business Rule Designer
AND
OR
NOT
Output
All
Input
All
Business Process Attributes
FileClient.receive.Output
text
Business Process Attributes
ADABAS_OTD.ADABWINESelectOne.Input
input
whereClause
BP_Service

Returning the Value

The OTD output container queries the ADABAS database for all text values of the type **WNAME** that fit the requested name, then writes the resulting value to a text file using the **FileClient.write.Input** container. See Figure 24.

Figure 24 Returning the Requested Value



5.2.4 Completing the Project

For instructions on how to finish implementing your Project, see the following sections:

- **“Creating the Project’s Environment” on page 48**
- **“Setting eWay Properties” on page 48**
- **“Deploying a Project” on page 49**

5.3 Summary: Sample Collaboration (Java) Project

This section explains generally how to implement the ADABAS eWay using the eGate Project sample with Java-based Collaborations. This sample is included on your installation CD-ROM. The sample is named **ADABAS__Project_JCE**. It allows you to observe an end-to-end data-exchange scenario involving eGate and the ADABAS eWay.

For instructions on how to import the sample Project, see the **procedure on page 38**. For an overview of the sample Project and what is does, see **"Overview: Sample Projects" on page 40**.

See the *eGate Tutorial* for details on how to build an end-to-end Project.

5.3.1 Using the eWay With Java-based Collaborations

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.

Creating Business Rules Within Collaboration Definitions

A Collaboration Editor (Java) 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 ADABAS OTD and other OTDs. ADABAS OTD nodes represent ADABAS functions, which are in turn able to call ADABAS methods.

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

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

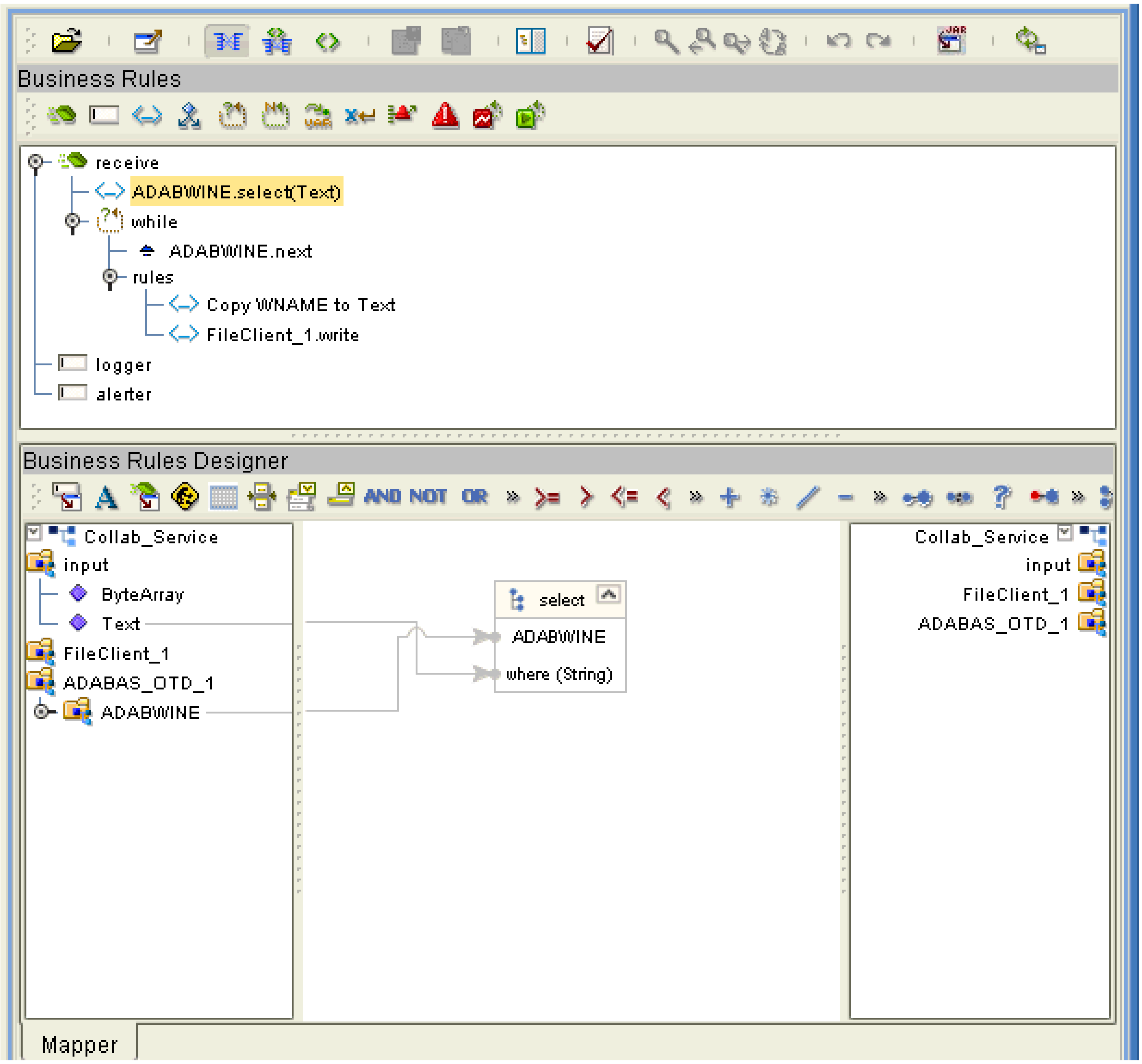
Using the Collaboration Editor (Java)

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.

See the *eGate Integrator User's Guide* for complete information on how to use the Collaboration Editor (Java). To complete a Java-based Collaboration Definition, you can use this editor to create Business Rules.

Figure 25 on page 47 shows an example of the Collaboration Editor (Java).

Figure 25 Collaboration Editor (Java) With Business Rules



Business Rules

```
receive
  ADABWINE.select(Text)
  while
    ADABWINE.next
    rules
      Copy WNAME to Text
      FileClient_1.write
logger
alerter
```

Business Rules Designer

5.3.2 Completing the Project

For instructions on how to finish implementing your Project, see the following sections:

- **"Creating the Project's Environment" on page 48**
- **"Setting eWay Properties" on page 48**
- **"Deploying a Project" on page 49**

5.4 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 ADABAS 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.

5.5 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 ADABAS eWay properties, see **Chapter 3**.

To set properties for the File eWays

1. From the **Project Explorer**, open the **CMap_ADABASSelect** 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.

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. 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** sheet.

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 ADABAS eWays

1 To begin changing the default properties for the ADABAS eWay, double-click the **ADABAS1** external system's eWay icon.

The eWay **Properties** sheet 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 ADABAS External System**. Give this system the same name as you gave the corresponding external application (**ADABAS1**) created on the Connectivity Map in the **Project Explorer**.

For details on how to do these operations, see the *eGate Tutorial.*

6 In the left pane, right-click the **ADABAS1** external system icon and select **Properties** from the pop-up menu.

The eWay **Properties** sheet 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.

Project deployment

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

5.6 Deploying a Project

This section provides general procedures for Project deployment.

5.6.1 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 48** 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.

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*.

5.6.2 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*.

Chapter 6

Using ADABAS OTDs

This chapter describes operations you can perform using ADABAS Object Type Definitions (OTDs) in eGate.

Chapter Topics

6.1 Overview

This section explains the types of ADABAS OTDs used with the eWay within the Collaboration Editor (Java), including the OTD's methods.

You can use ADABAS OTD methods with:

- Tables (table OTD)
- Views (view OTD)
- Prepared statements (prepared statement OTD)

Views are read-only and for informational purposes only. Use the OTD wizard (see **Chapter 4**) to create the ADABAS Database OTDs.

User-defined OTD

You can use the OTD wizard to create an eGate User-defined OTD. See the *eGate Integrator User's Guide* for a complete explanation of how to create a User-defined OTD.

6.2 Using Tables

A table OTD represents a database table. It consists of fields and methods. Fields correspond to the columns of a table, while methods are the operations that you can apply to the OTD. This setup allows you to perform select, update, insert, and delete SQL operations in a table.

6.2.1 Using the select Method

The **select** method returns result sets according to a group of predefined defaults. However, you can change these defaults, if desired. In using the **select** method, you can specify the following types of result sets:

- TYPE_FORWARD_ONLY
- TYPE_SCROLL_INSENSITIVE
- TYPE_SCROLL_SENSITIVE

You can also specify the following result sets with a type of concurrency:

- CONCUR_READ_ONLY
- CONCUR_UPDATABLE

To perform an update, insert or delete operation, the type of the result set returned by the **select** method must be CONCUR_UPDATABLE.

Instead of specifying the type of result set and concurrency in the **select** method, you can also use the following methods:

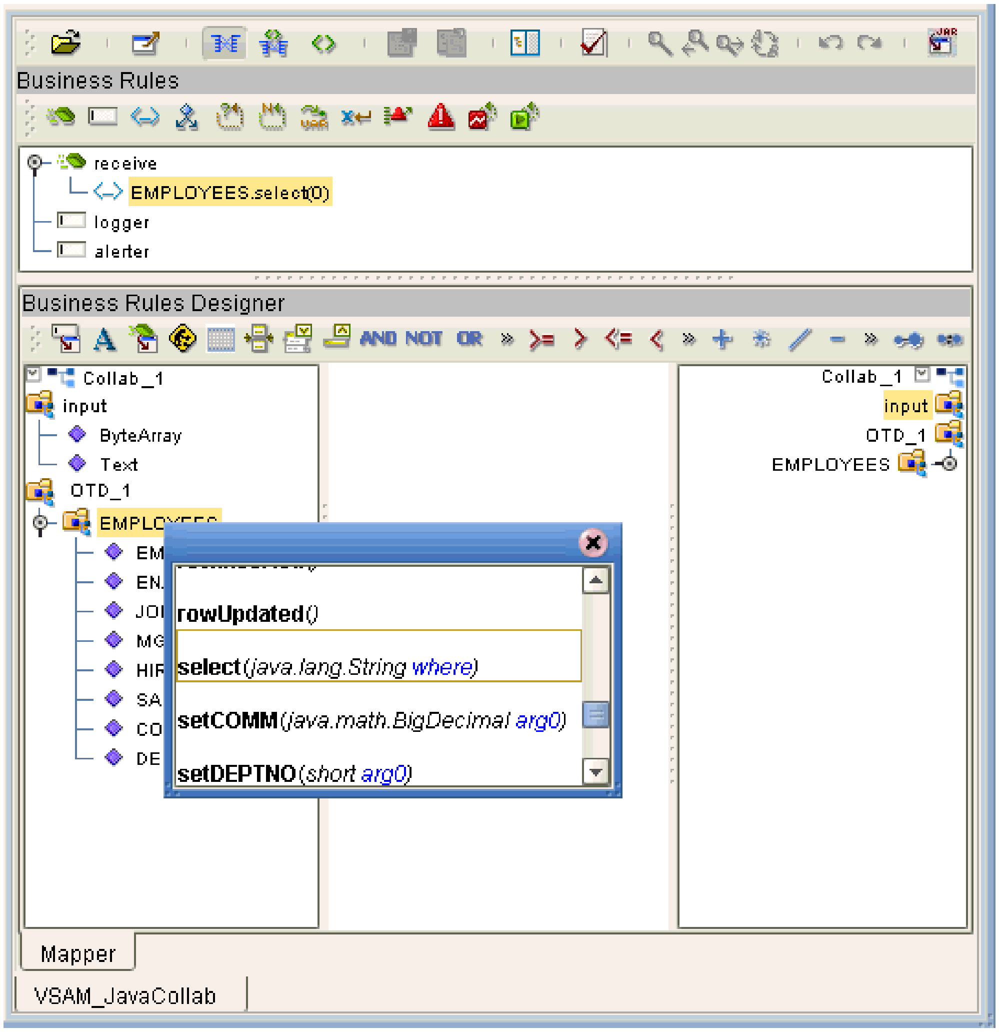
- **setConcurrencytoUpdatable**
- **setConcurrentlytoReadOnly**
- **setScrollTypetoForwardOnly**
- **setScrollTypetoScrollSensitive**
- **setScrollTypetoInsensitive**

To call a method in the Collaboration Editor (Java)

1. Open the eGate Enterprise Designer's Collaboration Editor (Java) for the desired Collaboration Definition.
2. In the editor, navigate to the Business Rules Designer.
3. Right-click on the desired node (right or left pane) where you want to call the method.

A pop-up **Method** selection menu appears, showing the methods available for the node. See Figure 26.

Figure 26 Method Selection Menu for the select Method



4 Select the desired method.

A **Method** box appears in the center pane of the Business Rules Designer. This box represents the chosen method.

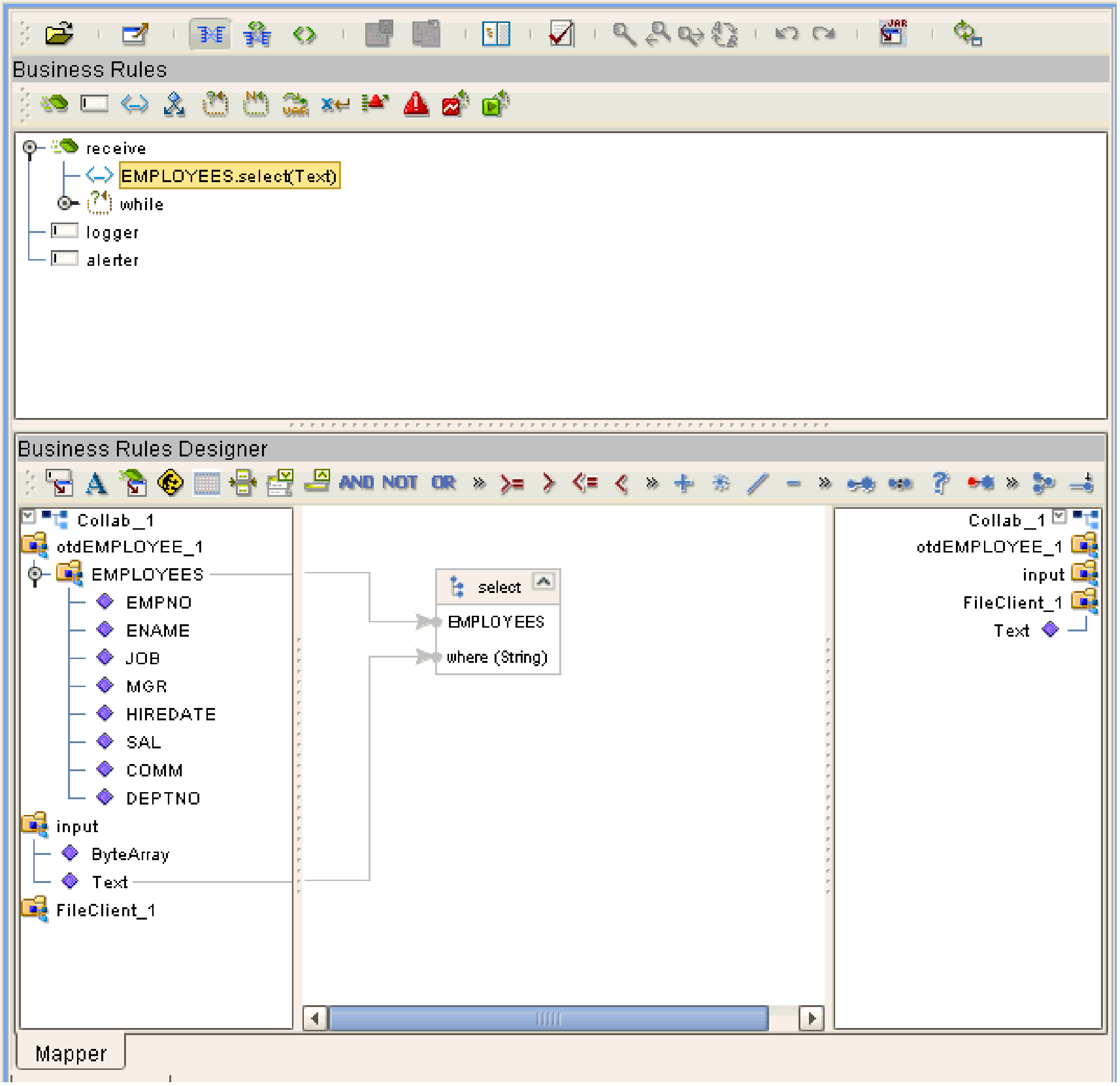
Note: *Using the **commit** method is optional for all update operations because the Collaboration automatically commits the current transaction when it is done.*

6.2.2 Select Operations

To perform a select operation on a table

1 Call the **select** method with the **where** clause specified, if necessary. See Figure 27.

Figure 27 Select Operation: select Method



The text from the **Text** node instructs the Business Process exactly which data needs to be selected. For example, the content of the input text could be **EMPNO>100**.

2 Loop through the result set using the **next** method. See Figure 28.

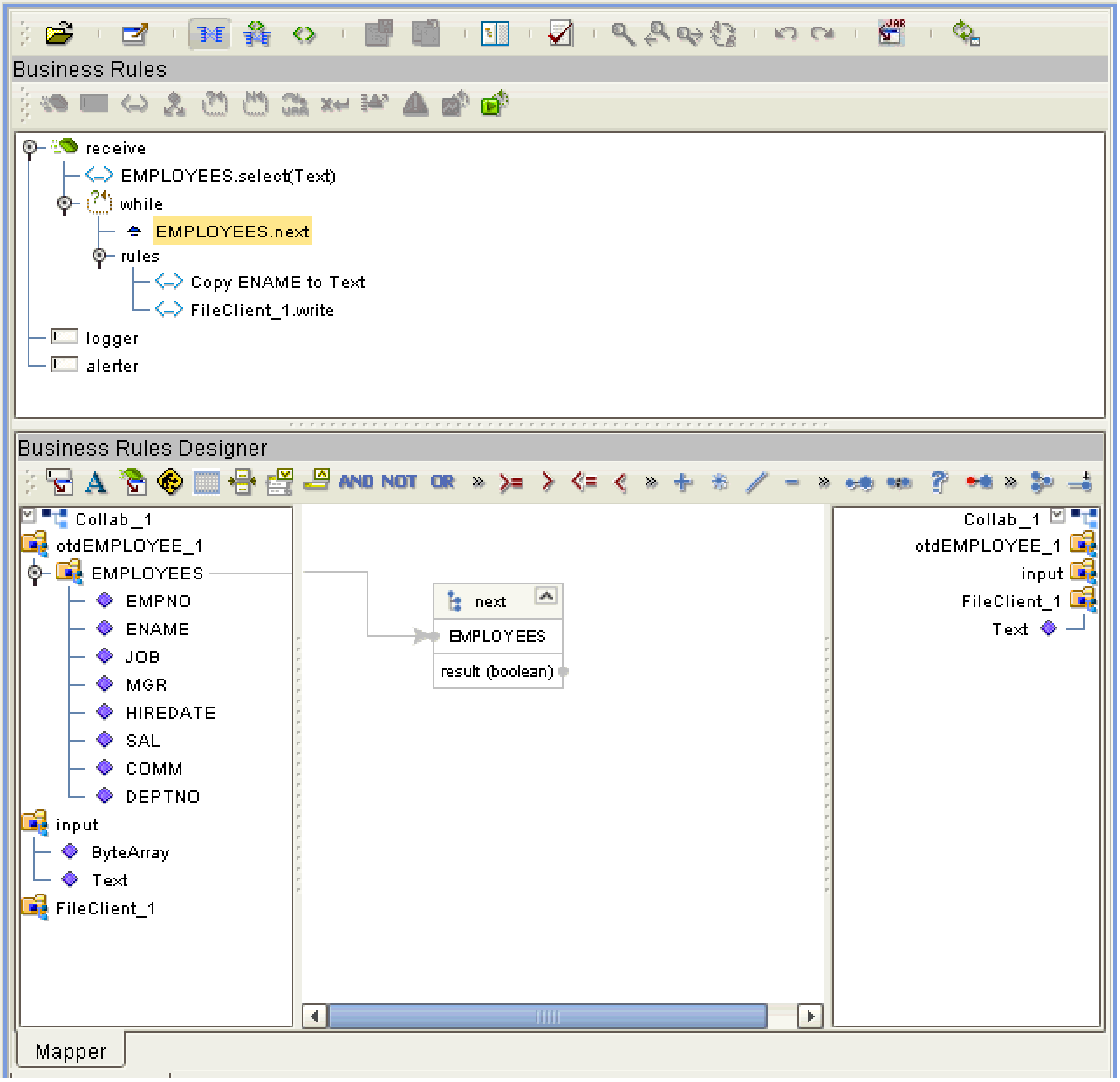


Figure 28 Select Operation: next Method

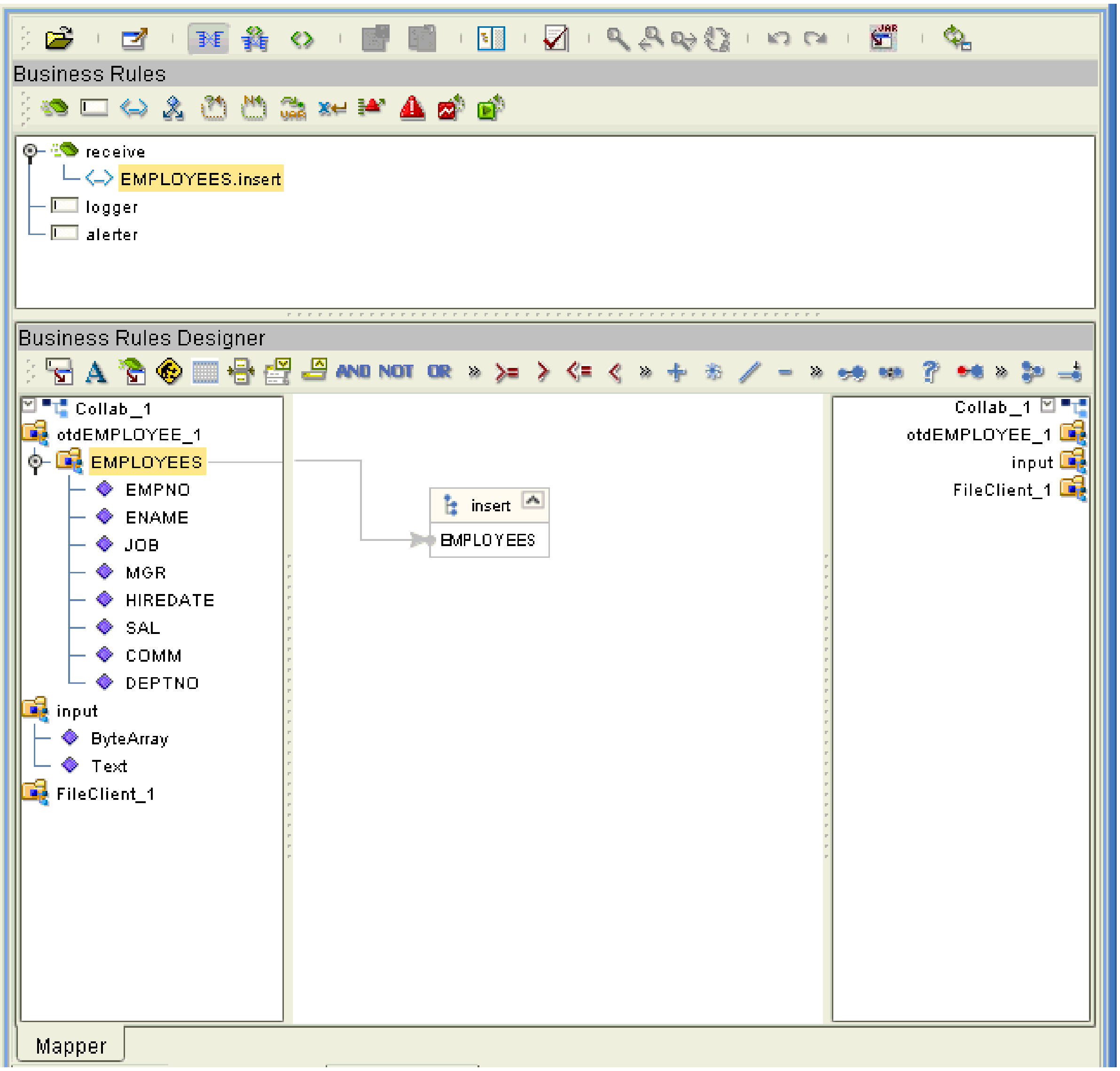
3 For each loop, process the return record.

6.2.3 Insert Operations

To perform an insert operation on a table

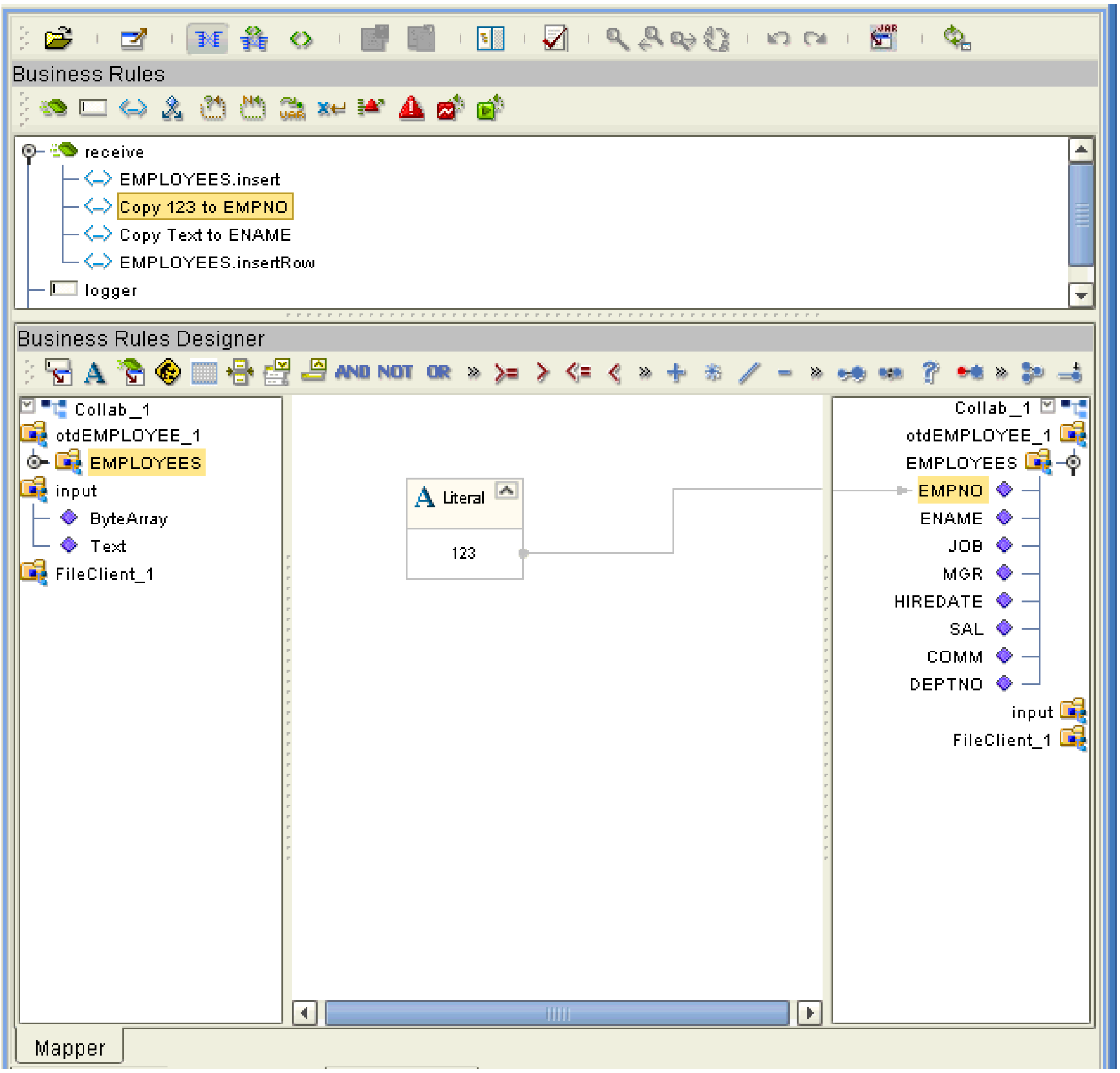
1 Call the **insert** method. See Figure 29.

Figure 29 Insert Operation: insert Method



2 Set the fields of the table OTD where you want to do the insertion and insert the row using a **Literal**. See Figure 30.

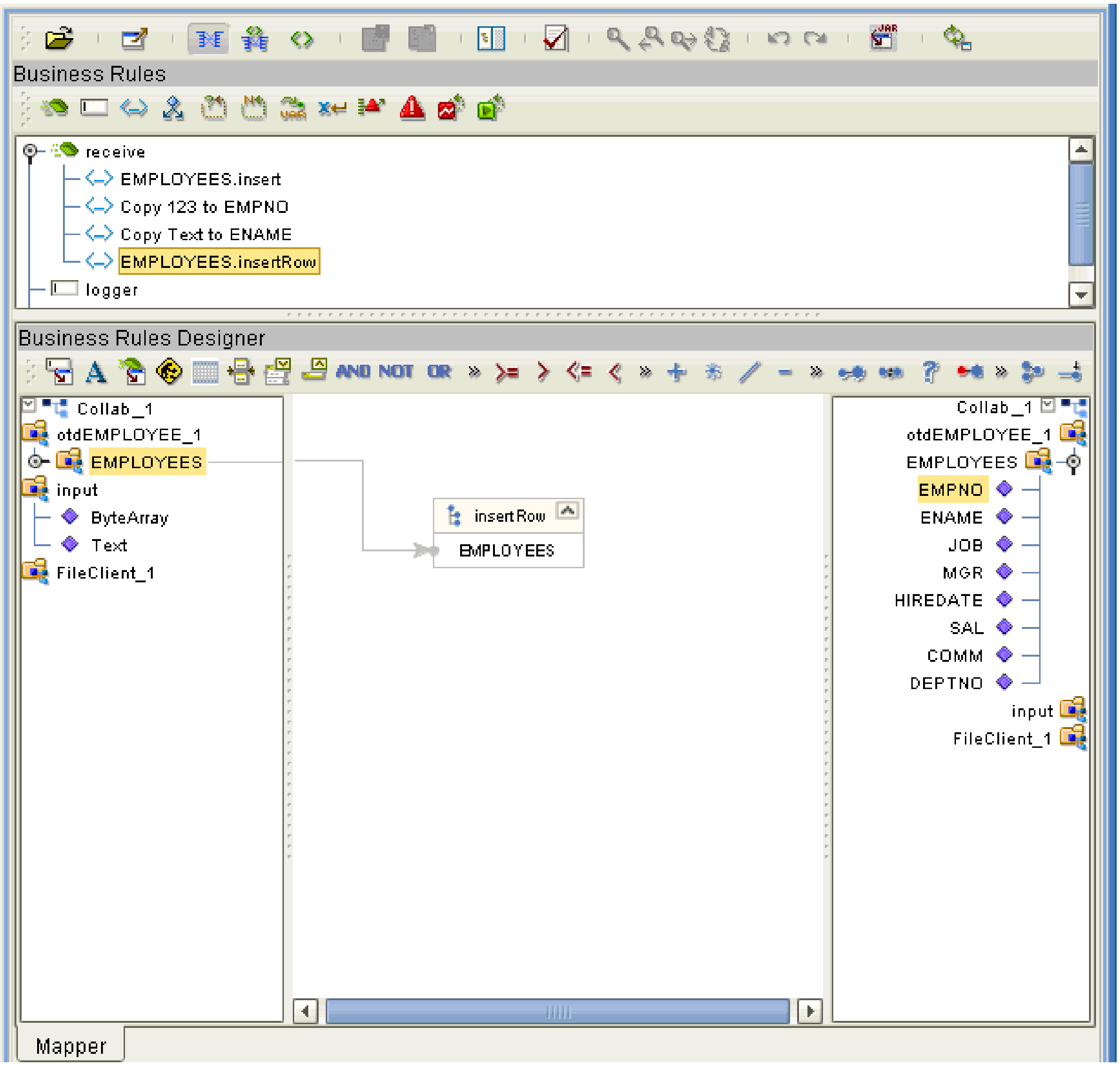
Figure 30 Insert Operation: Literal Indicates the Value Inserted



If you want to update multiple columns, you can repeat the **Copy** rule, as desired.

3 Activate the insertion by calling the **insertRow** method. See Figure 31.

Figure 31 Insert Operation: insertRow Method



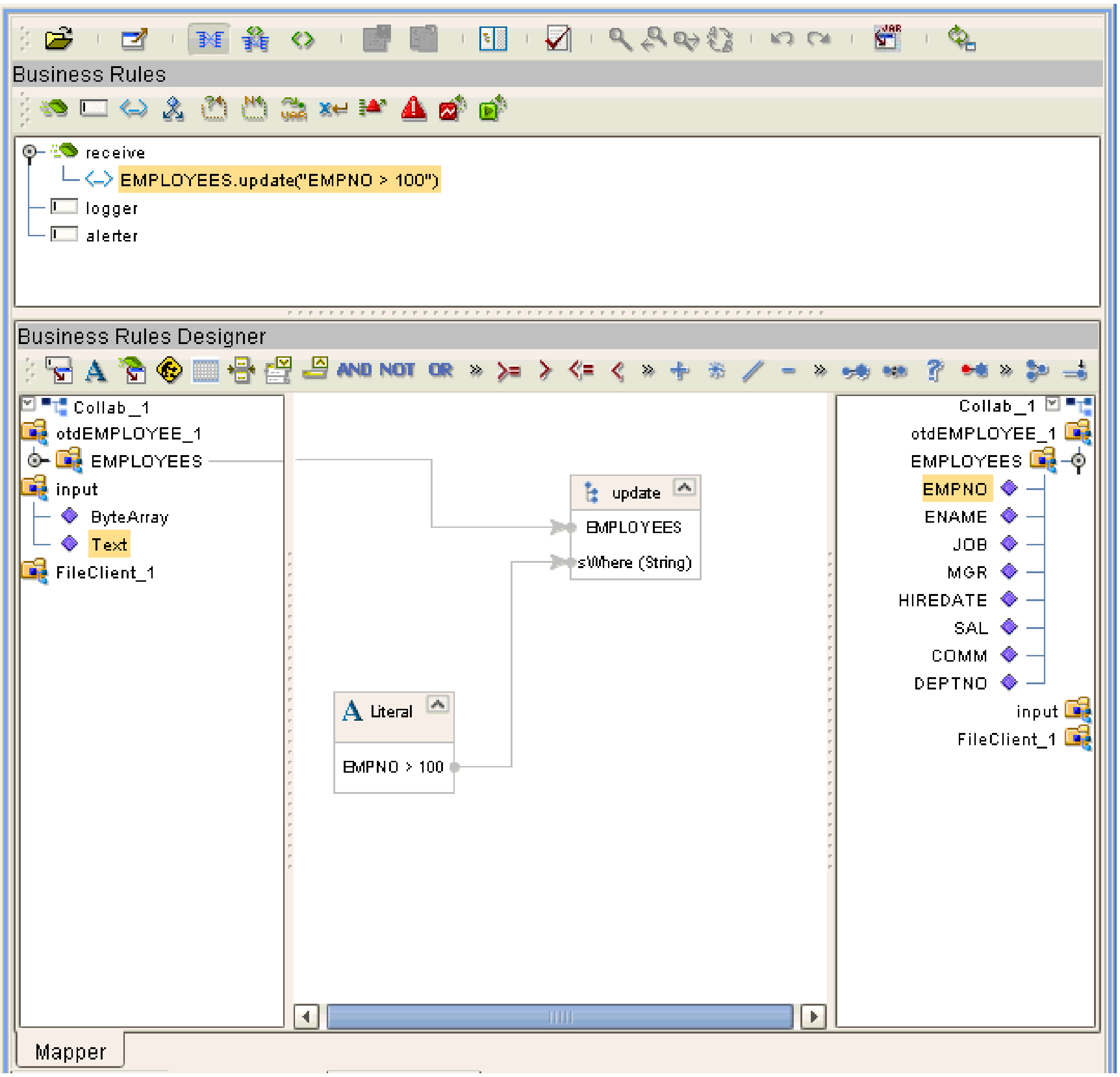
For multiple insert operations, you can use a **while** rule.

6.2.4 Update Operations

To perform an update operation on a table

1 Call the **update** method with the **where** clause specified, if necessary. See Figure 32.

Figure 32 Update Operation: update Method With Literal

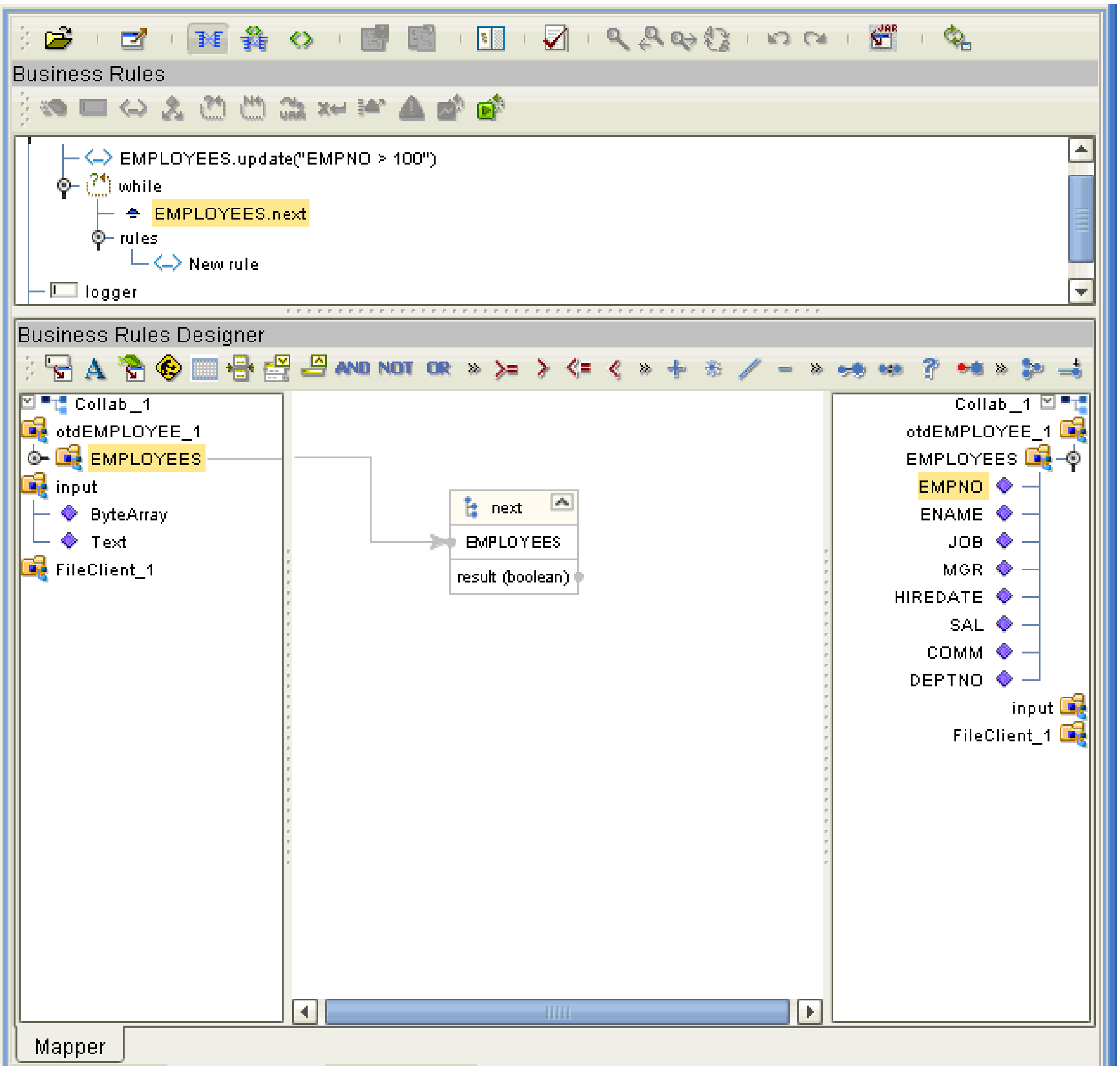


receive
EMPLOYEES.update("EMPNO > 100")
logger
alerter

In this case, a **Literal** is used instead of the text node as shown in Figure 29 on page 56. Either way of specifying the **where** clause is acceptable.

2 Call the **next** method. See Figure 33.

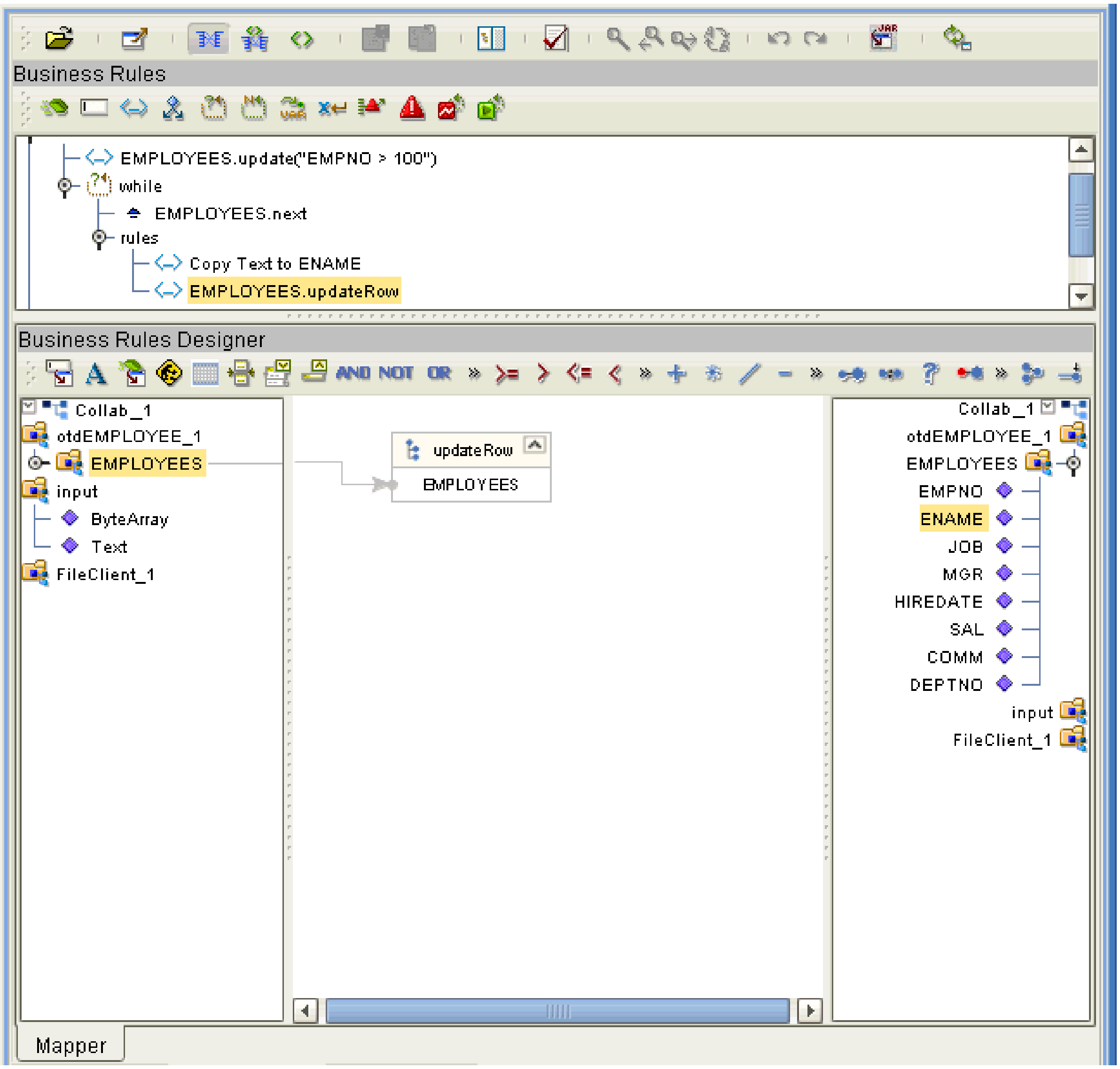
Figure 33 Update Operation: next Method



EMPLOYEES.update("EMPNO > 100")
while
EMPLOYEES.next
rules
New rule
logger

3 Update the row by calling **updateRow**. See Figure 34.

Figure 34 Update Operation: updateRow Method



```
EMPLOYEES.update("EMPNO > 100")
while
    EMPLOYEES.next
    rules
        Copy Text to ENAME
        EMPLOYEES.updateRow
```

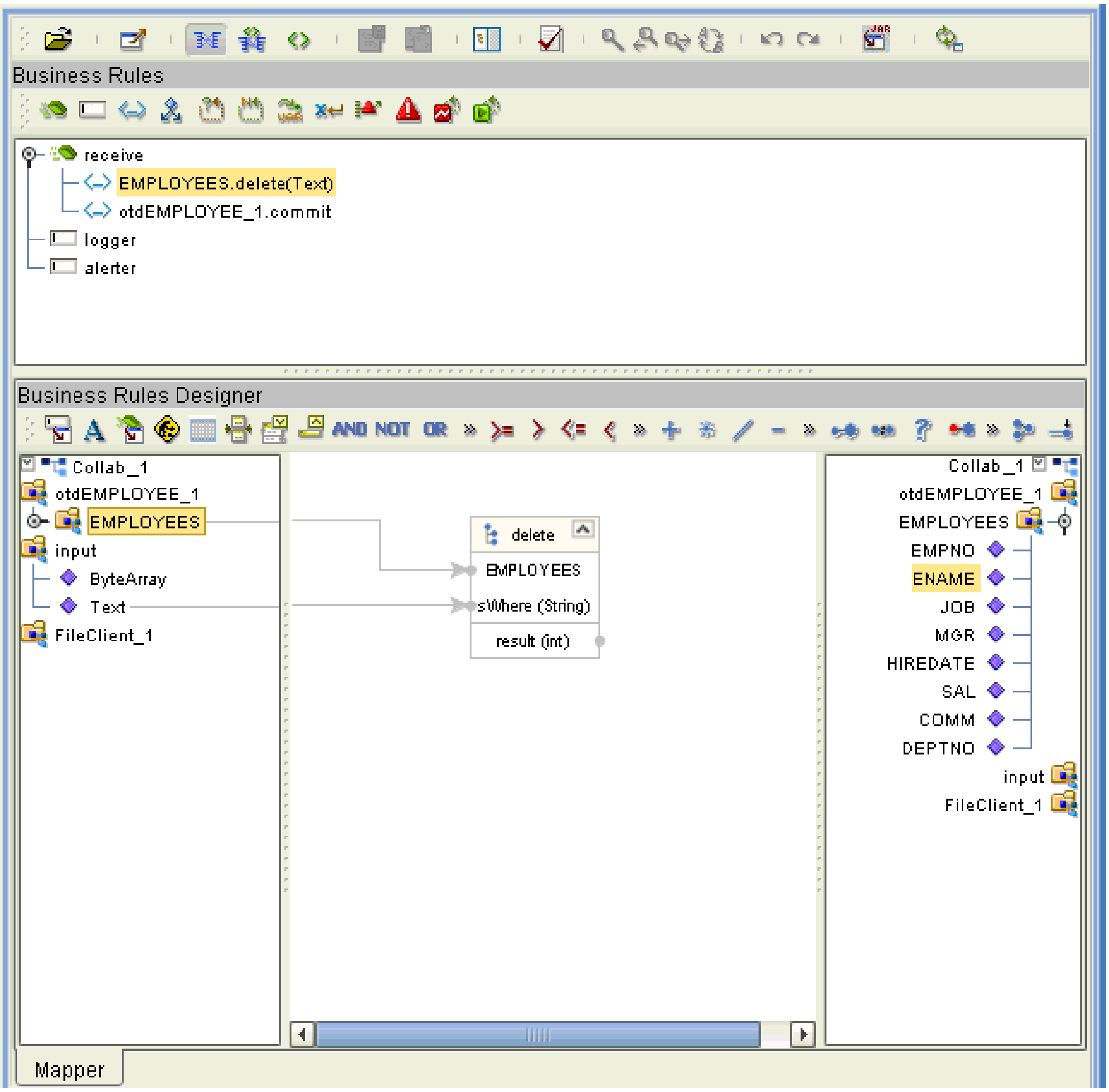
To update more than one column, you can repeat the **Copy** rule, as desired.

6.2.5 Delete Operations

To perform a delete operation on a table

- Call the **delete** method with the **where** clause specified, if necessary. See **Figure 35 on page 62**.

Figure 35 Delete Operation: delete Method



Business Rules

receive
EMPLOYEES.delete(Text)
otdEMPLOYEE_1.commit
logger
alerter

Business Rules Designer

6.3 Using Views

Views are used to look at data from selected columns within selected tables. View OTDs are read-only.

For select operations, see **"Select Operations" on page 54**.

6.4 Using Prepared Statements

You can perform table operations with prepared statements with the same method nodes you use for the regular table operations, that is, select, update, insert, and delete SQL operations.

Prepared statements allow you to create any valid SQL statements that comply with ANSI standards. Using this feature, you can execute **select** statements such as **joins** and **union**, **insert**, **update** and **delete** operations. You can also include database operations such as **to_date**, to manipulate input or output values. Parameter markers are also supported.

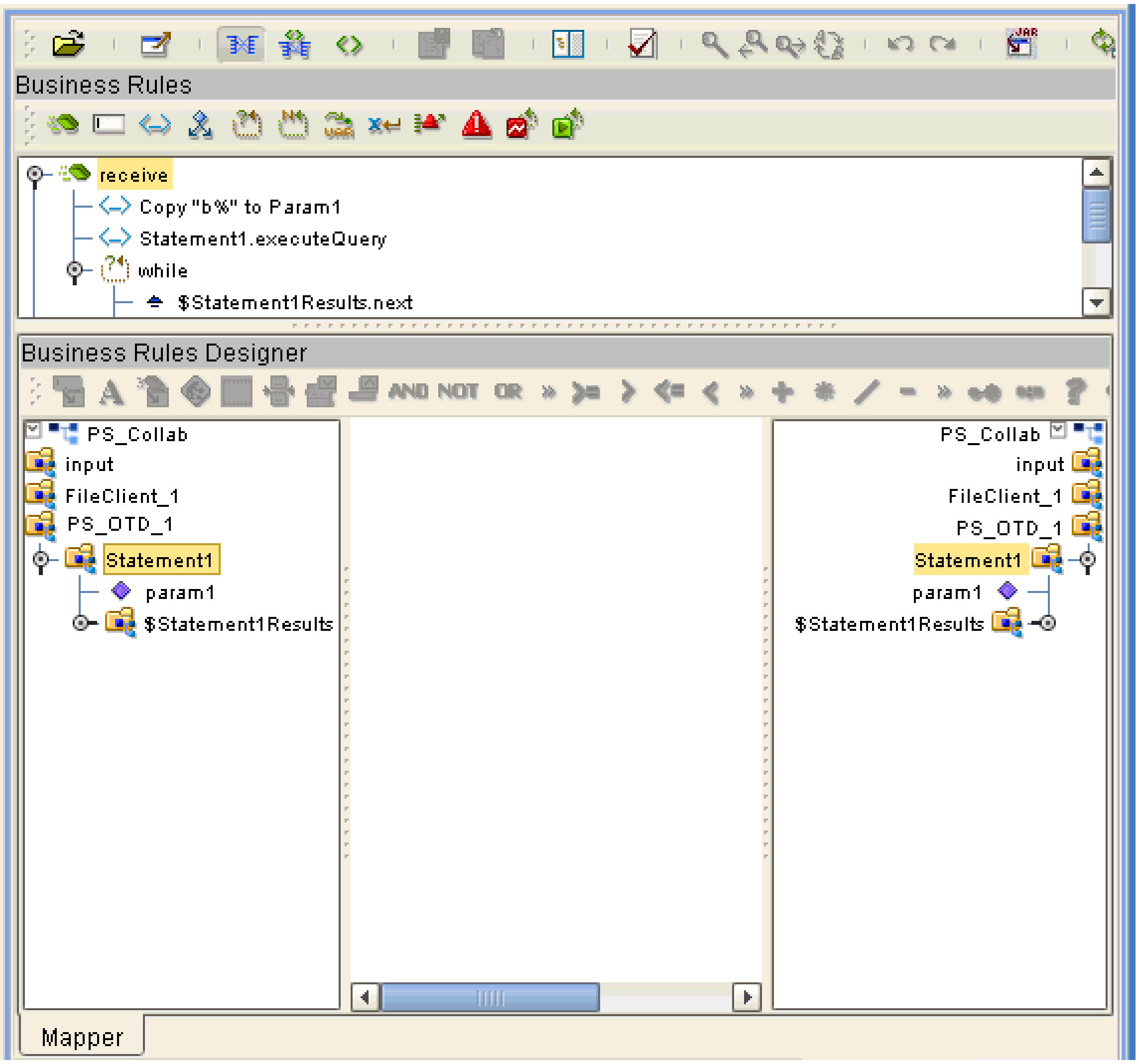
Any prepared statement you create shows up in the Collaboration Editor (Java) on the OTD where it was created, as a node with the name you assigned to it. You can implement the statement by dragging its node onto the method box for the action (the desired select, update, insert, or delete operation) in which you want the statement to be used.

To use a prepared statement in a Collaboration (Java)

1. Using the Enterprise Designer and ADABAS OTD wizard, create an OTD that contains a prepared statement. See the **procedure on page 31** for details.
2. Create the desired Collaboration Definition (Java) using this OTD.

   The Collaboration Editor (Java) displays the OTD with a **Prepared Statement** node as shown in **Figure 36 on page 64**.

Figure 36 Prepared Statement Collaboration

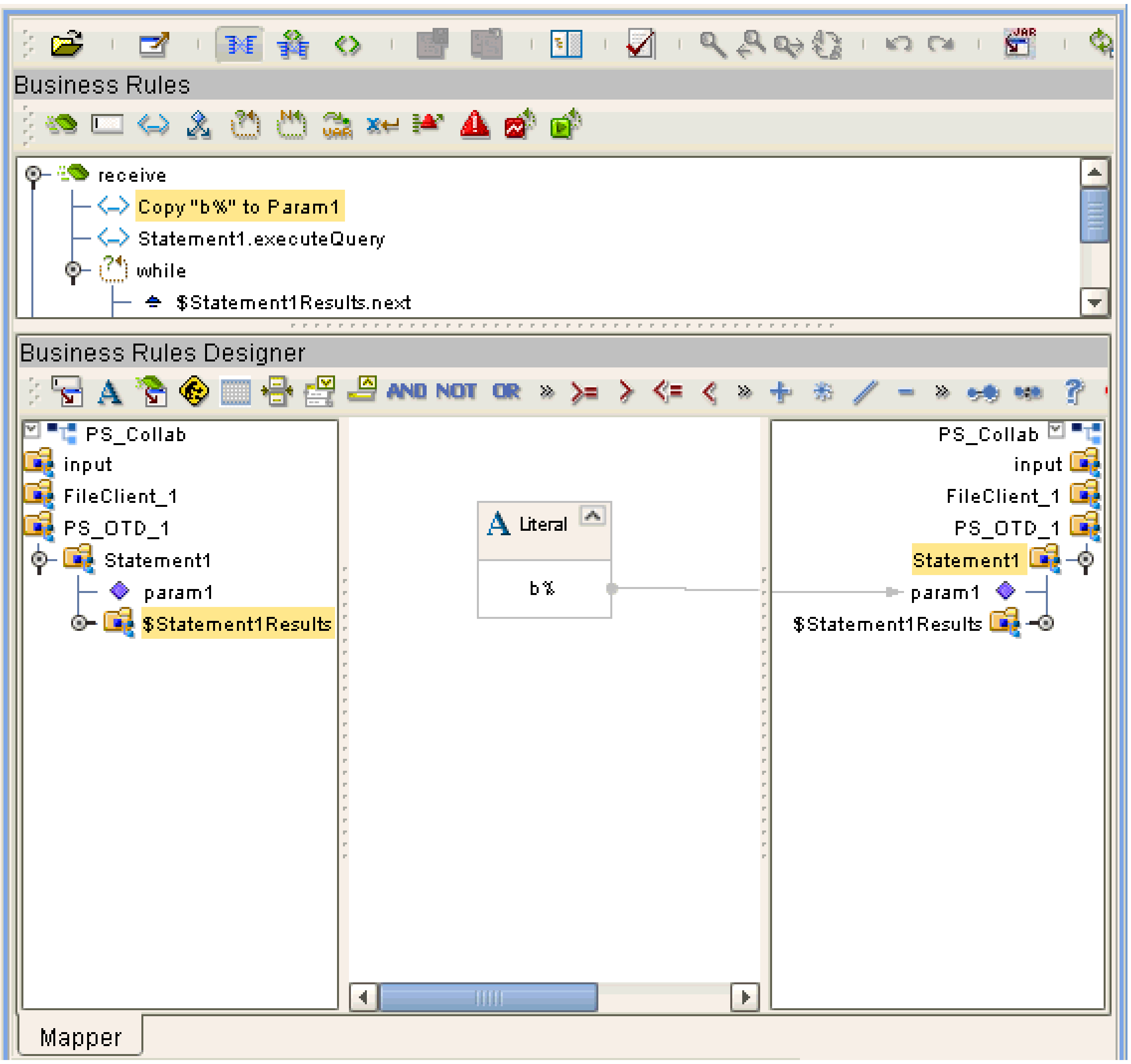


```
receive
  Copy "b%" to Param1
  Statement1.executeQuery
  while
    $Statement1Results.next
```

In the OTD structure, you can see both the input parameter and the result set represented as nodes.

3 You can assign a **Literal** value or the input from another OTD to the parameter. In the example shown in Figure 37, the **Literal** value assigned to the parameter is **b%**.

Figure 37 Prepared Statement Collaboration: Assigning a Literal Value



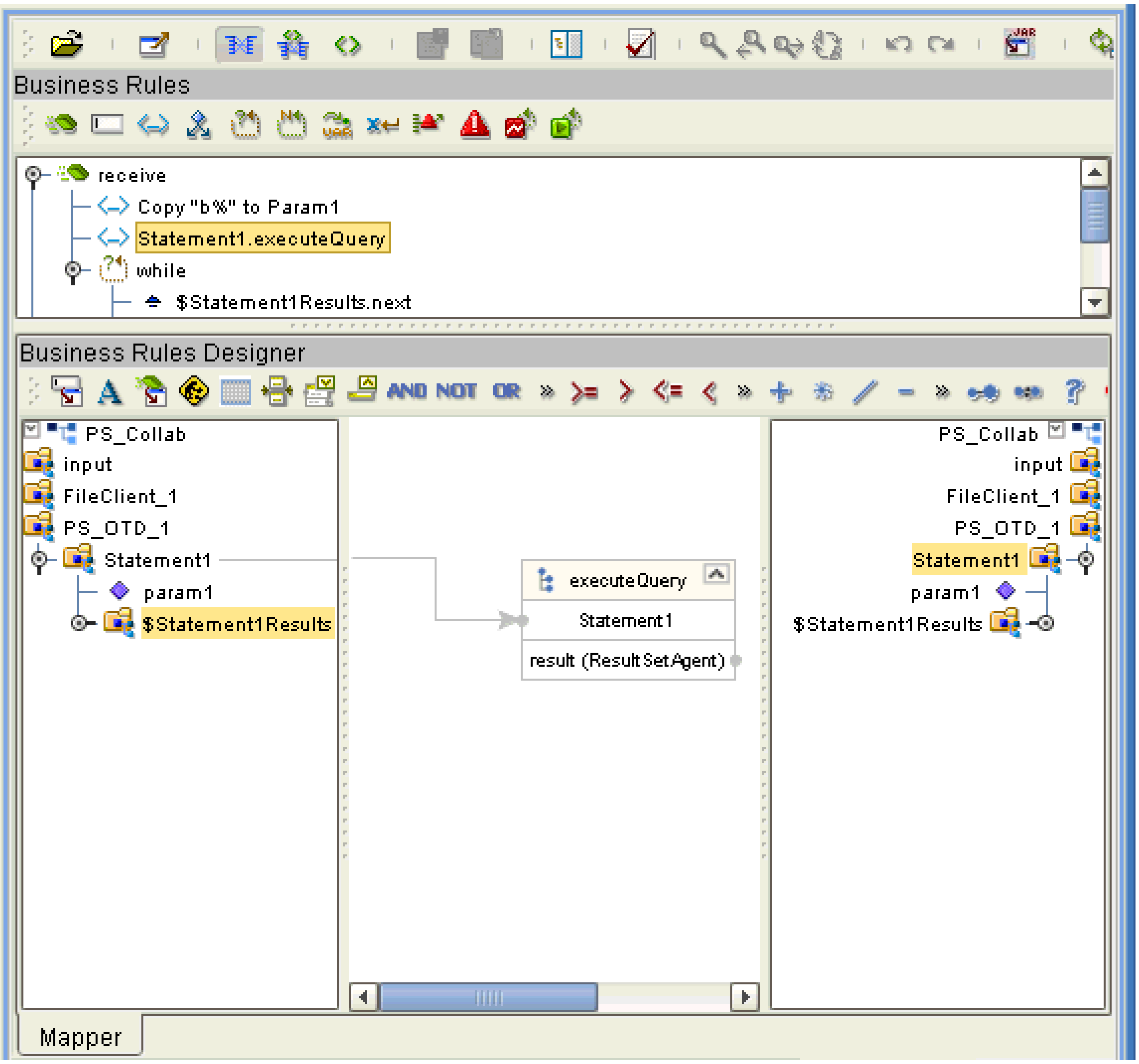
Business Rules

```
receive
  Copy "b%" to Param1
  Statement1.executeQuery
  while
    $Statement1Results.next
```

Business Rules Designer

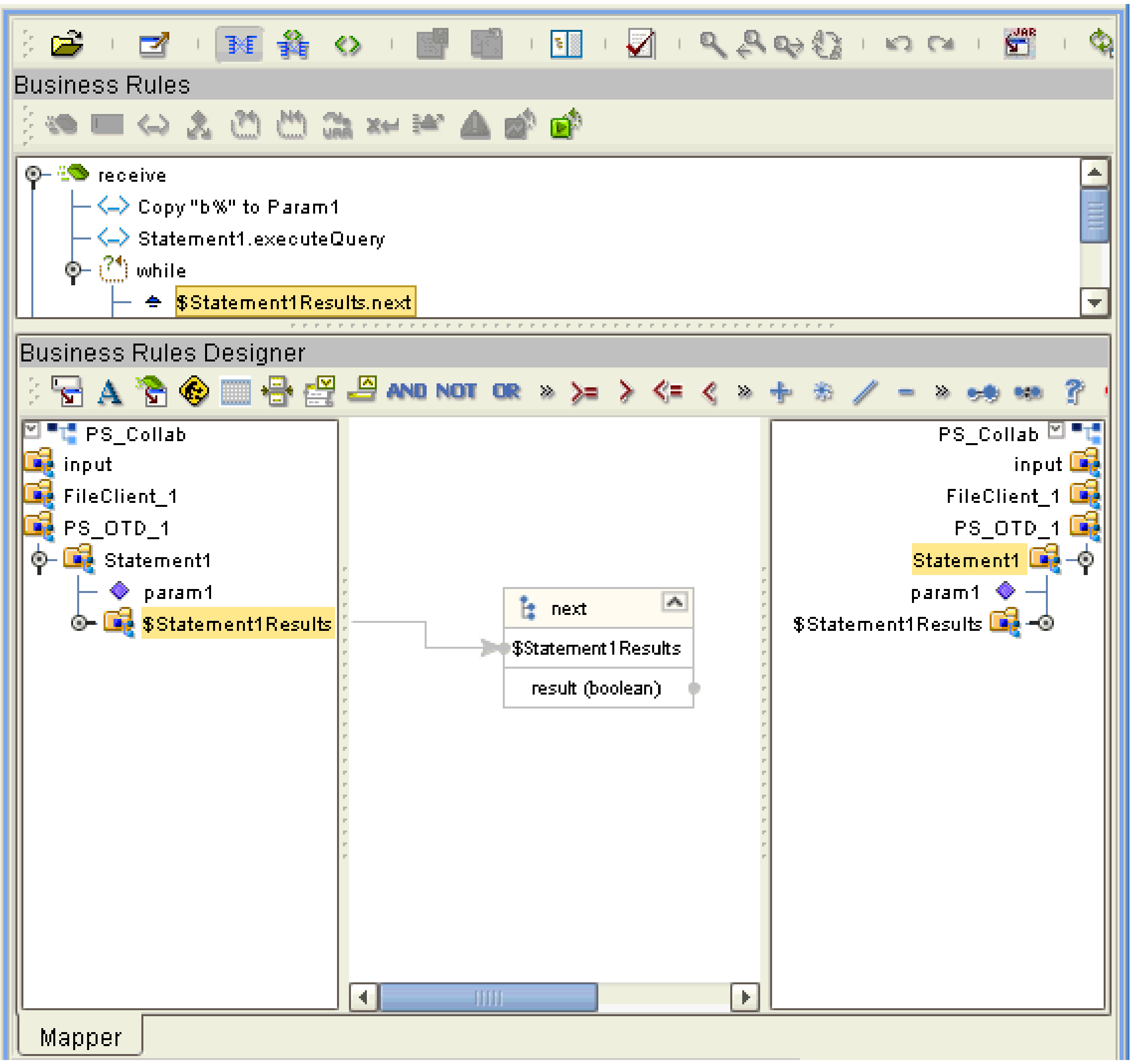
4 To run the prepared statement, select the **executeQuery** method from the **Prepared Statement** node. See Figure 38.

Figure 38 Prepared Statement Collaboration: Running the Statement



5 To view the result set after the query is finished, create a **while** rule. For the condition, create a **next** method from the result set node (**$Statement1Results** in the example). See Figure 39.

Figure 39 Prepared Statement Collaboration: Viewing Results



You can continue creating your business logic by adding more Business Rules, if desired.

Chapter 7

Using eWay Java Methods

This chapter provides an overview of the Java classes/interfaces and methods contained in the ADABAS eWay. These methods are used to extend the functionality of the eWay.

Chapter Topics

7.1 ADABAS eWay Methods and Classes: Overview

The ADABAS eWay exposes various Java methods to add extra functionality to the eWay. These methods make it easier to set information in the ADABAS eWay Object Type Definitions (OTDs), as well as get information from them.

7.1.1 Relation to eWay Properties

The nature of this data transfer depends on the properties you set for the eWay in the eGate Enterprise Designer. For more information on the eWay's properties, see **Chapter 3**.

7.1.2 ADABAS eWay Javadoc

For a complete list of the Java methods within the classes listed in this chapter, refer to the **Javadoc**. You can download the Javadoc while you are installing the eWay. For complete instructions, see the *SeeBeyond ICAN Suite Installation Guide*.

7.2 eWay Java Classes/Interfaces

The ADABAS eWay provides the following Java classes/interfaces:

- **ADABASApplicationConnection**: The class that implements the **ApplicationConnection** interface for the database application connection.
- **ADABASApplicationConnectionFactory**: A factory class for creating the ADABAS application connection.
- **ADABASCallableStatementAgent**: The agent that hosts an ADABAS callable statement.
- **ADABASConnector**: The connector class for the ADABAS database eWay.
- **ADABASCPManagedConnectionFactoryExt**: The factory for the ADABAS connection pool managed connection.
- **ADABASSession**: The session that hosts a connection interface.
- **ADABASTableResultSet**: The result set to map selected records of a table in ADABAS.
- **ADABASXAManagedConnectionFactoryExt**: The factory for ADABAS's XA-managed connection.

Index