SeeBeyond ICAN Suite

eTL Integrator User's Guide

Release 5.0.4

Version 20040604110030.

Contents

About This Guide

This chapter provides an overview of the this user's guide, including its contents and writing conventions.

In This Chapter

1.1 Contents of This Guide

This guide contains the following information:

- **Chapter 1, "Introducing eTL Integrator" on page 9** provides an overview of eTL Integrator and provides a brief overview of the steps necessary to build and deploy eTL Projects.
- **Chapter 2, "Installing eTL Integrator" on page 15** describes how to install eTL Integrator, its documentation, and its sample Projects.
- **Chapter 3, "Building and Deploying eTL Projects" on page 17** describes how to build eTL Projects.
- **Chapter 4, "Building Business Logic for eTL Projects" on page 36** describes how to build the business logic for eTL Projects.
- **Chapter 5, "Working with eTL Sample Projects" on page 72** describes how to import and use the sample Projects provided with eTL Integrator.
- **Chapter 6, "Managing Runtime eTL Projects" on page 85** describes how to monitor and reconfigure deployed eTL Projects.

1.2 Writing Conventions

The following writing conventions are observed throughout this document.

Table 1 Writing Conventions

Text	Convention	Example
Button, file, icon, parameter, variable, method, menu, and object names.	**Bold** text	▪ Click **OK** to save and close. ▪ From the **File** menu, select **Exit**. ▪ Select the **logicalhost.exe** file. ▪ Enter the **timeout** value. ▪ Use the **getClassName()** method. ▪ Configure the **Inbound** File eWay.
Command line arguments and code samples	`Fixed` font. Variables are shown in ***bold italic***.	`bootstrap -p` ***`password`***
Hypertext links	**Blue** text	**http://www.seebeyond.com**

Additional Conventions

Windows Systems

For the purposes of this guide, references to "Windows" will apply to Microsoft Windows Server 2003, Windows XP, and Windows 2000.

Path Name Separator

This guide uses the backslash ("\") as the separator within path names. If you are working on a UNIX system, please make the appropriate substitutions.

1.3 Supporting Documents

The following SeeBeyond documents provide additional information about the SeeBeyond ICAN Suite:

- *SeeBeyond ICAN Suite Installation Guide*
- *eGate Integrator User's Guide*
- *eGate Integrator System Administrator Guide*
- *eInsight Business Process Manager User's Guide*
- *eInsight Enterprise Service Bus User's Guide*
- *File eWay Intelligent Adapter User's Guide*
- *DB2 Universal Database eWay Intelligent Adapter User's Guide*
- *Oracle eWay Intelligent Adapter User's Guide*
- *Sybase eWay Intelligent Adapter User's Guide*
- *SQL Server eWay Intelligent Adapter User's Guide*

1.4 SeeBeyond Web Site

The SeeBeyond Web site is a useful source for product news and technical support information at **www.seebeyond.com**.

Chapter 1

Introducing eTL Integrator

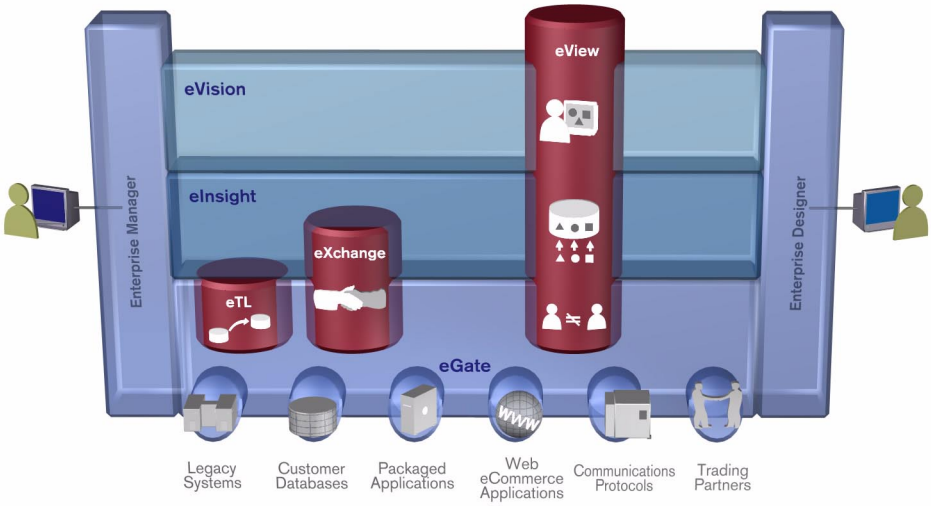
This chapter provides an overview of eTL Integrator and its implementation in ICAN Projects.

In This Chapter

1.1 About eTL Integrator

Extraction Transform and Load (ETL) is a data integration technology that extracts data from several heterogeneous data sources, transforms the data, then loads the data in a uniform format into a target data source.

Figure 1 eTL and the ICAN Product Suite



SeeBeyond's eTL Integrator technology is optimized for very large record sets and build data scenarios that are fully integrated with the SeeBeyond ICAN suite (Integrated Composite Application Network Suite) to unify the domains of eAI (eBusiness and Application Integration), and Enterprise Information Integration (EII). With these unified domains you can build unprecedented solutions using both message based processing (eGate) and dataset based processing (eTL) technologies.

The eTL Integrator product provides excellent performance at runtime for high volume extraction and load of tabular data sets. The eTL Integrator can be integrated into the enterprise business processes or used as a standalone product.

The eTL Integrator product can be used to acquire a temporary subset of data for reports or other purposes, or acquire a more permanent data set for the population of a data mart or data warehouse. The product may also be used for conversion of one database type to another or for the migration of data from one database or platform to another.

1.1.1 The ETL Process

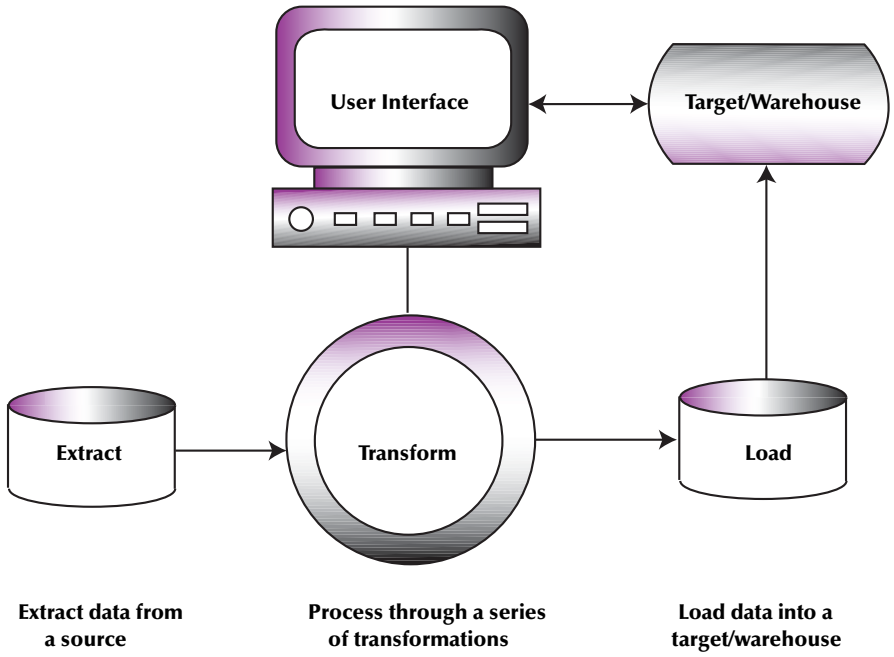
In an ETL process, data is extracted from data sources. The data is then transformed (or processed), using rules, operators, or filters, into a desired state suitable for loading into a database or data warehouse. See the following figure.

In managing databases, extract, transform, load (ETL) refers to three separate functions combined into a single programming tool.

1. First, the extract function reads data from a specified source database and extracts a desired subset of data.
2. Next, the transform function works with the acquired data – using rules or lookup tables, or creating combinations with other data – to convert it to the desired state.

3 Finally, the load function is used to write the resulting data to a target database, which may or may not have previously existed.

Figure 2 The ETL Process



1.1.2 eTL Features

Transformation Capability

eTL Integrator enables seamless filtering and data transformation.

- Merge/upsert (updates or inserts as appropriate)
- Drag and drop GUI design features (create joins across disparate data sources)
- Validate Collaborations before performing the ETL processes
- User friendly, state-of-the-art, design tools reduce development time and cost
- GUI based Collaboration editor employs drag and drop design features
  - User friendly Wizards (easy OTD creation)
  - Graphical operators (dragged from a toolbar)
  - Graphical tools (create underlying SQL)
- Tight integration among ICAN Suite business data systems
  - Web Services interface
  - Seamless integration with the ICAN Suite
  - ICAN Suite provides versioning and history

Operators and Transformation Tools and Functionality

Development is simplified with GUI based development tools that are appropriate for SQL Collaborations. Graphical drag and drop modeling tools enable SQL operations in various categories:

- Number
- Date
- Comparison
- Boolean
- SQL operators
- Mathematical operators
- String manipulations
- Conditional data extractions and transformations
- Join
  - Auto-detect primary key relationships between tables, as indicated in OTDs
  - Between tables from disparate data sources that have no relationship
  - Supports inner, left, right, and full outer joins
- Lookups
  - Extensive lists of operators allows for the creation of lookups as part of the eTL process, using joins across tables
- Merge
  - Automatic update if row exists
  - Automatic insert if row doesn't exist
- Test data and test runs
- Runtime variables (configured by the user)

Architecture

Fully integrated with the ICAN Suite to make eTL a more versatile and powerful tool.

- A deployed eTL engine runs as a JCA compliant (J2EE) resource adapter inside the SeeBeyond Integration server.
- The business rules defined by the eTL Collaboration definition are stored in the SeeBeyond Repository.
- At deployment time, the business rules are used to generate the appropriate platform specific SQL.
- eTL Integrator leverages OTDs defined in the Enterprise Designer so you don't have to create OTDs specifically for an eTL Collaboration.

1.2 System Requirements

The system requirements for eTL Integrator are the same as those for eGate Integrator and eInsight Business Process Manager. For information, refer to the *SeeBeyond ICAN Suite Installation Guide*.

1.3 Supported Operating Systems

eTL Integrator is available for the following operating systems:

- Windows Server 2003, Windows XP SP1a, and Windows 2000 SP3
- Sun Solaris 8 and 9
- HP Tru64 V5.1A
- HP-UX 11.0, 11i (PA-RISC), and 11i v2.0 (11.23)
- IBM AIX 5.1L and 5.2
- Red Hat Enterprise Linux Advanced Server 2.1
- Red Hat Linux 8

1.4 Required ICAN Suite Products

To use eTL Integrator, you must first purchase and install eGate Integrator version 5.0.4 or eInsight Business Process Manager version 5.0.4.

1.5 Supported ICAN Suite Products

eTL Integrator currently supports the following ICAN Suite products, available for purchase from SeeBeyond:

- eGate Integrator version 5.0.4
- eInsight Business Process Manager version 5.0.4
- eInsight Enterprise Service Bus version 5.0.4
- DB2 Universal Database eWay Intelligent Adapter version 5.0.3
- Oracle eWay Intelligent Adapter version 5.0.4
- SQL Server eWay Intelligent Adapter version 5.0.4
- Sybase eWay Intelligent Adapter version 5.0.4

1.6 Support File Formats

eTL Integrator supports fixed, delimited, and tabular ASCII files, including comma-separated value (CSV) files.

1.7 Supported Databases

eTL Integrator is compatible with the following external systems:

- DB2 Universal Database version 8.1
- Oracle version 8i with patch 8.1.7
- Oracle version 9i release 9.0.2
- Microsoft SQL Server 2000 with driver support for Data Direct Drivers JDBC 3.3
- Sybase Server 12.5

Chapter 2

Installing eTL Integrator

This chapter describes how to install eTL Integrator, its documentation, and the sample eTL Projects. This chapter also describes how to install the eWays that are supported by eTL Integrator.

In This Chapter

2.1 Installing eTL Integrator and Sample Projects

During the eTL Integrator installation process, the Enterprise Manager, a web-based application, is used to select and upload products as .sar files from the eGate installation CD-ROM to the Repository.

The installation process includes installing the following components:

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

Follow the instructions for installing the eGate Integrator in the *SeeBeyond ICAN Suite Installation Guide*, 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**, e**Insight.sar,** or **eInsightESB.sar** file, select and upload the following below as described in the *SeeBeyond ICAN Suite Installation Guide*:
   - **eTL.sar** (to install eTL Integrator)
   - **FileeWay.sar** (to install the File eWay, used in the sample Projects)
   - **eTLDocs.sar** (to install the user guide and the sample Projects)

   To use eTL Integrator with its supported eWays, download the following files:
   - **OracleeWay.sar** (to install the Oracle eWay)
   - **OracleeWayDocs.sar** (to install the user guide and JavaDocs)

- **SybaseeWay.sar** (to install the Sybase eWay)
- **SybaseeWayDocs.sar** (to install the user guide and JavaDocs)
- **SQLServereWay.sar** (to install the SQL Server eWay)
- **SQLServereWayDocs.sar** (to install the user guide and JavaDocs)
- **DB2eWay.sar** (to install the Oracle eWay)
- **DB2eWayDocs.sar** (to install the user guide and JavaDocs)

2 In the Enterprise Manager, click the **DOCUMENTATION** tab.

3 Click **eTL Integrator**.

4 In the right-hand pane, click **Download Sample**, and select a location for the .zip file to be saved.

For information about importing and using the sample Projects, refer to **"Working with eTL Sample Projects" on page 72**.

5 To download the eWay documentation and JavaDocs, repeat steps 2 to 4 for each eWay.

6 Restart Enterprise Designer.

2.2 After Installation

Once you have installed eTL Integrator, the first step is to create an eTL Project in Enterprise Designer. The next chapters describe how to create eTL Projects and Environments and how you build the business logic for eTL Business Processes and Collaborations.

Chapter 3

Building and Deploying eTL Projects

This chapter describes how to build and deploy eTL Projects.

In This Chapter:

3.1 Quick Start Guide to Building eTL Projects

This section provides a quick overview of the overall process of building and deploying eTL Projects.

1. Create a Project as described in **"Creating eTL Projects" on page 18**.
2. Create the flat file and/or database OTDs as described in **"Creating Flat File OTDs" on page 18** and **"Creating Database OTDs" on page 24**.
3. Create the eTL Collaboration Definition as described in **"Creating eTL Collaboration Definitions" on page 24**.
4. Create the Business Logic for the Business Processes or Collaborations as described in **"Building Business Logic for eTL Projects" on page 36**.
5. For eInsight users only, create the eTL Business Process as described in **"Creating Business Processes (eInsight Only)" on page 27**.

6 Create a Connectivity Map as described in **"Creating eTL Connectivity Maps" on page 29**.

7 Add the Project components to the Connectivity Map as described in **"Adding Connectivity Map Components" on page 29**.

8 Link the Connectivity Map components as described in **"Linking Connectivity Map Components" on page 31**.

9 Configure the external applications as described in **"Configuring the External Applications" on page 32**.

10 Create the eTL Collaboration Definition as described in

11 Create an Environments as described in **"Creating the ICAN Environment" on page 33**.

12 Create and activate the Deployment Profile as described in **"Creating the Deployment Profile" on page 34**.

3.2 Creating eTL Projects

To create eTL Projects

- In the **Project Explorer** tab of the Enterprise Designer, right-click the Repository, click **New Project**. This adds a new Project under the Repository.

To rename the Project, right-click it, click **Rename**, and enter a new name.

3.3 Creating Flat File OTDs

This section describes how to create flat file OTDs using the Flat File OTD wizard. During the creation of the OTD you specify the names of the input and output flat files. These flat file must be a small example of the actual flat files that are used for input and output. Generally, the actual files are too large to use while designing the OTD.

To create flat file OTDs

1 In the **Project Explorer** tab of the Enterprise Designer, right-click the flat file Project, click **New**, and click **Object Type Definition**. The **New Object Type Definition** dialog box appears as shown below.

Figure 3 Creating Flat File OTDs

2 Click **Flat file** and click **Next**.

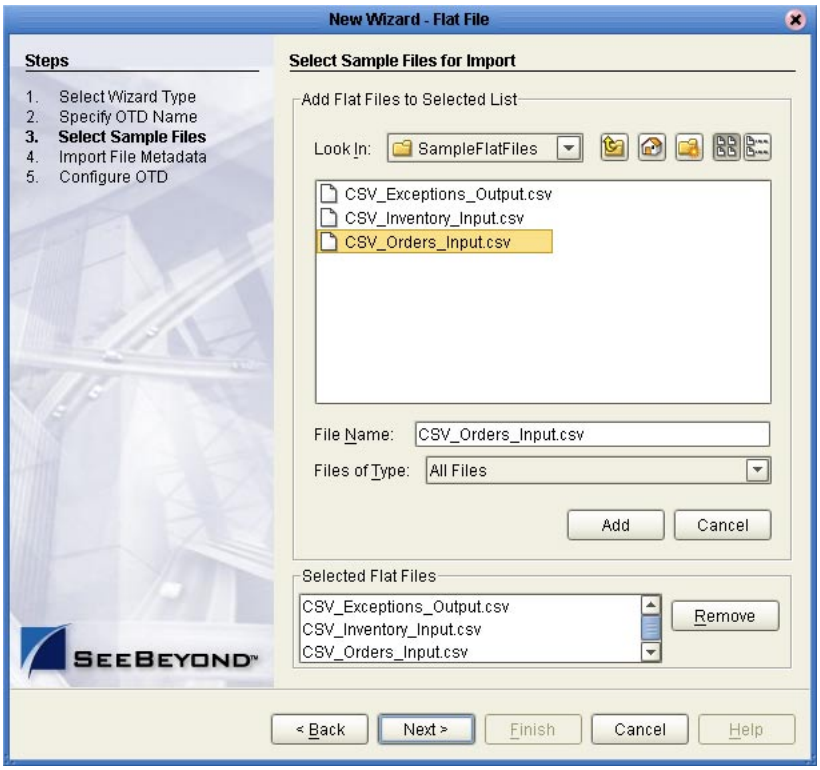
3 In the **New OTD Name** box, enter the name for the flat file OTD and click **Next**. The **Select Sample Files for Import** page appears.

4 In the **Look In** box, navigate to the folder where the sample input and output flat files are located. You must have write permission to the files.

Note: *The sample files must be small examples of the actual input and output files; the Flat File OTD wizard cannot handle the large size of actual flat files.*

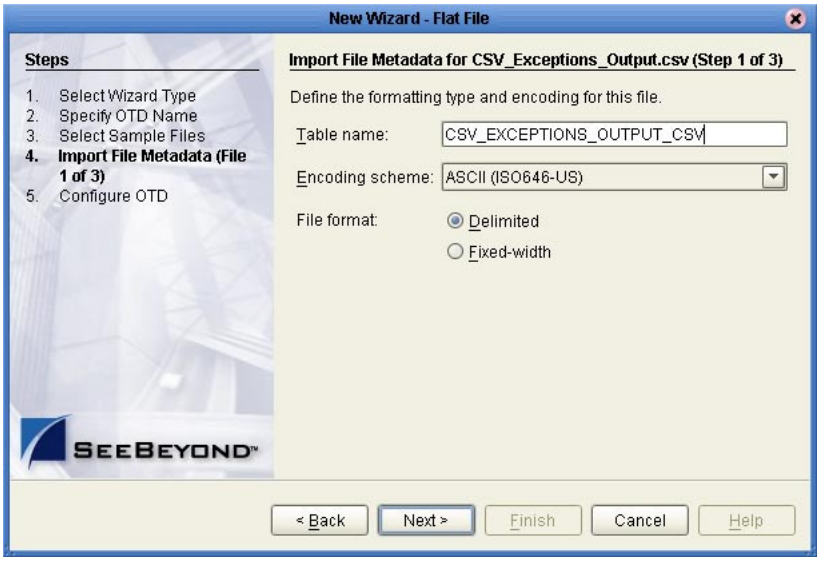
5 Double-click the files that this Project is to use as its input or output files. You can add multiple flat files. This adds the files to the **Selected Flat Files** box.

Figure 4 Selecting Input and Output Files



6 Click **Next**. The **Import File Metadata (Step 1)** page appears.

Figure 5 Importing File Metadata—Step 1



7 Enter the following settings and click **Next**:

For This Option	Enter
Table name	The name of the table
Encoding scheme	ASCII
File format	**Delimited** to indicate that this flat file is delimited format or **Fixed width** to indicate this flat file is fixed width format

The next page that appears depends on whether you a selected delimited or fixed width format. For a delimited file format, continue with the next step. For a fixed file format, go to step 9.

8 For a delimited file format, enter the following settings to indicate how to parse the flat file and click **Next**:

This Setting	Indicates	Options
Default SQL type	The default SQL type used for all elements in the flat file OTD structure	varchar char date float integer time timestamp
Record delimiter	The delimiter that separates flat file records	newline (lf) carriage return: cr cr-lf
Field delimiter	The delimiter that separates fields in flat file records	comma tab pipe

This Setting	Indicates	Options
Text qualifier	The qualifier used to indicate text	none double quote: “ single quote: ‘
First line contains field names	Whether the names specified in the header row are used as element names of the OTD (**True**) or whether eTL Integrator will assign the initial names dynamically to the OTD elements (**False**)	True False

9 For a fixed-width file format, enter the following settings to indicate how to parse the flat file and click **Next**:

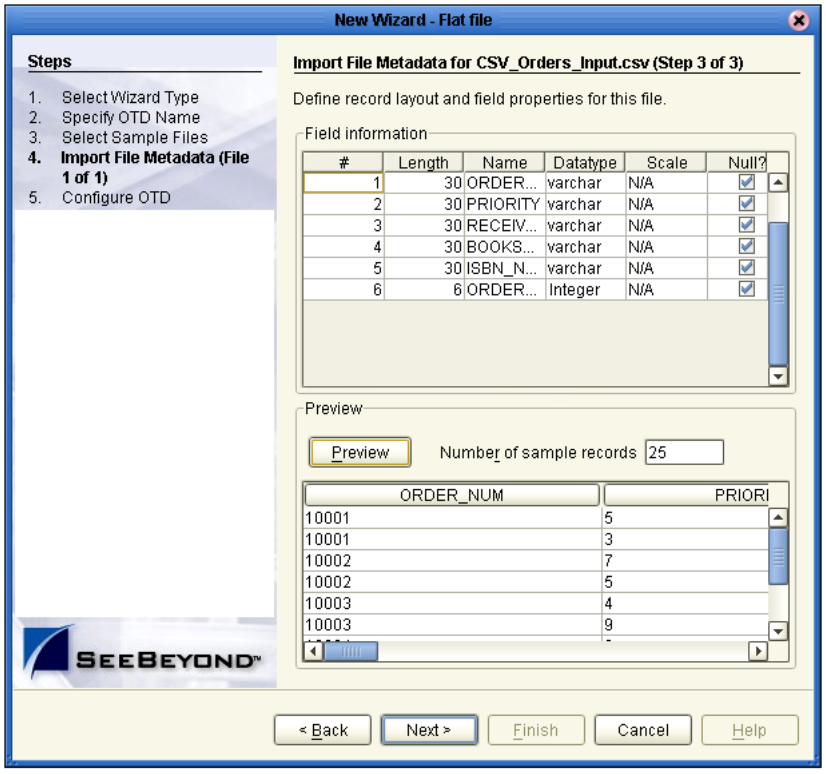
This Setting	Indicates	Options
Default SQL type	The default SQL type used for all elements in the flat file OTD structure	varchar char date float integer time timestamp
Record length	The maximum length of a record in number of characters	< 0 (must be appropriate for the selected data type and must be the same for all fields)
Field count	The number of fields per record	< 1 (must be the same for all fields)
Record delimiter	The delimiter that separates flat file records	newline (lf) cr cr-lf
First line contains field names	Whether the names specified in the header row are used as element names of the OTD (**True**) or whether eTL Integrator will assign the initial names dynamically to the OTD elements (**False**)	True False
Offset to start record (bytes)	The number of bytes to skip before reaching start of the first record; value is ignored if the **First line contains field names** setting is set to **True**	<= 0

10 In the **Import File Metadata (Step 3)** page, change the record layout and field properties by double-clicking the fields below and entering the correct information if necessary.

This Setting	Indicates	Options
Length	The maximum number of digits to use in representing a number field; for example, the number 9876.543 has a precision of 7	< 0 (must be appropriate for the selected data type) and must be the same for all records in fixed-width formats
Name	The name of the field	unlimited number of characters
Datatype	The datatype of the record	<varchar char date float integer time timestamp
Scale	The number of digits to the right of the decimal point in a number field; for example, 9876.543 has a scale of 3	< 0
Null	Whether the field can be null	Select (the field can be null) or deselect (the field cannot be null)

11 To see a preview of a number of sample records, enter the number of records to preview in the **Number of sample records** box and click **Preview**.

Figure 6 Defining Record Layout for Orders Input

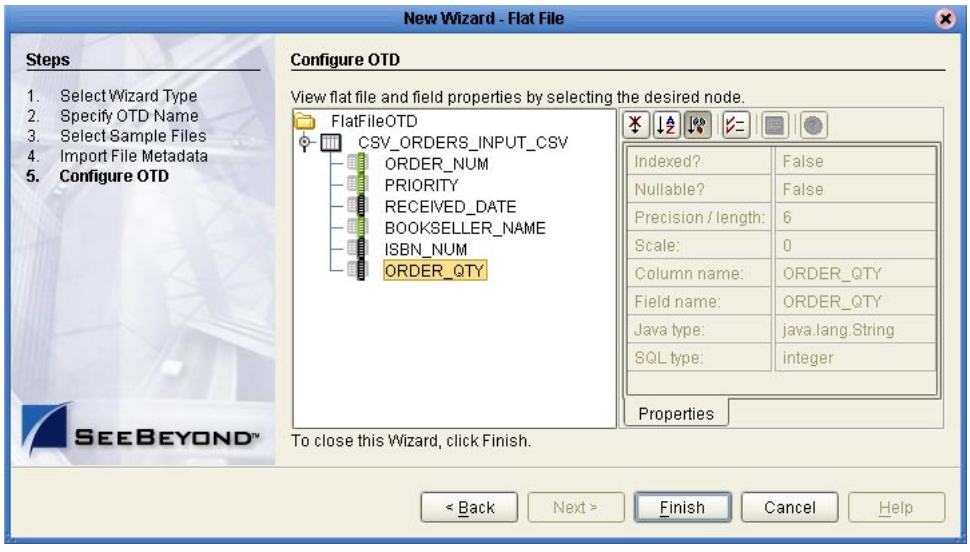


12 Click **Next**. The **Configure OTD** page appears.

If you have to define additional flat files, the **Import File Metadata (Step 1)** page appears. Return to step 7.

13 Expand the OTD and the flat file, and click each field to verify that the record properties are correct.

Figure 7 Verifying Flat File Record Properties



Records are marked with either green or black icons. A green icon indicates that a field is nullable. Records marked with black icons cannot contain a null value.

14 Click **Finish**. The new OTD displays in the **Project Explorer** tab in the Enterprise Designer.

3.4 Creating Database OTDs

eTL Integrator supports several database types as described in **"Supported Databases" on page 14**. You must have the eWay for the particular database type installed. For instructions on creating an OTD for a database, refer to the eWay user's guide for that database. For a list of eWay documents, refer to **"Supporting Documents" on page 7**.

3.5 Creating eTL Collaboration Definitions

Once you have created the OTDs for the eTL Project, you create the eTL Collaboration Definitions as described in the procedure below. The eTL Collaboration Definition defines the source tables, the source files to be joined, and the target tables.

To create eTL Collaboration Definitions

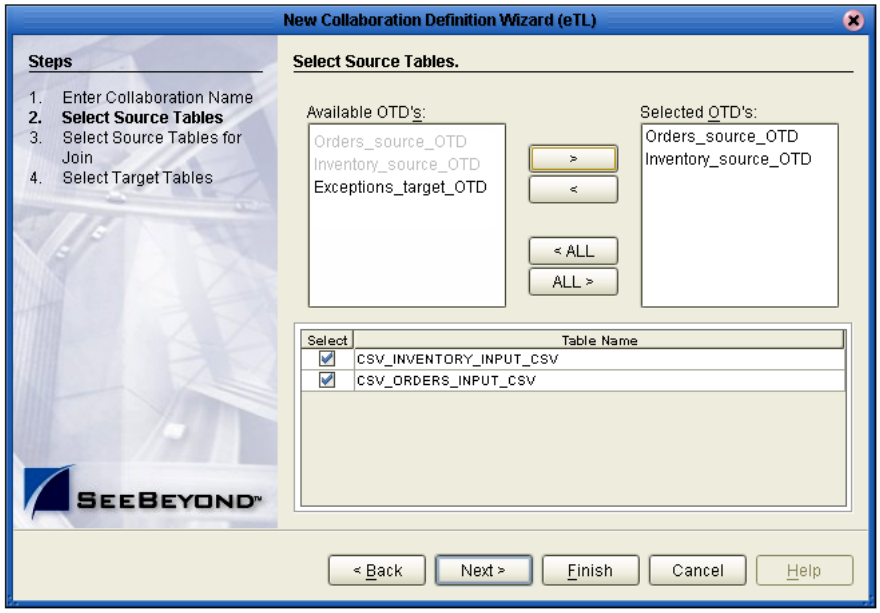
1 Right-click the eTL Project, click **New**, and click **Collaboration Definition (eTL)**. The **New Collaboration Definition Wizard (eTL)** dialog box appears.

2 In the **New Collaboration Name** box, enter a name for the Collaboration and click **Next**. The **Select Source Tables** page appears.

3 Click the OTDs that contain the source tables in the **Available OTDs** box and click the right arrow to move them to the **Selected OTDs** box.

4 Under **Select**, click the tables to be used as source tables. The figure below shows an example of selected source tables.

Figure 8 Selecting Source Tables



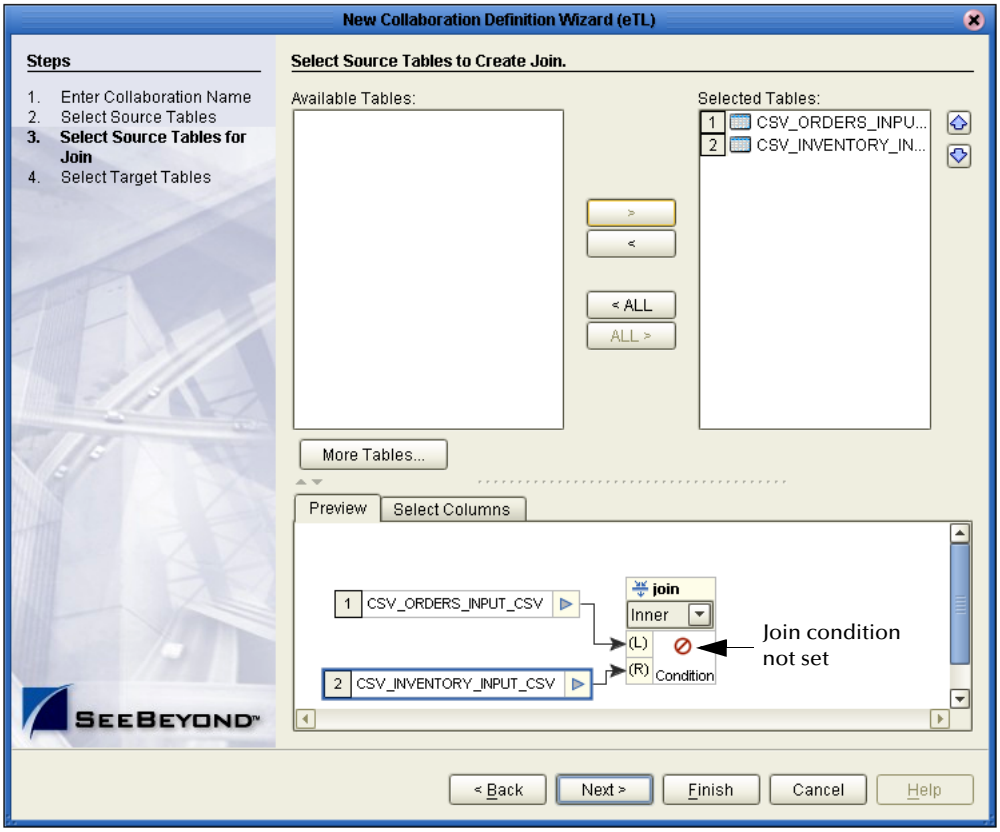
5 Click **Next**. The **Select Source Tables for Join** page appears (if there are multiple tables).

6 Click the tables in the **Available Tables** box and click the right arrow to move them to the **Selected Tables** box.

The **Preview** box shows the join between the tables, usually with the join condition not set as indicated below.

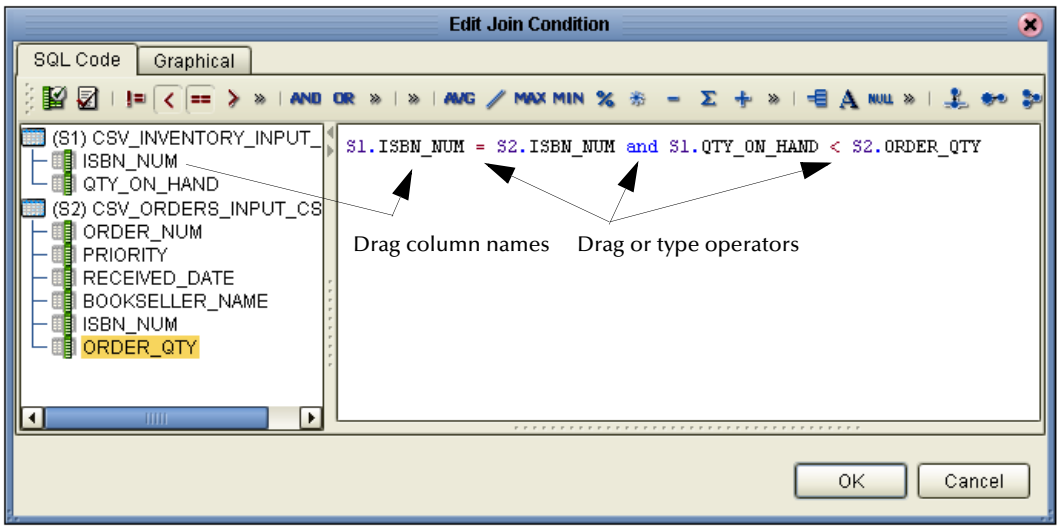
If eTL detects a join relationship between two tables in the same database, it automatically sets the join condition as indicated by the icon.

Figure 9 Joining Source Tables



7 To set the join condition, double-click the ⊘ icon. The **Edit Join Condition** dialog box appears.

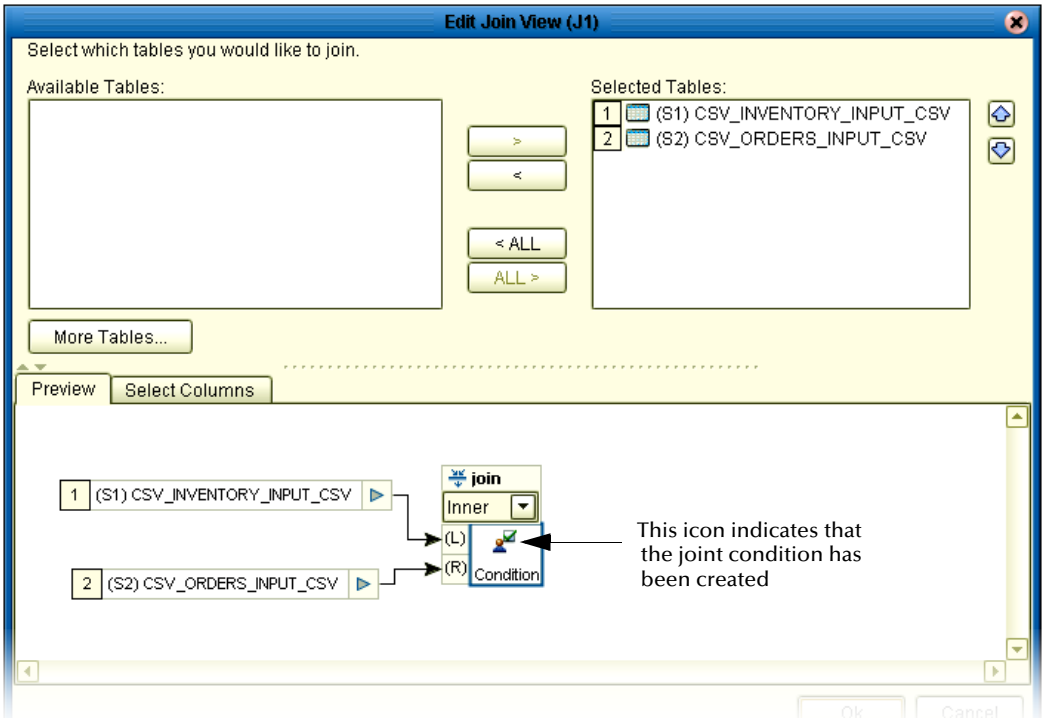
Figure 10 Specifying Join Conditions



8 Drag element names and drag or type operators into the SQL text line as shown in the figure above.

9 Click **OK**. The **Edit Join View** dialog box appears.

Figure 11 Viewing the Join



10 Click **OK** again to close the **Edit Join View** dialog box.

11 In the **Select Source Tables for Join** page, click **Next**. The **Select Target Tables** page appears.

12 Click the OTDs that contain the target tables in the **Available OTDs** box and click the right arrow to move them to the **Selected OTDs** box.

13 Under **Select**, click the tables to be used as target tables.

14 Click **Finish**. This completes the Collaboration Definition and adds the Collaboration under the eTL Project in the **Project Explorer** tab.

3.6 Creating Business Processes (eInsight Only)

If you are using eGate Integrator with eInsight Business Process Manager, use the procedure below to create the eTL Business Process.

To add Connectivity Map components

1 In the **Project Explorer** tab of the Enterprise Designer, right-click the eTL Project, click **New**, and click **Business Process**.

2 To rename the Business Process, right-click it, click **Rename** and enter a new name.

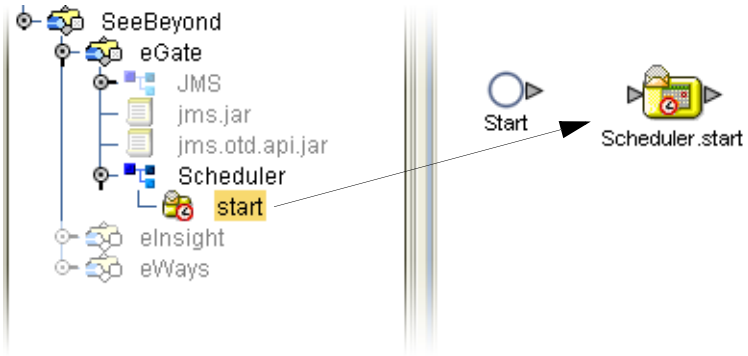
3 Click and expand the **SeeBeyond** services.

4 Click and expand your **eGate** Service.

5 Click and expand your **Scheduler** Service.

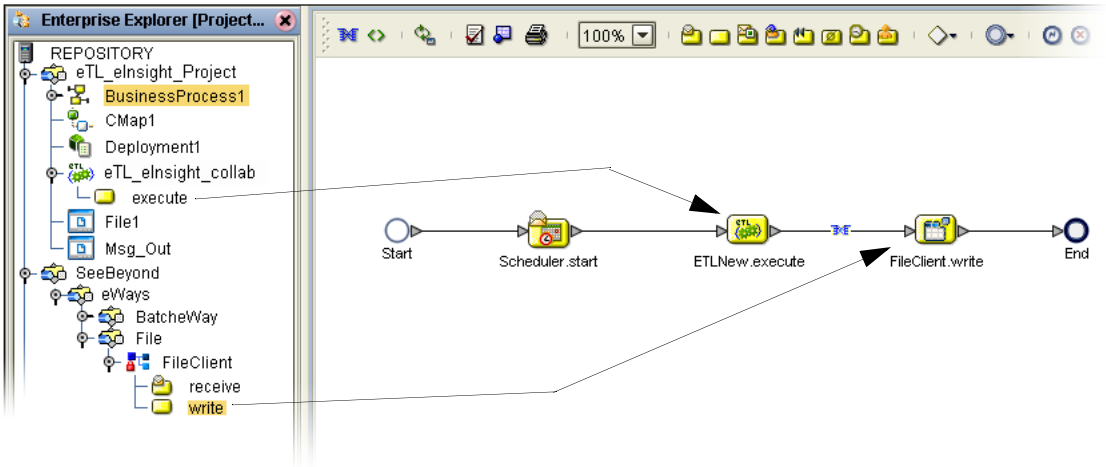
6 Click and drag **start** to the Business Process designer pane.

Figure 12 Scheduler for Business Process



7 Drag in the other Services as shown in the following figure.

Figure 13 Business Process Map



8 Drag the eTL Collaboration's subfolder **execute** to the designer map and place it just to the right of the **FileClient.receive** Service.

9 Drag the **Write** Service to the map and place it to the right of the eTL Collaboration.

10 Connect the nodes to all components from Start to End.

11 Click **Save**.

To create a File Client Write Business Rule

1 Add a Business Rule between the eTL Collaboration and the **FileClient.write** Service. This rule formats the output for the status and count messages.

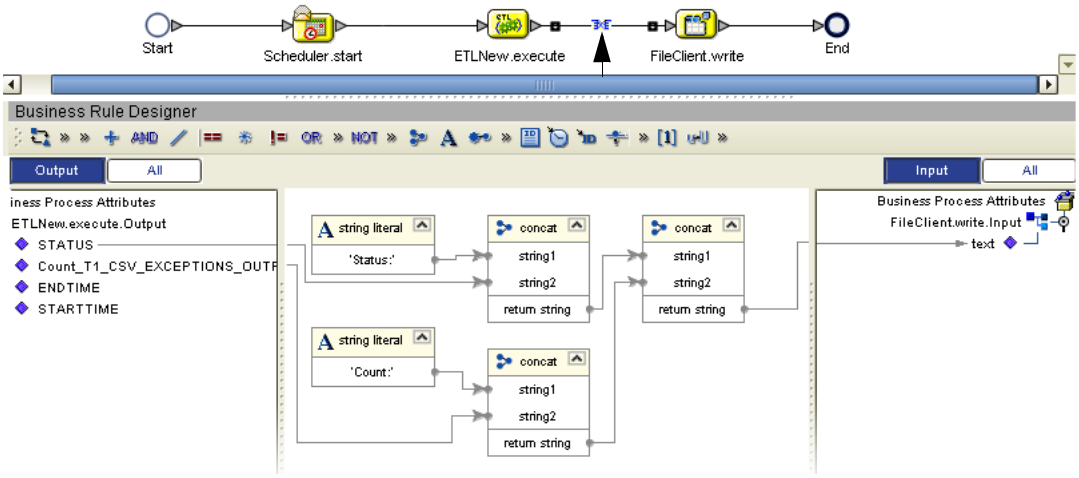
2 Right-click the node between the **eTL Collaboration Service** and the **FileClient.write Service** and select Add Business Rule.

3 Double-click the Business Rule icon to display the Business Rules Designer pane.

You will create the business rules by dragging operators to this work space.

4 Connect the nodes in the Business Rule Designer as shown in Figure 14 to create the Business Rule.

Figure 14 Business Rule



3.7 Creating eTL Connectivity Maps

To create eTL Connectivity Maps

- In the **Project Explorer** tab of the Enterprise Designer, right-click the eTL Project, click **New**, and click **Connectivity Map**. This adds a new Connectivity Map under the eTL Project.

To rename the Connectivity Map, right-click it, click **Rename** and enter a new name.

3.8 Adding Connectivity Map Components

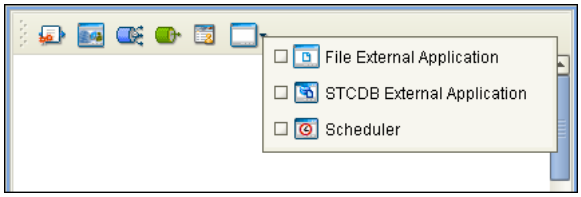
To add Connectivity Map components

1 In the **Project Explorer** tab of the Enterprise Designer, click the Connectivity Map for the eTL Project.

2 Add necessary components such as schedulers and Services.

3 To add a flat file application, click the **External Applications** icon and click **STCDB External Application**.

Figure 15 Selecting the Flat File Application

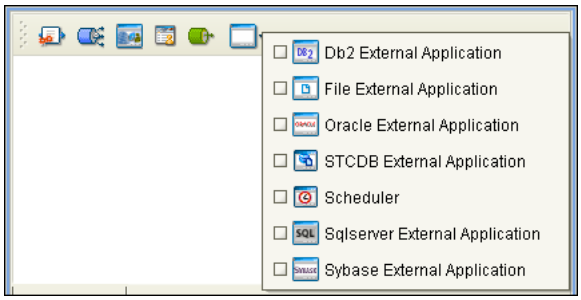


4 Drag the STCCB icon from the Connectivity Map toolbar onto the Connectivity Map canvas.

5 To add eTL supported and installed eWay components to the Connectivity Map, click the **External Applications** icon, click any of following options, and drag the icon to the canvas:

- **Db2 External Application** for DB2 Universal Database eWays
- **File External Application** for File eWays
- **Oracle External Application** for Oracle eWays
- **SqlServer External Application** for SQL Server eWays
- **Sybase External Application** for Sybase eWays

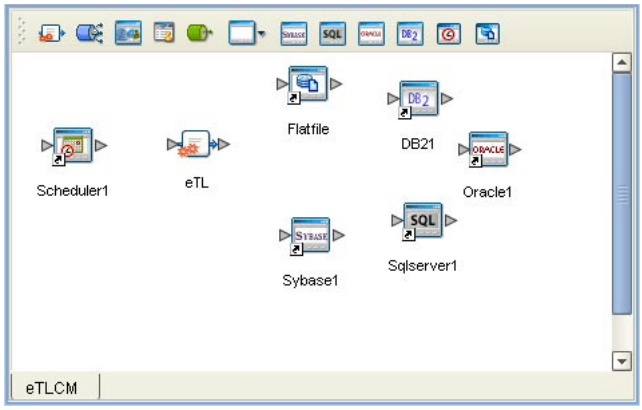
Figure 16 Selecting Supported eWay Components



6 To rename an external application, right-click the icon, click **Rename**, and enter a new name.

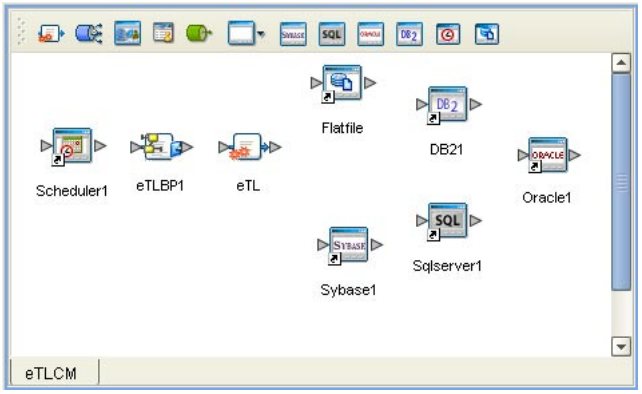
The figure below shows an example of a Connectivity Map for an eTL Project.

Figure 17 Connectivity Map for an eTL Project



The figure below shows an example of a Connectivity Map for an eInsight eTL Project.

Figure 18 Connectivity Map for an eInsight eTL Project

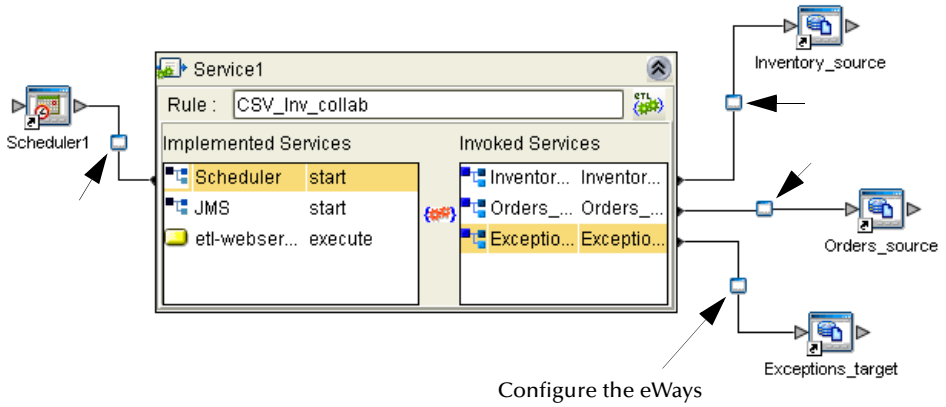


3.9 Linking Connectivity Map Components

To link Connectivity Map components

1 Connect the nodes as shown in the following figure.

Figure 19 Linking Connectivity Map Components



2 Double-click the **eWay** between the Scheduler and the Service to set the Scheduler parameters. In this scenario set the frequency in minutes to 2.

3 Double-click the other three **eWays** to set the data paths. Enter the path to the directory where you placed your sample CSV files.

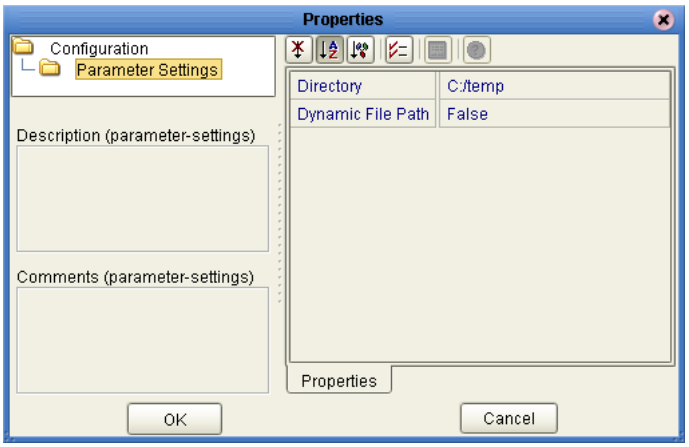
3.10 Configuring the External Applications

You configure the logical properties the eWays and flat file application in the Connectivity Map. Physical properties are configured in the ICAN Environment. For information about configuring the supported eWays, refer to the eWay documentation. The procedure below describes how to configure the STCDB external application used for flat file OTDs.

To configure STCDB external applications

4 In the Connectivity Map for the eTL Project, double-click the small eWay icon for the STCDB application. The **Properties** dialog box appears as shown below.

Figure 20 Configuring STCDB Applications



5 In the **Directory** box, enter the location of the flat files.

6 To use a dynamic file path, set the **Dynamic File Path** property to True. This property is disabled by default. For information about using dynamic file paths, refer to **Using Dynamic Flat File Names** on page 40.

7 Click **OK**.

3.11 Creating the ICAN Environment

To create the ICAN Environment

1 In the **Environment Explorer** tab of the Enterprise Designer, right-click the Repository and click **New Environment**.

2 Right-click the Environment and click **New Logical Host**.

3 Right-click the Logical Host and click **New SeeBeyond Integration Server.**

4 To add a flat file external application, right-click the Environment, click **New STCDB External System**, enter the name for the STCDB application and click **OK**.

5 To add database eWays, right-click the Environment, click any of the database external systems as applicable to your Project design, enter the component name, select inbound or outbound, and click **OK**. For detailed information, refer to the eWay documentation. The following eWays are supported:

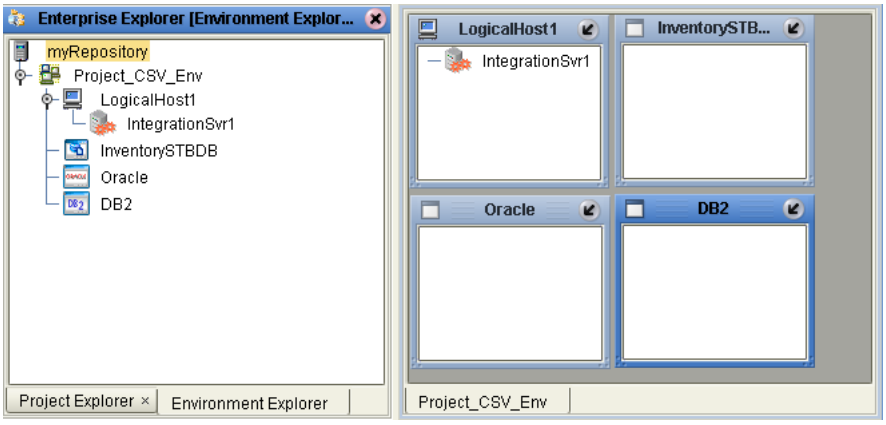
- Inbound/Outbound Oracle, Sybase, and SQLServer eWay (not XA)
- Inbound/Outbound DB2 eWay for Windows/UNIX

6 Configure the Logical Host, Integration Server, and database eWays by right-clicking the component in the **Environment Explorer** tab and clicking **Properties**.

7 Enter the properties and click **OK**. For information about eWay configurations, refer to the eWay documentation. For information about Logical Host and Integration Server configurations, refer to the eGate documentation.

The figure below shows a sample Environment for an eTL Project.

Figure 21 Sample ICAN Environment for an eTL Project

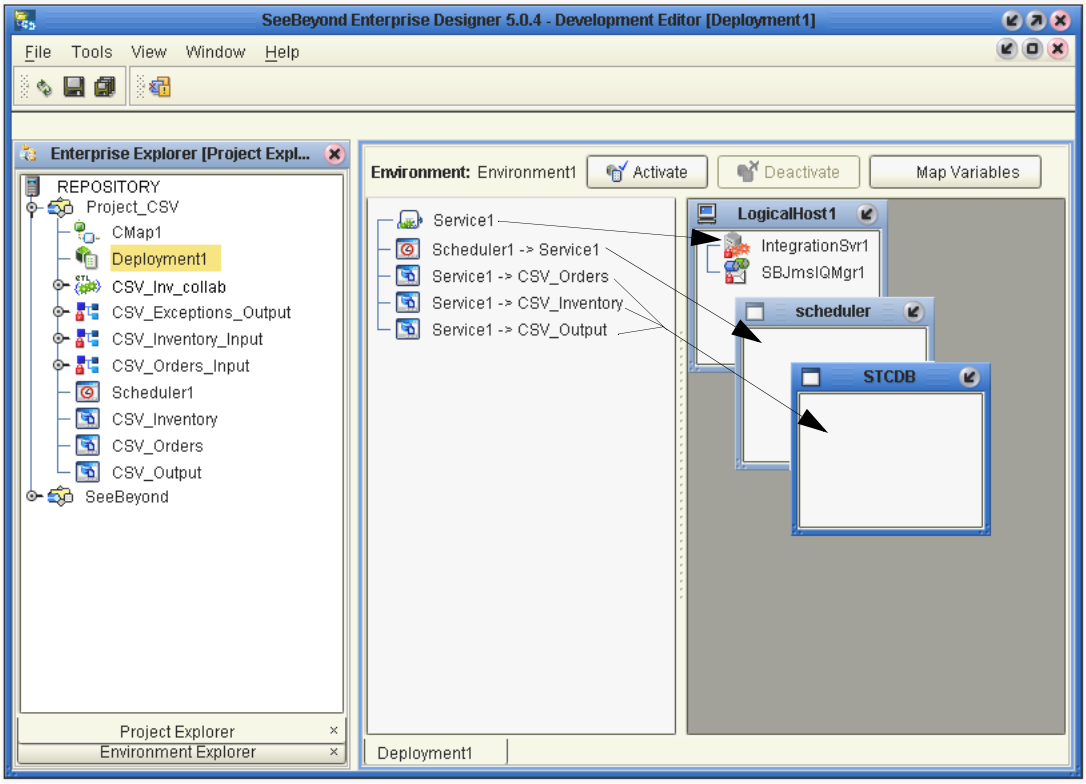


3.12 Creating the Deployment Profile

To create the Deployment Profile

1. In the **Project Explorer** tab of the Enterprise Designer, right-click the eTL Project and click **New Deployment Profile**.
2. Enter the name for the Deployment Profile, click the Environment you created for the Project, and click **OK**.
3. Drag the Project components to the Environment component as shown in the figure below.

Figure 22 Creating the Deployment Profile



For instructions on activating deployments and running Projects, refer to the *eGate Integrator User's Guide*. For an example, refer to **"Working with eTL Sample Projects" on page 72**.

Chapter 4

Building Business Logic for eTL Projects

This chapter describes how to build transformations for eTL Projects. Project business logic is contained in Business Processes for eInsight, and in Collaborations for eTL Integrator used without eInsight.

To build data transformations, you use the eTL Editor to edit the OTDs. For information about creating OTDs, refer to the previous chapter.

In This Chapter

4.1 Using the eTL Editor

After you have created the OTDs for the eTL Project as described in the previous chapter, you use the eTL Editor in the Enterprise Designer to create the data transformations. This section describes the eTL Editor toolbar.

4.1.1 About the eTL Editor Toolbar

The table below describes each button on the eTL Editor toolbar.

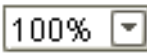
Table 1 eTL Toolbar Icons—Standard

	
	**Undo/Redo** Undo a previous action. For example, if you delete an operator, click Undo to restore it. (The Redo icon does the opposite.)
	**Test Run Collaboration** Executes your Project and generates a message log. The log shows messages and errors if the execution fails.

Table 1 eTL Toolbar Icons—Standard (Continued)

	**Validate Collaboration** Validates your mapping logic without executing your Project.
	**Edit Database Properties** Opens a dialog box that allows you to edit data source and target properties.
	**Select Source and Target Tables** Opens a window from where you can select OTDs and source or target tables to use in a Collaboration.
	**Toggle Output View** Toggles between a full screen Collaboration editor pane and a pane divided to show output messages: data from Logs, validations and output data.
	**Create/ Edit Join** Creates a new join view that permits the direct entry of source table relationships (joins), including the ability to view multiple joins. The new join dialog is automatically invoked when one of the following events occurs: 1. The user maps a specific source table column (s1) to the target table (t) and the system detects that the target table (t) was already mapped to a different source table (s2) which was not joined to the first source table (s1). 2. The user connects a specific source table column (s1) to an operator (o), such as 'concatenate', and the system detects that this operator (o) was already connected to a different source table (s2) which was not joined to the first source table (s1).
	**Edit Join View** Displays the Edit Join view (as explained in the row above) after the join view has already been created. Right-click in the header of a joined table view and select 'Edit Join View'.
	**Add/Edit Runtime Inputs** Input variables that are assigned by an external system, such as eInsight, are called Runtime Inputs. See **Using Runtime Arguments** on page 38.
	**Add/Edit Runtime Outputs** Output variables generated by an eTL Collaboration which can be used by an external system, such as eInsight, are called Runtime Outputs. See **Using Runtime Arguments** on page 38.
	**Expand all Graph Icons** Expands the GUI display to show all mapping elements and fields. This is the default view.
	**Collapse all Graph Icons** Collapses the GUI display to make the Collaboration designer pane less cluttered. For example, if the view is collapsed when tables are displayed, only the headers appear with the mapping arrows.
	**Auto Layout all Graph Icons** Automatically arranges all the graphical elements in your Collaboration designer pane.

Table 1 eTL Toolbar Icons—Standard (Continued)

	**Print Graph** Prints the graphic from the Collaboration designer pane. You may achieve the best print results by selecting "print using view's currently selected scale" or "Scale to fit page," from the drop-down print options.
	**Scale** Choose the size of the GUI display. This selection could also affect print scale.

4.2 Data Extraction

This section describes how to build data extraction logic to extract data from multiple, heterogeneous, and disparate sources.

4.2.1 Filtering Data

Using Runtime Arguments

The eTL runtime arguments explained in this section offer various ways to extract and filter data. Runtime arguments, **runtimeInput** and **runtimeOutput** are used with eTL, but the **runtimeOutput** requires an interface to eInsight.

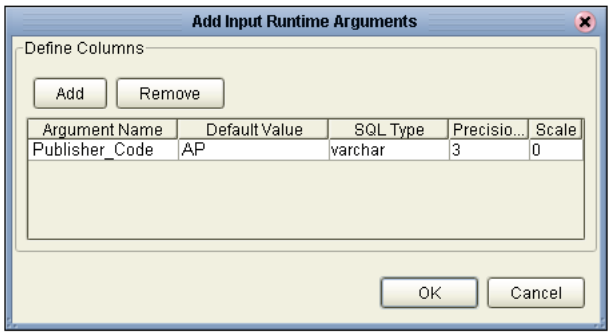
You can use runtime arguments to select and filter data. In the following example enter the variable 'AP' to select the **PUBLISHER_CD** (Publisher code) Adamson Publishing.

To use runtime input arguments

1 In the eTL Editor, click **Add/Edit Runtime Inputs**  .

The **Add Input Runtime Arguments** window appears.

Figure 23 Adding Input Runtime Arguments



2 Name the argument **Publisher_Code**.

3 Enter **AP** as the default value.

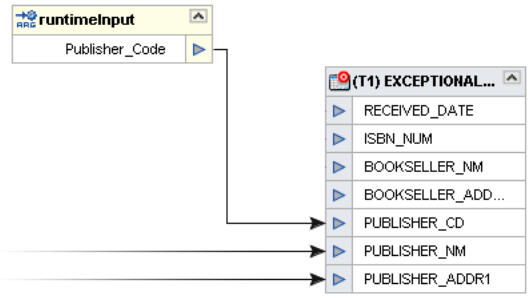
4 Select **Varchar**.

5 Type the maximum length required in the **Precision** field.

6 Click **OK**.

Refer to the following figure.

Figure 24 Map Runtime Input Argument



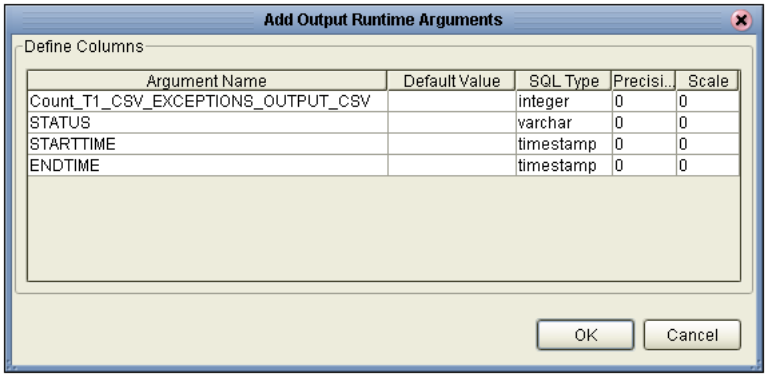
The runtime argument (filter) shown is used to capture only records for 'AP' (Adamson Publishing).

Set Runtime Output Argument

Runtime output arguments  are used to return counts, status and timestamps as explained in this section. For example, you can use the count parameter for a condition in an eInsight business process.

See the following figure.

Figure 25 Output Runtime Arguments



Output Runtime Arguments can be captured and displayed or written to a file. These "messages" are made available automatically by the system.

- Count - is the row count for INSERT, UPDATE, or DELETE, depending on your selected target table operation.

- STATUS - is either “successful” or “failure”
- STARTTIME - is the start time of the eTL process
- ENDTIME - is the end time of the eTL process

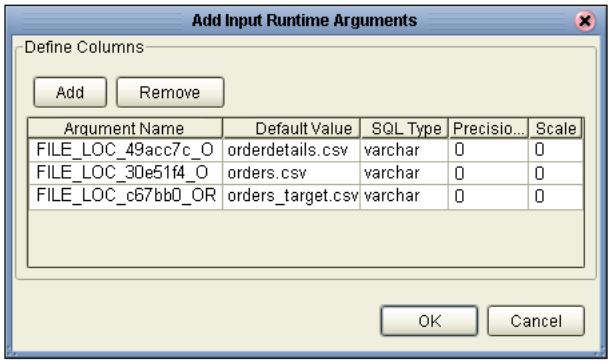
Using Dynamic Flat File Names

This feature enables the passing of file names from an eInsight Business Process. Flat file names can be passed to an eTL Collaboration for specific source and target table(s) at runtime. The file name can be passed by either an eInsight Business process or based on a flat file OTD’s absolute flat file name.

Runtime Parameters—Dynamic Absolute Names

Runtime parameters (see **“Using Runtime Arguments” on page 38**) are also used for dynamic file names. Refer to the following figure which shows absolute names.

Figure 26 Add Input Runtime Arguments—Dynamic Absolute Names



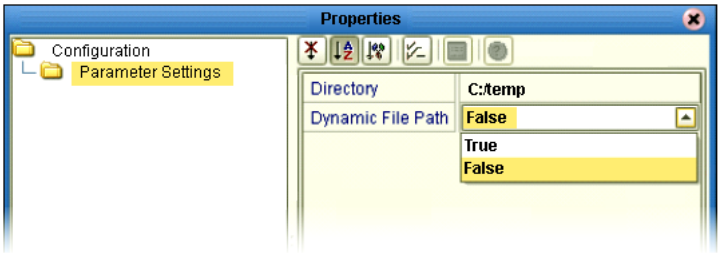
Runtime Parameters—Dynamic Relative Names

You may pass relative dynamic flat file names by specifying the path in the Connectivity Map when you set the Properties for the file eWay.

In the figure below a relative name (path) is shown. This relative path is ignored if an absolute file name is passed at runtime.

To invoke this property click Expert Properties on the Properties dialog. See **“Setting Source Table Expert Properties” on page 47**.

Figure 27 Add Input Runtime Arguments—Dynamic Relative Names



In the figure above a **True** setting indicates a fully qualified path/file name. (In the figure above the path and file name are not fully qualified).

- In the figure above a **False** setting tells the system to use the file and director set during the creation of the OTD along with the directory specified in the Connectivity Map eWay. The path shown in the figure is relative.

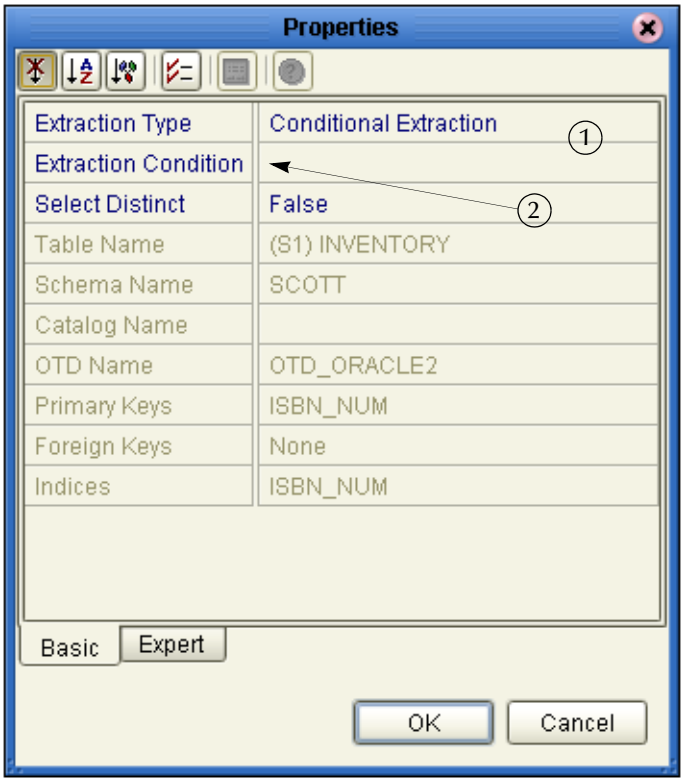
Using Conditional Extractions

The eTL Condition Builder is another tool for filtering and selecting data for extraction. Use Conditional Extractions to select and filter data from one or more selected source tables. Setting a condition with Condition Builder to filter a source table is called an "extraction condition." Setting a condition with Condition Builder to filter a target table is called a "condition." The following steps demonstrate a source table extraction condition:

To use source table extractions

1 Right-click the source table and click **Properties**. The **Properties** dialog box appears.

Figure 28 Source Table Properties



2 Click inside the Extraction Condition field (number 2 in the previous figure).

3 Click the ellipsis (...)

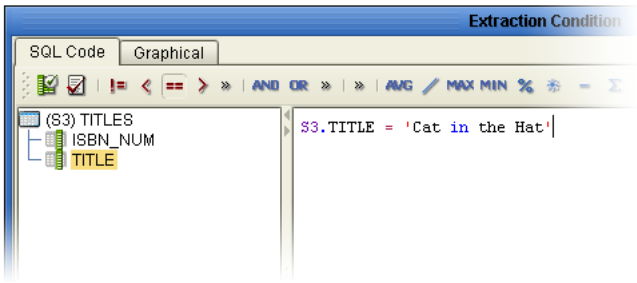
The Extraction Condition screen appears.

4 Drag **S3.TITLE** to the SQL code line.

5 Drag or type the '**equal**' operator.

6 Type '**Cat in the Hat**'.

Figure 29 Extraction Condition—Source



You can edit the Extraction Condition at any time.

There are two options for the Extraction Type (number 1 in In Figure 28.

- **Full Extraction**

  No filtering is done.

- **Conditional Extraction**

  Filtering is based on the conditions you set. Typically, you will select Conditional Extraction.

Using Subqueries

To include a subquery in one of your source tables, you can create a view including one or more subqueries in one of your source environments.

See the subquery below.

```
select *
from Orders s2
where not exists
   (select s1.qty_on_hand
   from Inventory s1
   where s1.qty_on_hand < 100
   and s1.isbn_num = s2.isbn_num)
```

To use subqueries

1 Implement the subquery inside your SQL statement as a database view.

After you create this view, create an OTD and use that OTD in an eTL Collaboration.

2 Create an OTD using the **Object Type Definition Wizard** to select your view to use in the eTL Collaboration.

3 Create an eTL Collaboration, using the OTD you created, and select your view as source table.

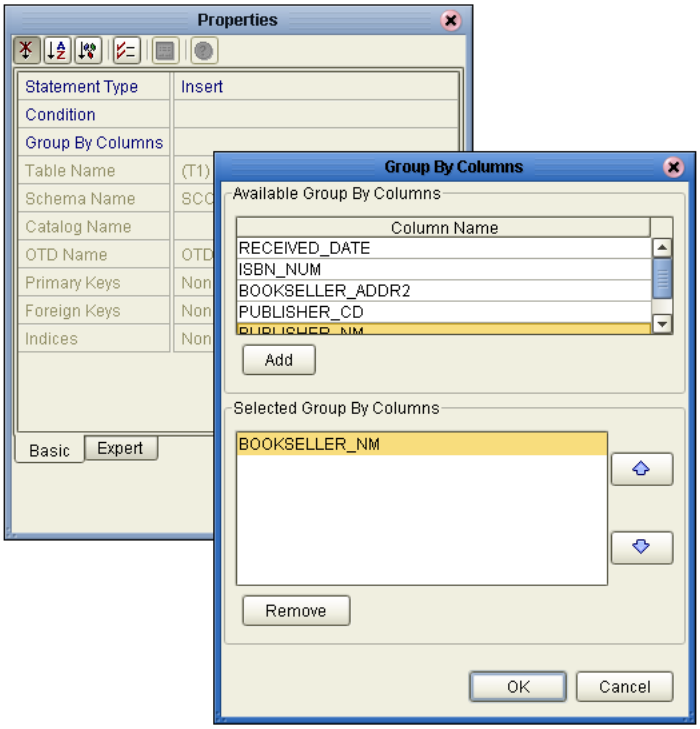
The view you create is used as your source and can be used the same as any source table.

Using Data Aggregations

The eTL product supports extracting aggregated data and applying special transformations and load to a target table. Specific transformations are supported for aggregated values such as minimum, maximum and average.

You can group by column based on a selection you can make from the **Group By Columns** condition on the Properties dialog box.

Figure 30 Group by Column



To select column names

In the example shown in the previous figure the columns are grouped by Bookseller name, i.e. BOOKSELLER_NM.

1. Right-click your target table and select **Properties**.
2. Select a **Column Name** from the list.
3. With the **Group By Columns** displayed, click **OK**.

   Refer to the previous figure.

Note: *In order to use the Group By Columns feature, your join must have at least one equal operator between columns of participating join tables. In addition, at least one aggregate function needs to be mapped to target column(s).*

4.2.2 Setting Join Conditions

When you attempt to link a source table to a target table, that is already joined to a different source table, you are prompted to create a new join view.

This feature serves two purposes:

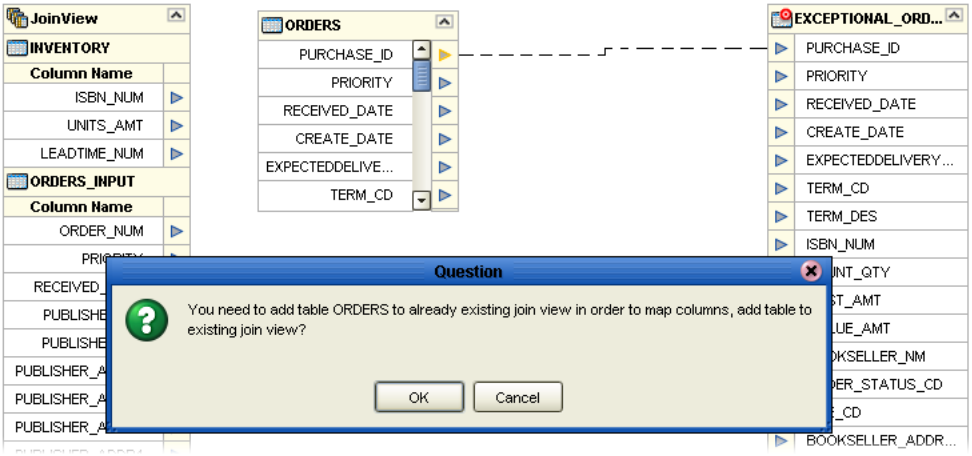
- It affords quick access to the set join condition interfaces
- It serves as a warning message if you don't really want to create a new join view

The following steps are an example only.

To create a join view using the Link Time Join

1 Attempt to connect a node from source table **ORDERS.PURCHASE_ID** to the target **EXCEPTIONAL_ORDERS.PURCHASE_ID**. The following message appears.

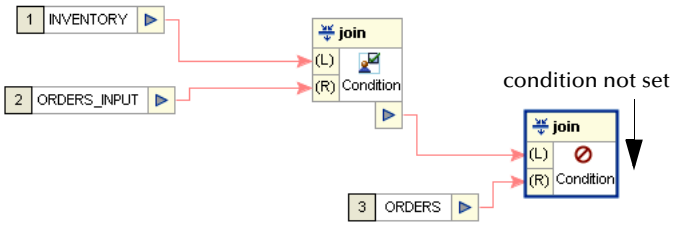
Figure 31 Adding Tables to Existing Views



2 Click **OK** to indicate your intention to add the **ORDERS** table to the existing join view.

A graphical view of the join appears. You can set the conditions based on the establishment of relationships between sources explained below.

Figure 32 Joined View



3 Click the icon/button shown in the previous figure.

Add the join conditions. See **Setting Join Conditions** on page 44.

4.2.3 Using Table Aliases with Multiple Source Table Views

You can use table aliases (S1, S2, S3, etc.) to map and load a target table from multiple instances of the same source table. Consider the following example using the following source tables: **EMP_TBL** and **CODES_TBL**. You can create a join view with these tables, then you can drag another view of the **CODES_TBL** to your Designer Pane to do a third join which will be used in the code lookup.

The following table displays the sample data for the **EMP_TBL** source table:

Table 2 Employee Table

NAME	ID	JOB CODE	DEPT CODE
Dave	1	P	D1
Judy	2	C	D2

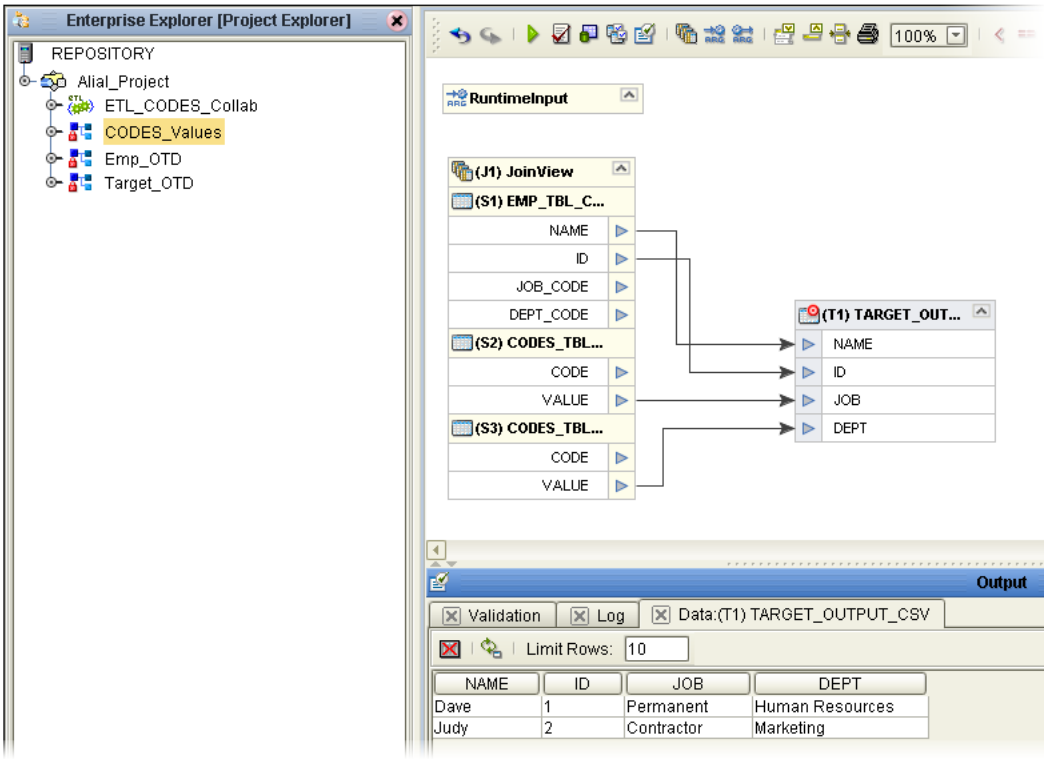
The following table displays the sample data for the **CODES_TBL** source table:

Table 3 Company Codes

CODE	VALUE
D1	Human Resources
D2	Marketing
P	Permanent
C	Contractor

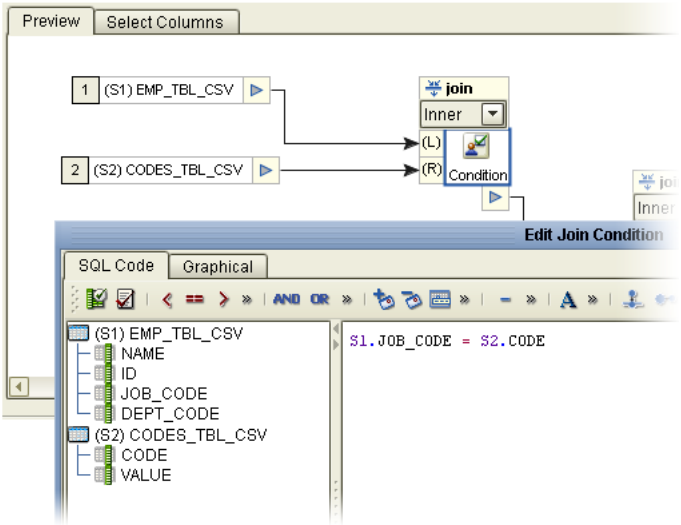
The following figure shows the Collaboration and mapping with the correct data from a test run. The lookup loads the description for both jobs and departments from the **CODES_TBL** table.

Figure 33 Employee and Codes Example



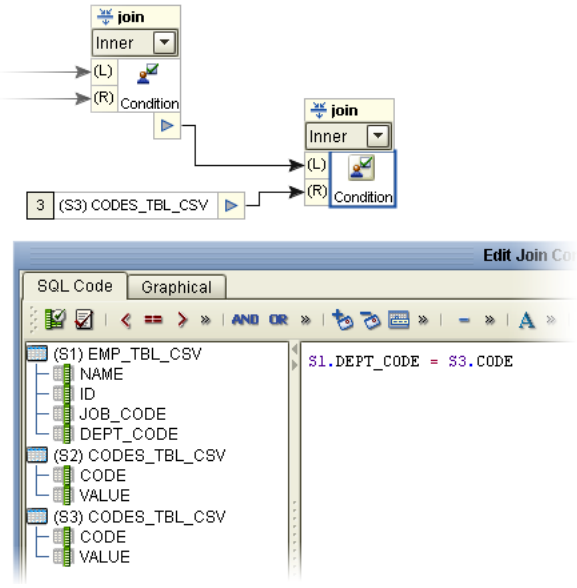
Refer to the first join shown in the following figure. This join view shows the condition **S1.JOB_CODE = S2.CODE**. This will load the job descriptions from the **CODES_TBL** to the target table column **JOB**.

Figure 34 Employee and Codes Example—First Join



Refer to the second join shown in the following figure. This join view shows the condition **S1.DEPT_CODE = S3.CODE**. This loads the department descriptions from the **CODES_TBL** to the target table column **DEPT**.

Figure 35 Employee and Codes Example—Second Join



4.2.4 Setting Source Table Expert Properties

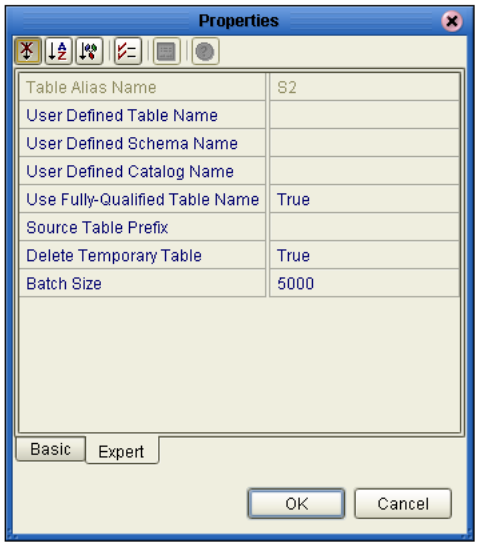
Expert Properties allow flexibility in naming tables, schemas, and catalogs. You can also tune performance by adjusting batch sizes. You also have the option to delete a temporary table or create a table. You can set an option to truncate a target table before loading.

To use Expert Properties, right-click on a table and click the Expert tab.

Source Table Expert Properties

The Expert property settings for the source table(s) are explained in this section.

Figure 36 Source Expert Properties



User Defined Table Name

You have the option to substitute a table name instead of the name derived from the OTD.

User Defined Schema Name

You have the option to substitute a schema name instead of the name derived from the OTD.

User Defined Catalog Name

You have the option to substitute a catalog name instead of the name derived from the OTD.

Use Fully-Qualified Table Name

Set to 'True' to use a fully qualified name, i.e. **catalog_name.schema_name.table_name**.

Source Table Prefix

You have the option to append a source table name prefix, if the name is defined, instead of the name derived from the OTD.

Delete Temporary Table

Set to 'True' to delete a temporary table if one is created during the extraction process.

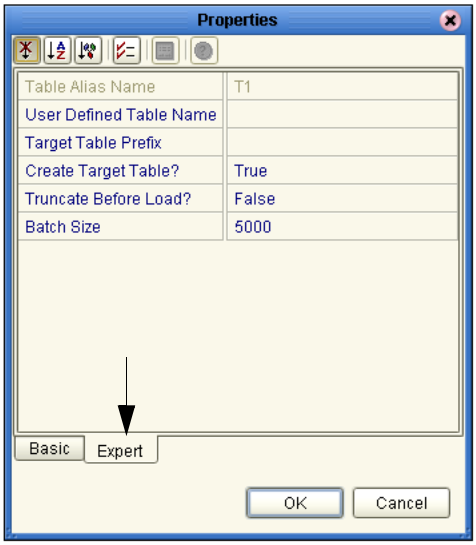
Batch Size

You can configure source table batch sizes to improve eTL Collaboration execution performance. By tuning the batch size you can load data more efficiently into source tables.

By default, 5000 rows are populated at the same time into a source table. There is no upper limit. The limit is determined by the amount of internal memory of the local host.

Generally, the lower the number the better, but adjust the value to determine the optimum performance.

Figure 37 Properties—Set Batch Size



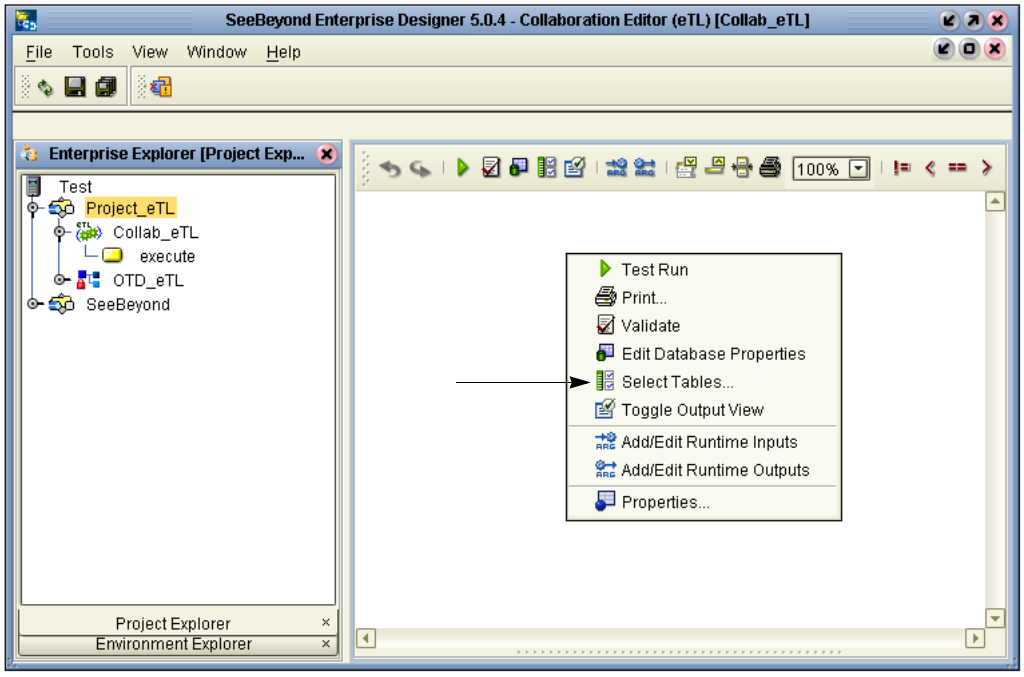
4.2.5 Optional Method for Selecting Tables

You can also select tables from an empty editor pane in your **eTL Collaboration Editor** window.

To select tables

1 In the eTL Editor, right-click the empty editor pane and click **Select Tables**.

Figure 38 Options Menu



The **Source and Target Tables** dialog appears.

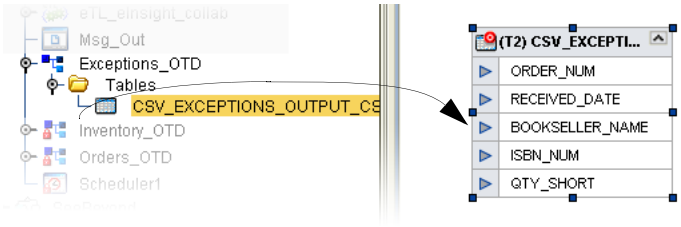
2 Click **Finish** in the **Source and Target Tables** dialog box.

The table appears in the designer pane.

To drag and drop from Project Explorer

1 Click the table in the Project Explorer pane and drag to the designer pane.

Figure 39 Dragging and Dropping Tables



4.3 Data Transformations

4.3.1 About Data Transformation Operators

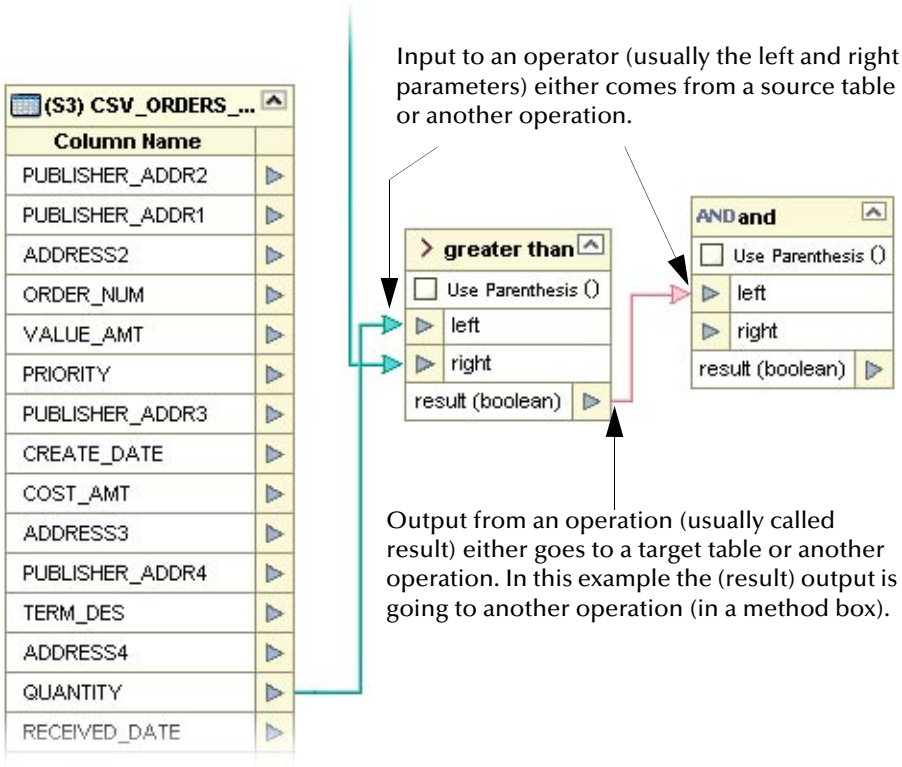
This section describes the following data transformation operators in the eTL Editor.

- **"About Comparison Operators" on page 52**

Table Input and Output Logic

The input to an operator (usually the *left* and *right* parameters of a method box) either comes from a source table or another operation. Output from an operation (usually the *result* parameter of a method box) either goes to another operation or to the target table. The following figure illustrates an example:

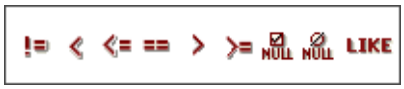
Figure 40 Table Input and Output Logic

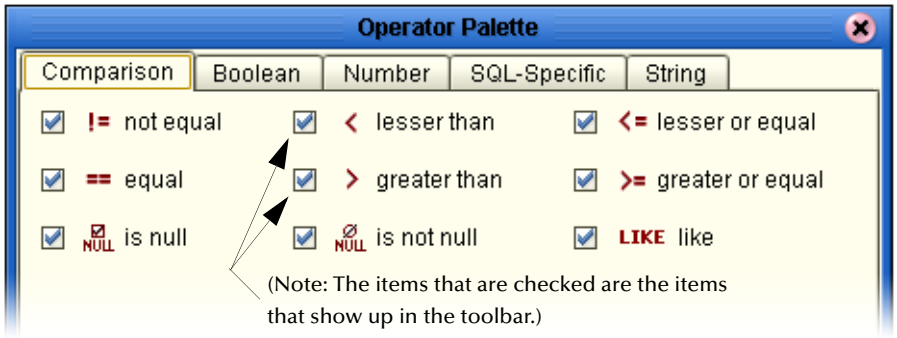


About Comparison Operators

The Comparison operators are explained in the following table. The operators are used within method boxes.

Table 4 Comparison Operators





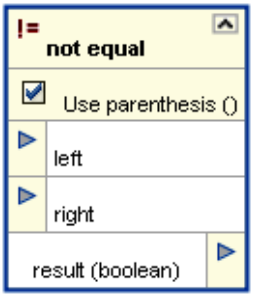
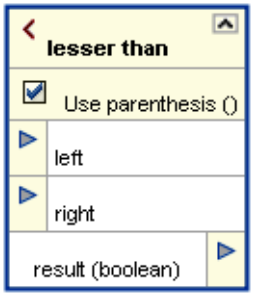
Method Box	Description/Usage
	**not equal** The **not equal** operator returns true if the data column mapped to the *left* property is not equal to the *right* property; otherwise, returns false.
	**lesser than** The **lesser than** operator returns true if the data value mapped to the *left* property is less than the data value mapped to the *right* property; otherwise, returns false.

Table 4 Comparison Operators (Continued)

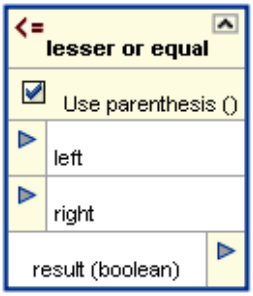
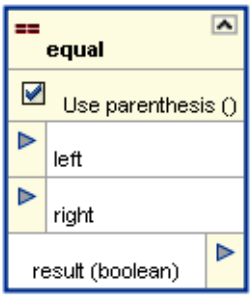
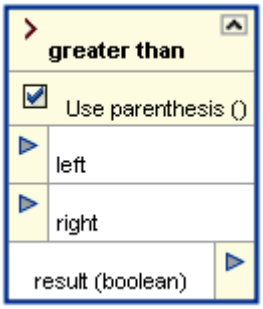
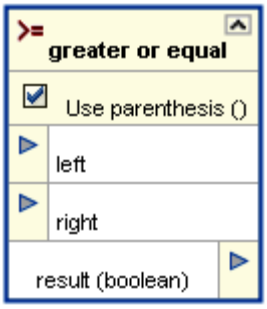
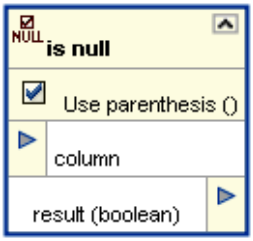
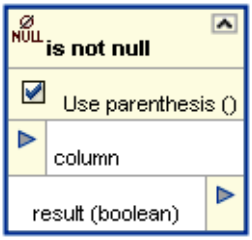
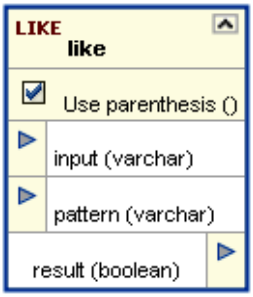
	**lesser or equal** The **lesser or equal** operator returns true if the data value mapped to the *left* property is less than or equal to the data value mapped to the *right* property; otherwise, returns false.
	**Equal** The **equal** operator returns true if the data value mapped to the *left* property is equal to the data value mapped to the *right* property; otherwise, returns false.
	**greater than** The **greater than** operator returns true if the data value mapped to the *left* property is greater than the data value mapped to the *right* property; otherwise, returns false.

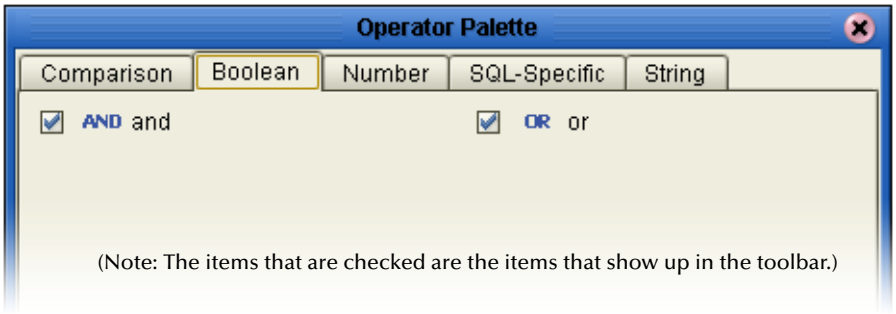
Table 4 Comparison Operators (Continued)

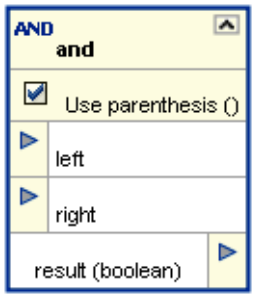
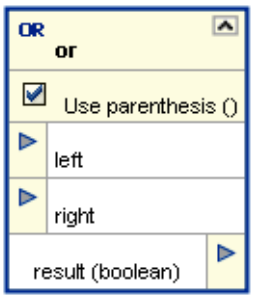
	**greater or equal** The **greater or equal** operator returns true if the data value mapped to the *left* property is greater than or equal to the data value mapped to the *right* property; otherwise, returns false.
	**is null** The **is null** operator returns true if the input *column* value is null; otherwise, returns false.
	**is not null** The **is not null** operator returns true if the input *column* value is not null; otherwise, returns false.
	**LIKE** The **LIKE** operator returns true if the *input (varchar)* value matches the mapped *pattern (varchar)* value within a string; otherwise, returns false. See **Using the Case and LIKE Operators** on page 67.

About Boolean Operators

The Boolean operators are explained in the following table. The operators are used within method boxes.

Table 5 Boolean Operators

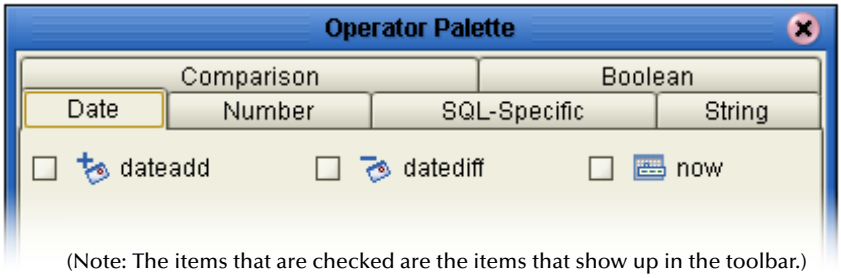


Method Box	Description/Usage
	**AND** The **AND** operator returns true if both the data value mapped to the *left* property and the data value mapped to the *right* property are true; otherwise, returns false.
	**OR** The **OR** operator returns true if either the data value mapped to the *left* property or the data value mapped to the *right* property is true; otherwise, returns false. (If both *left* and *right* are false, returns false.)

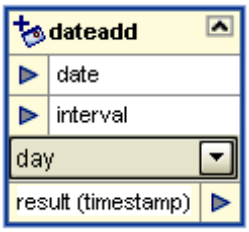
About Date Operators

The Date operators are explained in the following Table 4. The operators are used within method boxes.

Table 6 Date Operators



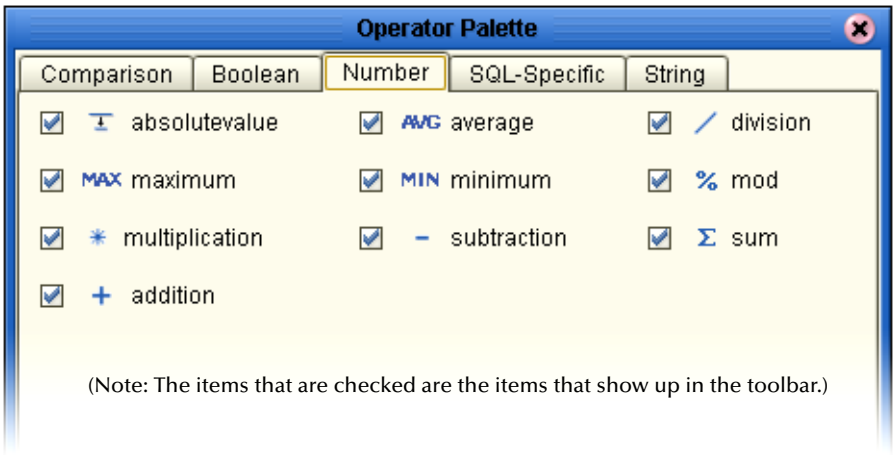
(Note: The items that are checked are the items that show up in the toolbar.)

Method Box	Description/Usage
	**datediff** The **datediff** operator calculates the difference between two dates. The resulting format is numeric. Click the drop-down arrow to select an option, such as days, weeks, months, etc. See **Calculating Dates** on page 65.
	**dateadd** The **dateadd** operator adds a user-selected date interval in days, weeks, months, etc. to the date field and return a new date. The resulting date format is determined by the date format of the target table's database. Click the drop-down arrow to select an option, such as days, weeks, months, etc. See **Calculating Dates** on page 65.
	**now** The **now** operator returns a date plus time in timestamp format. It is the current date and time. The format depends on the database. For example, in DB2 the format is: yyyy-mm-dd hh:mm:ss See the sample in **Figure 44 on page 66**.

About Number Operators

The Number operators are explained in the following Table 7. The operators are used within method boxes.

Table 7 Number Operators



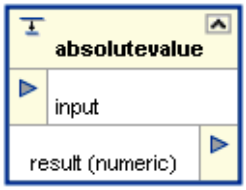
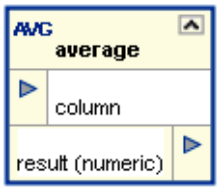
Method Box	Description/Usage
	**absolutevalue** The **absolutevalue** operator returns the value of a number without regard to a its algebraic sign. *input* is a number that is positive or negative; the *result(numeric)* is the absolute value without regard to sign.
	**Average** The **average** operator calculates the average (or the mean) of numeric data for a mapped data value (column) and returns the *result (numeric)* either to another operator or to a column in a target table.

Table 7 Number Operators (Continued)

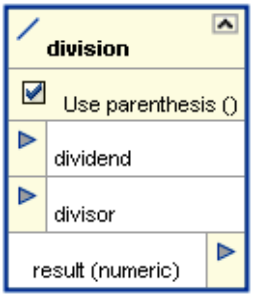
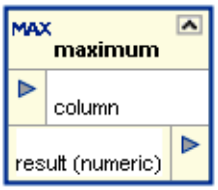
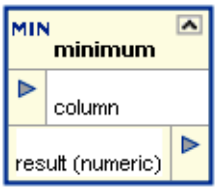
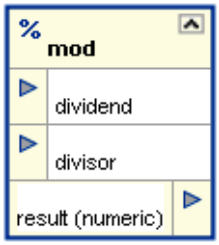
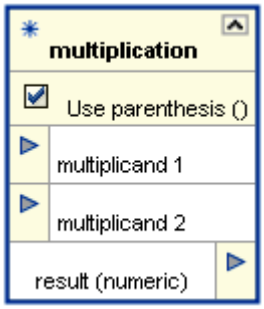
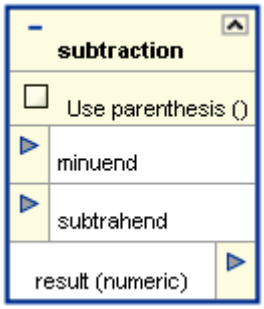
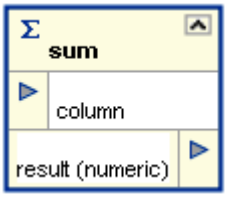
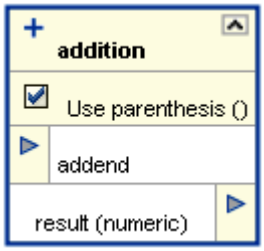
	**Division** The **division** operator divides the value of *dividend* by the value of *divisor* and returns the quotient *result (numeric)*.
	**Maximum** The **maximum** operator determines the largest value in a *column* and returns the *result (numeric)*.
	**Minimum** The **minimum** operator determines the smallest value in a *column* and returns the *result (numeric)*.
	**Mod** The **modulus** operator divides the numerical value of *dividend* by the numerical value of *divisor* and returns the remainder *result (numeric)*.
	**Multiplication** The **multiplication** operator multiplies the value of *multiplicand1* by the value of *multiplicand2* and returns the product *result (numeric)*.

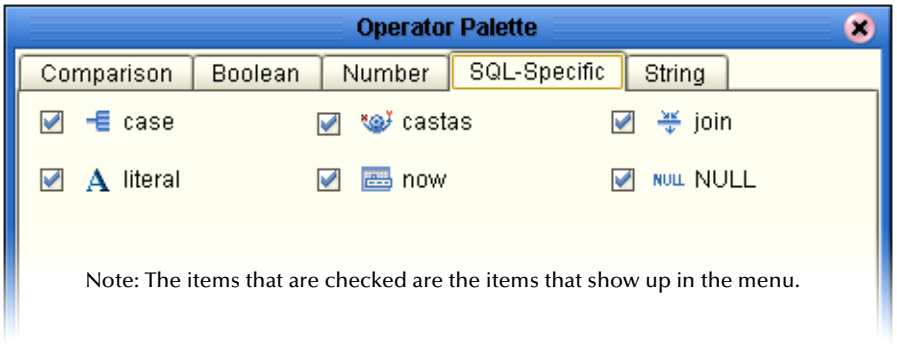
Table 7 Number Operators (Continued)

	**Subtraction** The **subtraction** operator subtracts the numerical value *subtrahend* from the numerical value of *minuend* and returns the difference *result (numeric)*. *minuend* is the number from which another number is subtracted. *subtrahend* is the number to be subtracted from another number.
	**Sum** The **sum** operator adds all the values in numeric *column*. The *result (numeric)* is the sum.
	**Addition** The **addition** operator adds the value of two input nodes which both map to the *addend* node. The *result (numeric)* is the sum.

About SQL Operators

The SQL operators are explained in the following Table 8. The operators are used within method boxes.

Table 8 SQL Operators



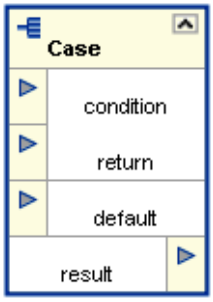
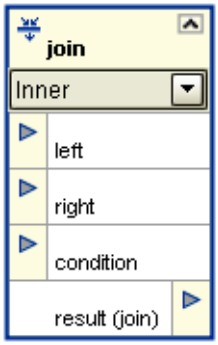
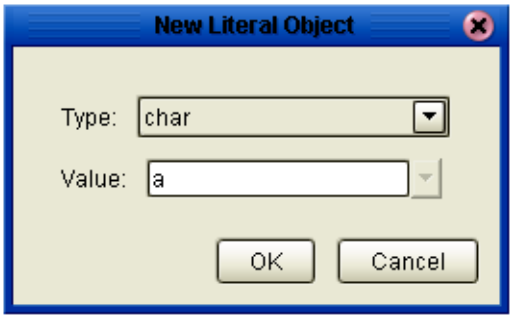
Method Box	Description/Usage
	**case** The **case** operator filters or selects data using *If (WHEN)*, *THEN* and *ELSE* logic. The case operator consists of multiple WHEN statements and one default statement. Each WHEN statement has one *condition* and one *return* input. The result value depends on the following: WHEN statements are evaluated in sequence until a *true condition* is met. When a *condition* is *true*, the *return* input is entered as the *result*. If none of the statements are *true*, the *default* value is the *result*. The graphic to the left shows only one WHEN statement. To add additional WHEN statements, right-click the **case** dialog box and select Add New When. See **Using the Case and LIKE Operators** on page 67.
	**castas** The **castas** operator is used to change the data type of data. For example, you could change numeric data to variable characters. Input to *column* is directly from a data value (for example a table column) and not a condition. The result is converted to char, double, float, integer, timestamp or varchar. The results of this operation could be unpredictable and the data could be truncated.

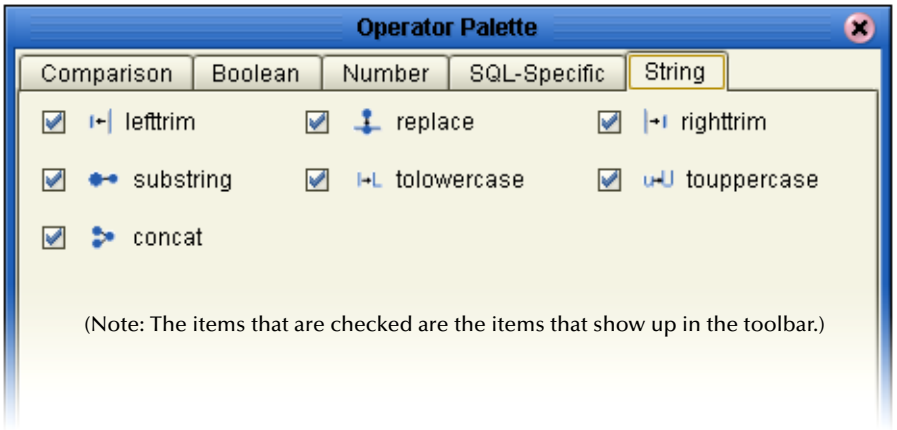
Table 8 SQL Operators (Continued)

	**join** (See **Setting Join Conditions** on page 44. The Join operator is an alternate method of joining tables.) Use the **join** operator to join two tables at one time. Example: Two tables may be joined, unconditionally by connecting a node from the table header arrows to the *left* and *right* input links of the **join** operation; no condition is required. When two tables are joined based on a condition, there is an input node from a conditional expression to *condition*. If you want to link multiple tables, *result (join)* can be input to another **join** operator.
 	**Literal** Use to set a literal value which may be a character, double precision, float, integer, timestamp or variable character. 1. Select a Type from the drop-down menu. 2. Enter a value. The Literal node appears.
	**now** The **now** operator is used to place a timestamp in a field in the default format of the source table.
	**NULL** The **NULL** operator is used to place a null value in another operator or target table.

About String Operators

The String operators are explained in the following Table 9. The operators are used within method boxes.

Table 9 String Operators



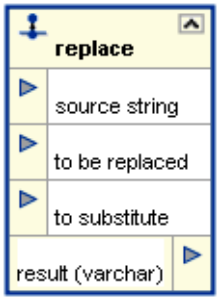
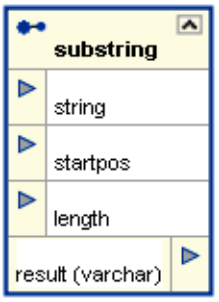
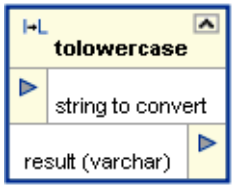
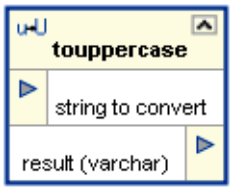
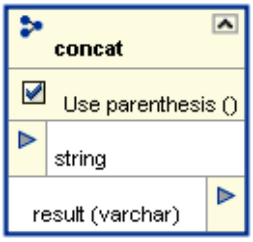
Method Box	Description/Usage
	**lefttrim** The **lefttrim** operator removes leading spaces from the left end of a string.
	**replace** The **replace** operator replaces (or substitutes) a string of characters with a string of your choosing. The *source string* is the input string. The *to be replaced* string is the entire string or part of the string that is replaced. The *to substitute* string is the actual new value replaces the *to be replaced* string. The *result (varchar)* is the output string. For example: If the **source string** = 'STC Corporation' and to be replaced = 'STC' and **to substitute** = 'SeeBeyond' the **result (varchar)** would = 'SeeBeyond Corporation'.

Table 9 String Operators (Continued)

	**righttrim** The **righttrim** operator removes trailing spaces from the right end of a string.
	**substring** The **substring** operator returns the substring of the first argument *string* starting at the position specified in the second argument *startpos* (starting position) which is a literal integer. The length specified in the third argument *length* is also a literal integer and represents the character length of the first argument. The *result (varchar)* is the resulting, shortened output string. For example: *string* 020 with a *startpos* of 2 and a *length* of 2 would *result* in 20.
	**tolowercase** The **tolowercase** converts the characters in input string to lower case characters. The *result string (varchar)* is all lower case.
	**touppercase** The **tolowercase** converts the characters in input string to upper case characters. The *result string (varchar)* is all upper case.
	**concat** Use to Link or concatenate two or more fields. Values are concatenated in the order they are connected to the *string*. Multiple nodes may be connected to the *string*. Note: see **Applying Parentheses to Operators** on page 63.

4.3.2 Applying Parentheses to Operators

Order of precedence is dependent on operators and the use of parenthesis. This is especially important for the proper execution of mathematical calculations.

The following is a precedence example in Oracle:

"ETLPOC"."EMP"."SID" + "ETLPOC"."EMP"."ID" * 10

By default multiplication has precedence over addition, so the multiplication "ETLPOC"."EMP"."ID" * 10 would be evaluated first then added to "ETLPOC"."EMP"."SID" as shown in Figure 41.

(See the arrows pointing to the check boxes in the following two figures.)

By using “use parenthesis” the addition can have precedence as shown Figure 42.

(("ETLPOC"."EMP"."SID" + "ETLPOC"."EMP"."ID") * 10)

"ETLPOC"."EMP"."SID" + "ETLPOC"."EMP"."ID" is evaluated first then multiplied by 10.

Figure 41 Multiplication Has Precedence

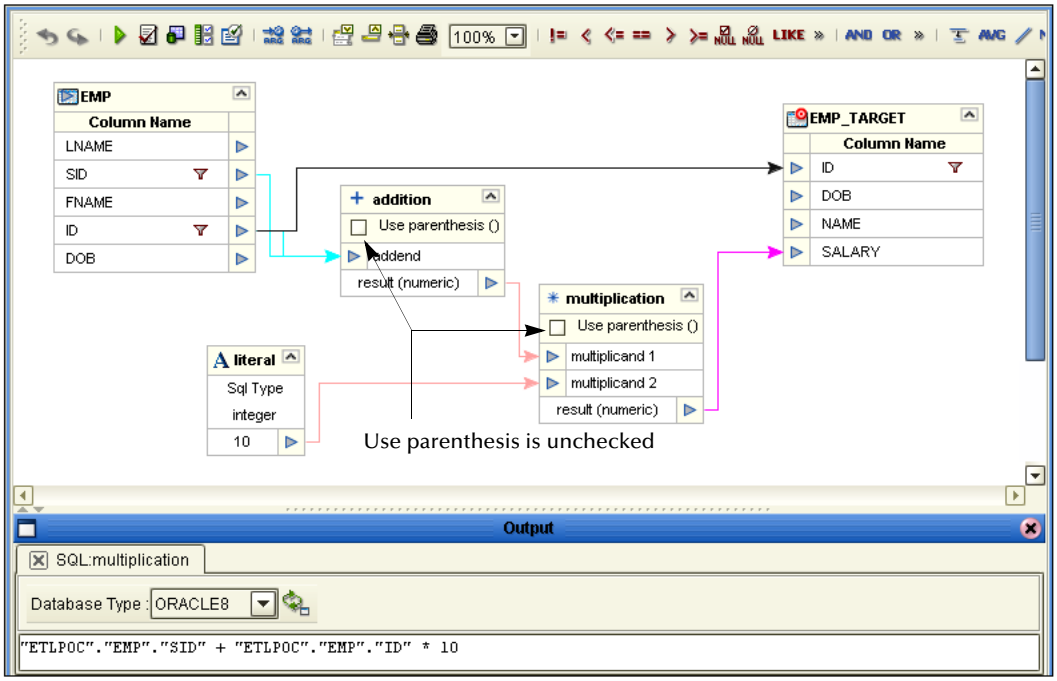
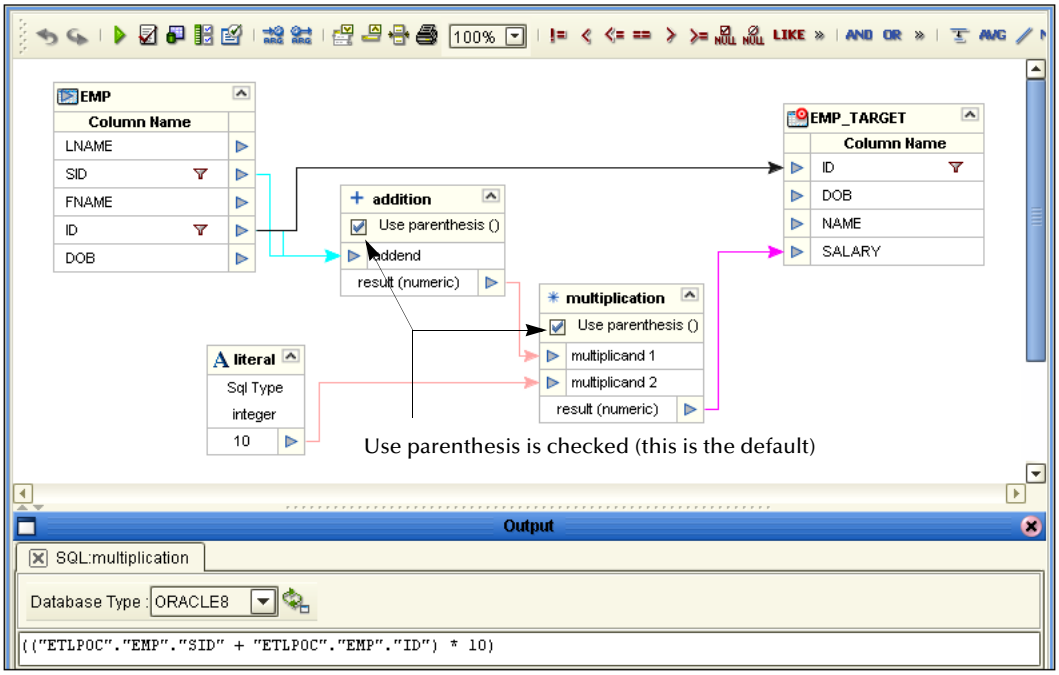


Figure 42 Addition Has Precedence



The example in Figure 42 shows the default, where parenthesis are used, and evaluation of the equation is left to right.

4.3.3 Calculating Dates

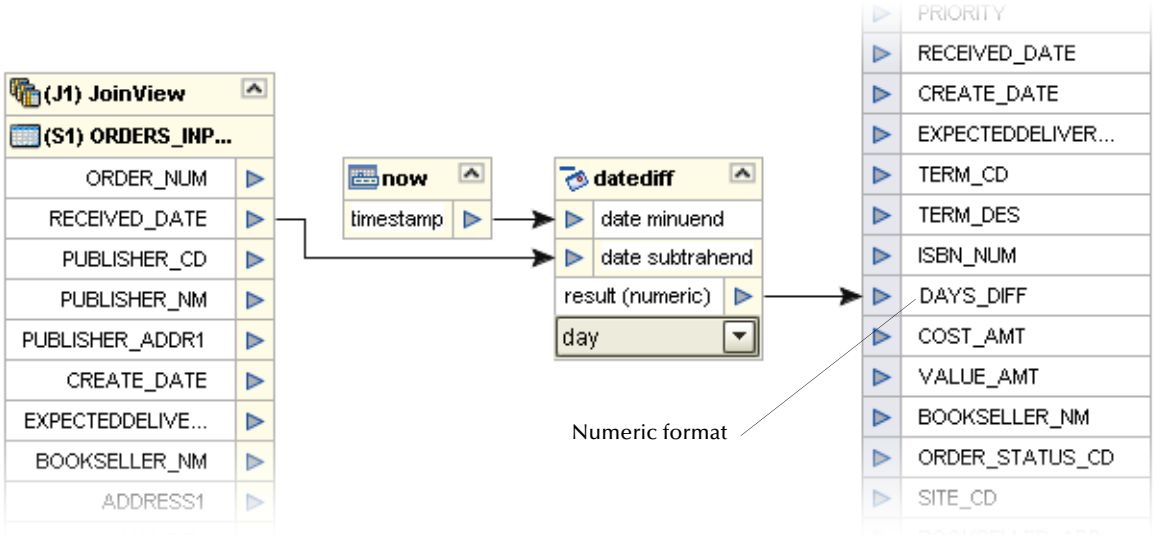
You can calculate the difference between two dates using any variable from second to year. See **About Date Operators** on page 56.

Date Difference

The **datediff** operator calculates the number of (days, weeks, months, etc.) between two dates and return a numeric result.

See the following figure.

Figure 43 Date Difference

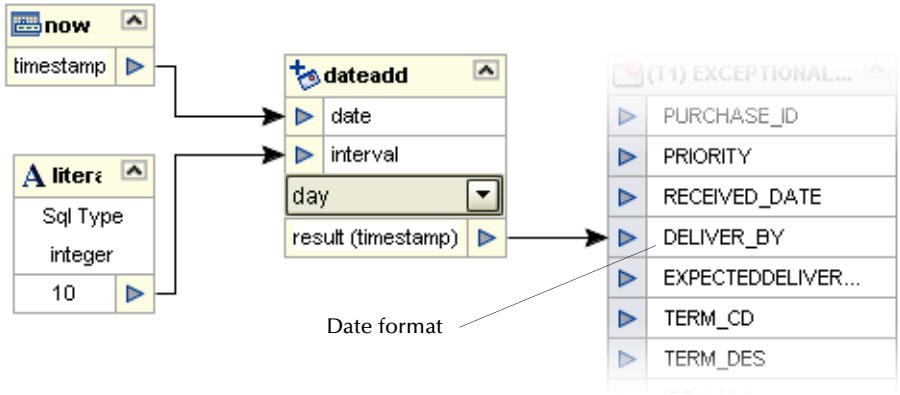


In this example the number of days since the **RECEIVE_DATE**, (which is subtracted from **now**, today's date) is passed as a numeric value to the target column as **DAYS_DIFF**. The target column is an integer that is not in date format.

Date Add Interval

The **dateadd** operator adds a date interval integer in (days, weeks, months, etc.) to the timestamp or other date input field and returns a new date. The resulting date format is determined by the date format of the target table's database.

Figure 44 Adding Date Intervals



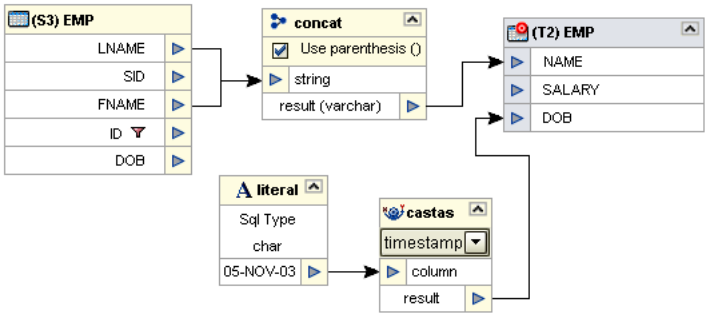
4.3.4 Using the Cast As Operator

This is a SeeBeyond operator used to change the data type. See **About SQL Operators** on page 60. When using **castas** be sure that the timestamp or date format is passed (cast function as an argument) in the exact format as the native database timestamp or date format.

For example, in Oracle9 you might use CAST ('05-NOV-03' AS timestamp), while for a flat file you might use CAST ('11-03-2003' AS timestamp).

See the following figure which illustrates **castas** in Oracle9-to-Oracle9.

Figure 45 Castas Example



4.3.5 Using the Case and LIKE Operators

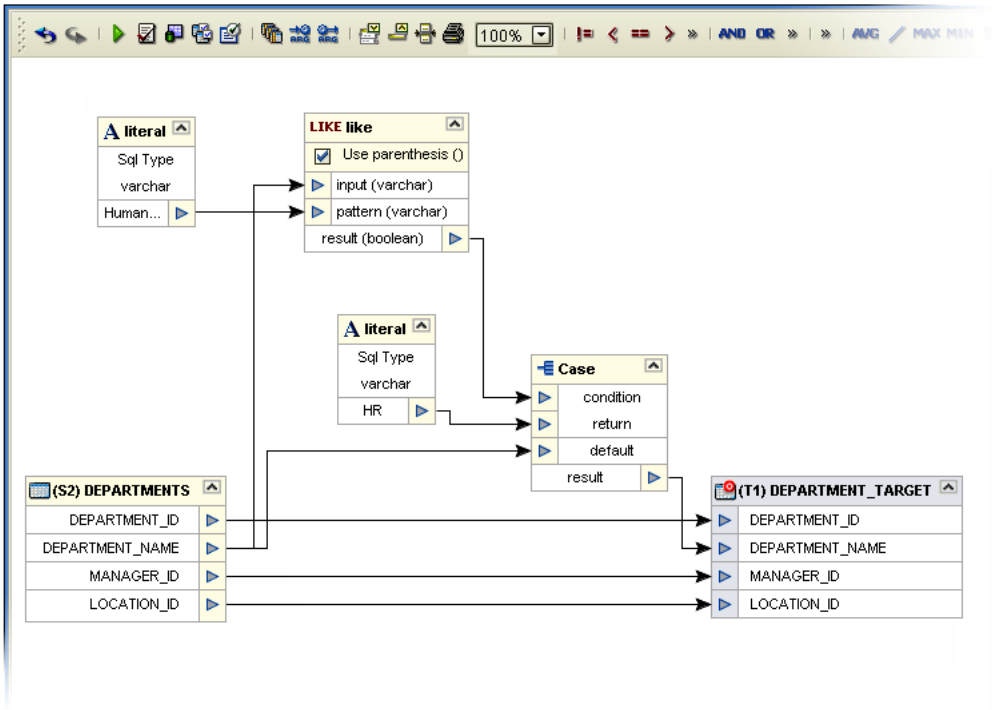
The following example demonstrates the use of the **Case** and the **LIKE** operators.

An Example of Using the Case and LIKE Operators

This example demonstrates one way to change the name of a department from "Human Resources" to "HR." The SQL, which is automatically generated, is shown below:

```
CASE
    WHEN ((S2.DEPARTMENT_NAME like 'Human Resources')) THEN ('HR')
    ELSE (S2.DEPARTMENT_NAME)
    END
```

Figure 46 Using the Case and LIKE Operators



4.4 Data Loading

4.4.1 Selecting Target Table Loading Options

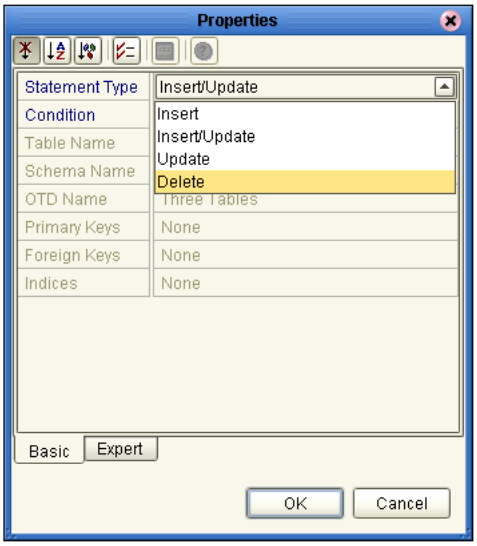
You can specify how to load target tables as described below. For a full load, select Insert. For conditional loading, select either upsert, update or delete.

To insert, upsert, update and delete rows

1. In the eTL Editor, right-click the target table and click **Properties**. The **Properties** dialog box appears.
2. In the **Statement Type** box, click one of the following:

Target Table Load Option	Description
Insert	Always append new rows (full load)
Insert/Update	Update an existing row or append a new row, depending on the evaluation of a condition (upsert)
Update	Update existing rows only
Delete	Delete rows

Figure 47 Inserting, Upserting, Updating and Deleting Rows

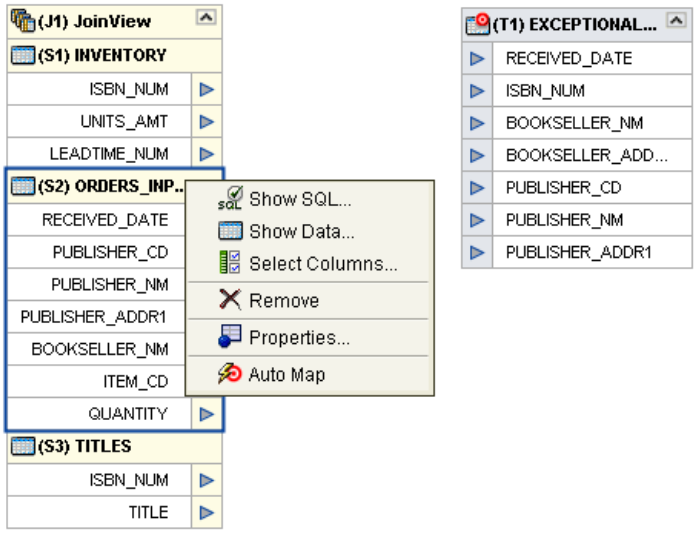


4.4.2 Using Auto Mapping

Auto mapping allows you to map all the matching columns of a source table to a single target table with two clicks of the mouse.

- Right-click the source table and click **Auto Map**.

Figure 48 Using Auto Mapping



All the columns from the selected source are automatically mapped to the matching columns in the target table. If two source tables have the same column name, only one will automap.

When multiple target tables have the same column name, a dialog box appears, enabling you to select the correct target table.

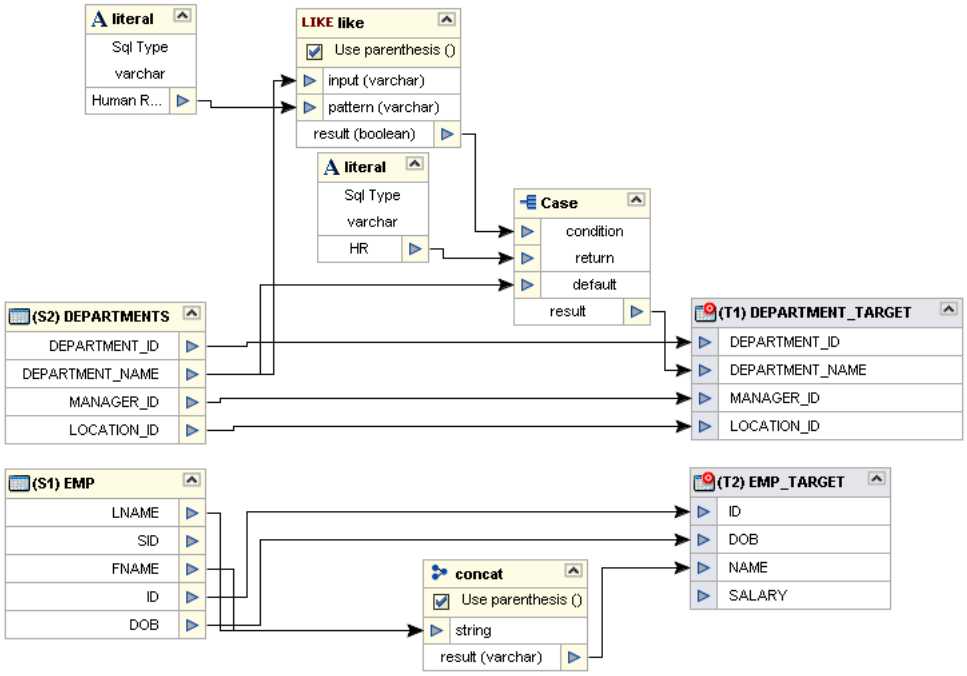
4.4.3 Mapping Multiple Targets

You may have multiple targets in one Collaboration. You may map each target table to a single source table or single joined table view.

An Example of Multiple Targets

In the following figure, two source tables (that are not joined) are mapped to two target tables. This is done in one Collaboration with parallel execution.

Figure 49 Multiple Targets Example

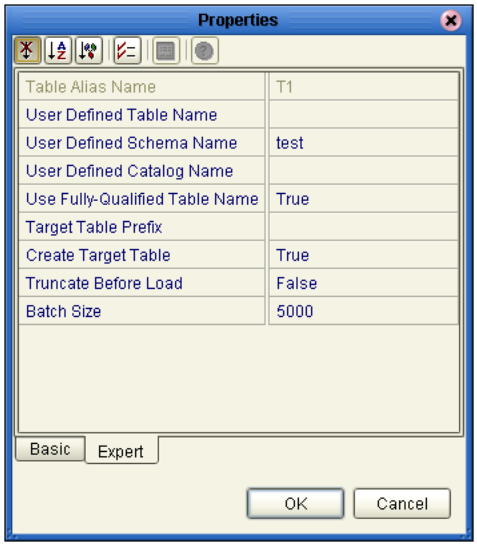


Note: *One source column maps to a single target column. You cannot map one source column to two or more target tables.*

4.4.4 Setting Target Table Expert Properties

The Expert property settings for the target table(s) are described in this section.

Figure 50 Target Table Expert Properties



The target table expert properties are the same as the source table properties except for the properties listed below. For information about rest of the properties, refer to **"Setting Source Table Expert Properties" on page 47**.

Target Table Prefix

You have the option to append a target table name prefix, if the name is defined, instead of the name derived from the OTD.

Create a Target Table

Set to 'True' to create the target table at runtime if it does not already exist.

Truncate Before Load

Set to 'True' to truncate the contents of the target before loading.

Chapter 5

Working with eTL Sample Projects

eTL Integrator comes with several sample Projects. You can import these Projects into Enterprise Designer and use them to quickly learn how to set up eTL Projects and business logic.

This chapter describes how you import and use the sample Projects.

In This Chapter

5.1 About the Sample Projects

eTL Integrator includes the two sample Projects that you can use to learn quickly how to set up eTL Projects.

The PeopleSoft eWay sample Projects are provided in the zip file **eTL_Integrator_User_Guide_Sample.zip**, which contains the following files:

- **Project_CSV.zip** (the flat file Project)
- **Project_eInsight.zip** (the eTL eInsight Project)
- **CSV_Exceptions_Output.csv**
- **CSV_Inventory_Input.csv**
- **CSV_Orders_Input.csv**

5.2 Locating the Sample Projects

The eWay sample Projects are included in the **eTLDocs.sar**. This file is uploaded separately from eTL Integrator sar file during installation. For information, refer to **"Installing eTL Integrator and Sample Projects" on page 15**.

Once you have uploaded the **eTLDocs.sar** to the Repository and you have downloaded the sample Projects (**eTL_Integrator_User_Guide_Sample.zip**) using the **DOCUMENTATION** tab in the Enterprise Manager, the sample resides in the folder specified during the download.

5.3 Importing the Sample Projects

You can import the PeopleSoft sample Projects as described below. To find out where the Projects reside, refer to **"Locating the Sample Projects" on page 72**.

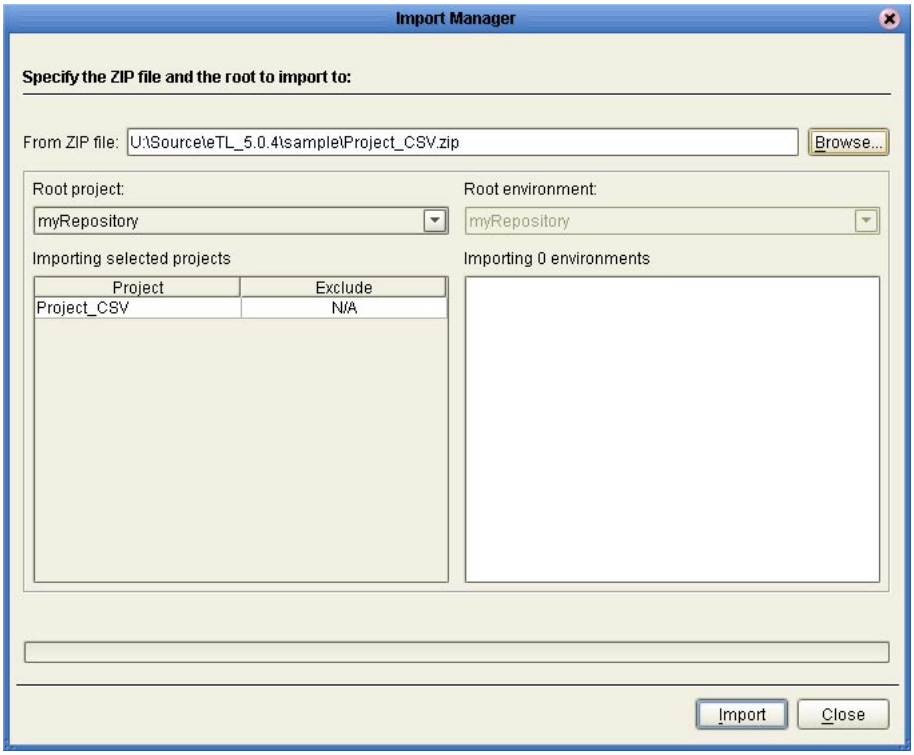
To import the sample Projects

1. Unzip the **eTL_Integrator_User_Guide_Sample.zip** file.
2. In the **Project Explorer** tab of the Enterprise Designer, right-click the Repository and click **Import**. A message confirms if you want to save your changes.
3. Click **Yes** to save your changes. The **Import Manager** dialog box appears.
4. Click **Browse** and navigate to the folder where you unzipped the sample zip file.
5. To import the flat file sample Project, click **Project_CSV.zip** and click **Open**.

   To import the eTL eInsight sample Project, click **Project_eInsight.zip** and click **Open**.

   The **Import Manager** dialog box appears.

Figure 51 Importing Sample Projects



6 Click **Import**. A dialog box confirms that the Project import was successful.

7 Click **OK** and click **Close**.

You can now explore the Connectivity Maps, the OTDs, and the business logic for the Collaborations or Business Processes.

5.4 Working with the Flat File Sample Project

In this project use three (CSV) flat files as your databases. Join two input source tables and output to a target table. Your output target table is a list of orders you cannot process (exceptional orders) because of insufficient inventory. Your target table gives order information and shows the quantity short.

5.4.1 Business Problem Description

Assume you manage a book exchange and you maintain an Inventory database and a Book Order database. You want to process orders that come in, but first you must check inventory to make sure that you have enough product (books) on hand. When you cannot immediately process the orders because inventory is too low, you want to create an exception report. You call this report "Exceptional Orders" (orders where the quantity ordered exceeds the inventory on hand).

The load orders activity executes an eTL Collaboration that has two input data sources:

Input Tables

The following two tables are joined:

- Customer Order table (**CSV_ORDERS_INPUT**): a staging table of orders
- Inventory table (**CSV_INVENTORY_INPUT**): a table representing the inventory tables for all available products

Output Table

The following table is the target:

- Exceptional table (**CSV_EXCEPTIONS_OUTPUT**): a compilation of orders that could not be processed because of insufficient inventory

5.4.2 Sample Data

To create the sample Project explained in this chapter, download the three sample CSV files from the ICAN Repository:

- **CSV_ORDERS_INPUT**
- **CSV_INVENTORY_INPUT**
- **CSV_EXCEPTIONS_OUTPUT**

Download the Sample Files

If you choose Option 1 in **"Project Overview" on page 81**, you can download the three sample CSV files from the Enterprise Manager. (You do not have to import the Project, **Project_CSV**, if you plan to create the Project yourself by going through the steps in the following scenario.)

1. Open Enterprise Manager and click the **DOCUMENTATION** tab.
2. Select **eTL Integrator** from the list of products.

   You have the options to download the user guide and/or the sample files which include the Project.
3. Click **Download Sample** and save and extract the files to a location on your computer, such as **C:\eTLSamples**.

   The downloaded file **eTL_project_sample.zip** contains both the Project (**Project_CSV**) and the sample CSV flat files listed above. Remember where you locate the sample files so you can enter a path to them later in the scenario.

The layout for the three CSV flat files used in the scenario for this chapter are shown below.

CSV_ORDERS_INPUT

The following table displays the sample data for the orders input:

Table 10 Orders Input Table (CSV)

ORDER_NUM	PRIORITY	REVEIVED_DATE	BOOKSELLER_NAME	ISBN_NUM	ORDER_QTY
10001	5	9/26/2004	Adamson Pub	0-4545-2110-2	250
10001	3	9/26/2004	Adamson Pub	0-4545-2210-1	400
10002	7	4/4/2004	Firestone Livres	0-4545-2310-7	120
10002	5	4/4/2004	Firestone Livres	0-4545-3221-2	20
10003	4	5/11/2004	Hardcount Pub	0-4545-3366-5	210
10003	9	5/11/2004	Hardcount Pub	0-4545-3413-2	20
10004	2	6/10/2004	Moonves Books	0-4545-3421-6	400
10004	5	6/10/2004	Moonves Books	0-4545-3535-3	100
10004	7	6/10/2004	Moonves Books	0-4545-4369-1	20
10005	8	10/17/2004	Ural Russian Lit	0-4545-5413-8	70

CSV_INVENTORY_INPUT

The following table displays the sample data for the inventory, showing quantity on hand:

Table 11 Inventory Table (CSV)

ISBN_NUM	QTY_ON_HAND
0-4545-2110-2	300
0-4545-2210-1	200
0-4545-2310-7	80
0-4545-3221-2	300

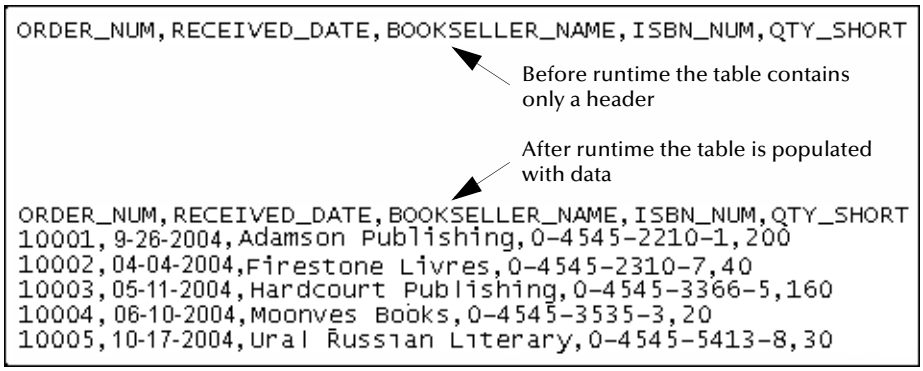
Table 11 Inventory Table (CSV) (Continued)

ISBN_NUM	QTY_ON_HAND
0-4545-3366-5	50
0-4545-3413-2	20
0-4545-3421-6	400
0-4545-3535-3	80
0-4545-4369-1	30
0-4545-5413-8	40

CSV_EXCEPTIONS_OUTPUT

The following table displays the sample data for the output (Exceptional Orders). The fields are populated at runtime:

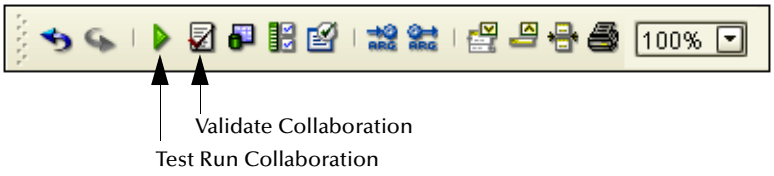
Figure 52 Output Data



5.4.3 Validating and Testing Collaborations

You can validate your Collaboration to verify the generated Collaboration source code syntax at design time. When you execute a test, you can check the SQL, the result data, or the log. (The log lists execution, success or failure messages, and other messages. See *eGate Integrator System Administration Guide* for more information about log files.) See the following figure.

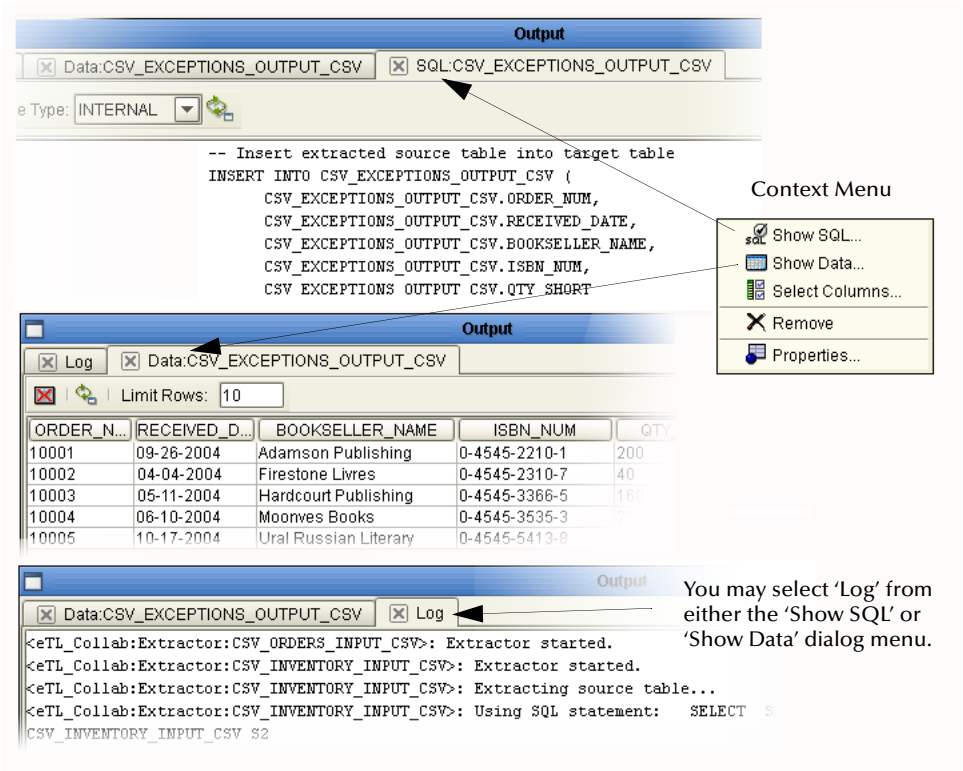
Figure 53 Menu Icons



- Click the **Validate Collaboration** icon to verify that there are no violations of the Collaboration rules.
- Click the **Test Run Collaboration** icon to run your Project.

- Right-click the **Target** table. The Context menu appears. See the following figure.

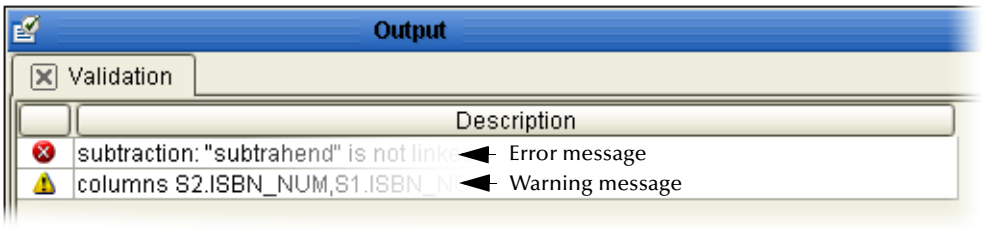
Figure 54 Show SQL, Data, and Log



- Verify that the logic has produced the correct output data.

  See the following example of warning and error messages.

Figure 55 Warning and Error Message Example



- Error message - not a valid eTL Collaboration
- Warning message - warning only

5.4.4 Creating the eInsight HTTP Sample Environment

The procedure below describes how you create the sample Environment for the eInsight HTTP Project. For detailed information about creating Environments, refer to the *eGate Integrator User's Guide*.

5.4.5 Creating The Deployment Profile

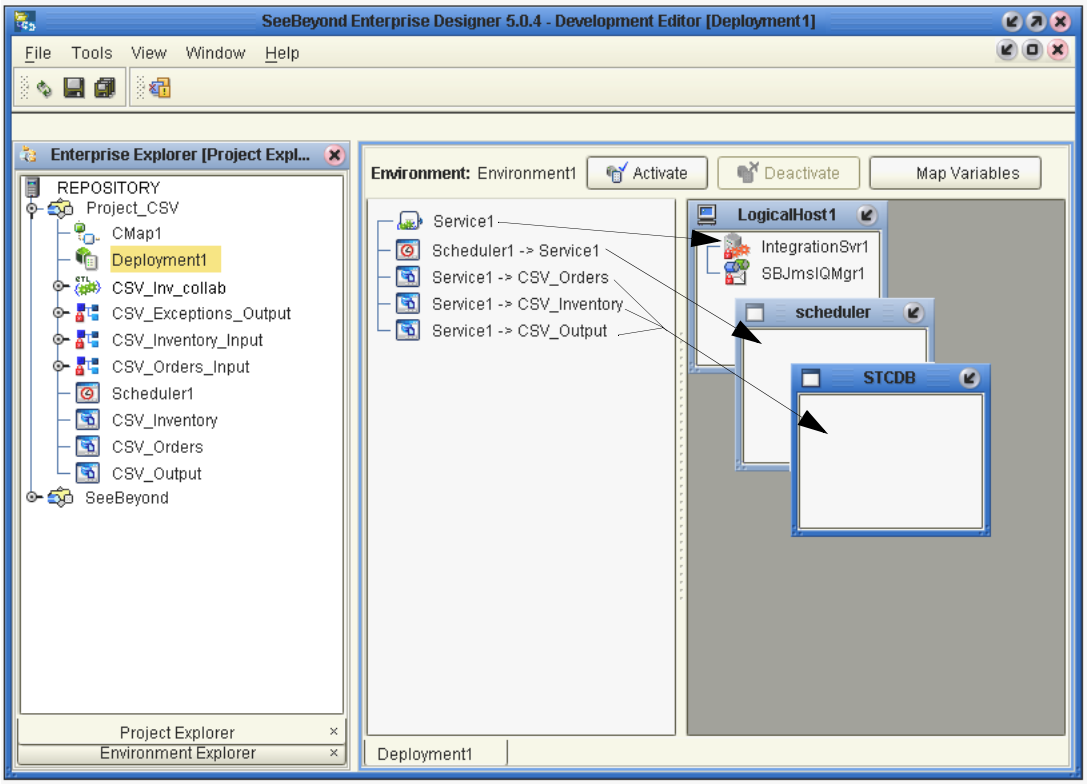
A Deployment Profile contains information about how Project components are deployed in an Environment, and it also maps components to the Environment.

To create the Deployment Profile

1. Click **View** and **Project Explorer** to return to your Project Explorer pane.
2. Right-click your Project (**Project_CSV**). Click **New** and then click **Deployment Profile**.

   Accept the default names **Deployment1** and **Environment1**.
3. Click **OK**.
4. Deploy the elements by dragging them into the servers in the right pane as shown in the following figure.

Figure 56 Deployment Editor



5. Click **Save All**.

   This completes your deployment.

5.4.6 Starting the Bootstrap and Management Agent

The Bootstrap process executes your ICAN Project and begins the process of polling your input data. The Bootstrap process is started from a command prompt or by

executing **Bootstrap.bat**. Bootstrap picks up the Deployment profile the first time it runs; after that redeploy your project to apply the most recent changes to the Logical Host.

Note: *The examples in this section explaining the running of the Bootstrap batch file are for Windows platforms only.*

At run time, the Logical Host Bootstrap script starts the Bootstrap Java program that downloads the Management Agent, the Message Server, and the Integration Server from the Repository. The Management Agent is then started, which in turn starts the Message Server(s) and Integration Server(s).

Start Bootstrap from a Command Prompt

1. From a Command Prompt, navigate to the Bootstrap bin directory for your Logical Host. For example, on a Windows system this could be: **c:\ican50\logicalhost\Bootstrap\bin**.
2. To start the Bootstrap service, type the following command:

```
Bootstrap -e Environment1 -l LogicalHost1 -r repository_url
-i username -p password
```

where

repository_url is the path to your Repository. For example, **http://MyServer:12000/MyRepository**. Substitute the name and port for your Repository server as well as the name of your repository.

username is your user name, and

password is your password.

Figure 57 Bootstrap Command Example

5.4.7 Running The Project

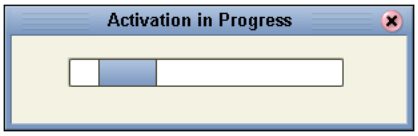
When you deploy your **Project**, a deployment file is created and later is picked up by the Logical Host. The Project you created is "code generated" into a package that is passed to the Repository. The Logical Host picks up this package from the Repository.

Before running your Project, make sure the Bootstrap has finished activating and the Logical Host and Environment is ready.

Activate Environment

1. Click the **Activate** button. The Activation in Progress message appears.

Figure 58 Activation in Progress Message



2 Check the box, Apply Environment Updates, the first time you execute the bootstrap, then click **YES**. See the following figure.

Note: *If you do not check the box, the Management Agent does not pick up the current Environment changes but only picks up the runable project file.*

Figure 59 Activation Options

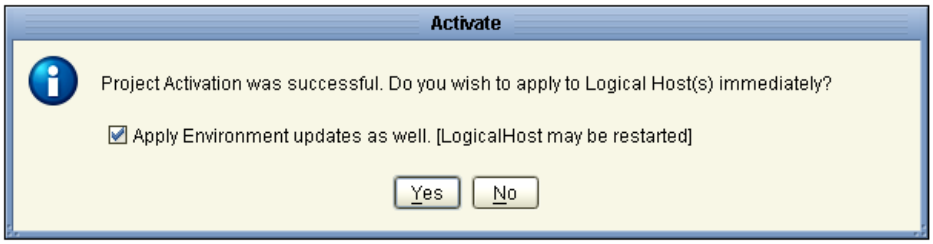
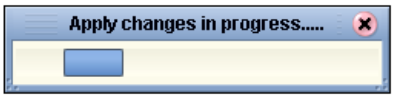
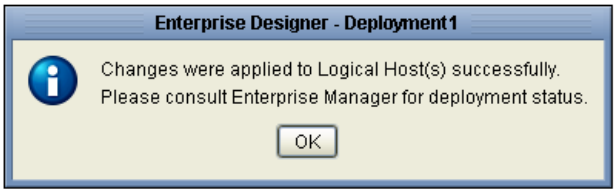


Figure 60 Apply Changes in Progress



Applying changes to the Deployment Profile may take a few minutes. If your changes are successful the following message appears.

Figure 61 Changes Applied to Logical Host



3 Click **OK**.

Note: *You can make changes while the Project is running and then Reactivate to get the latest changes.*

5.4.8 Verify the Output Data

The Management Agent processes your source files and writes to **CSV_EXCEPTIONS_OUTPUT.CSV**. This file is in the location where you downloaded

your sample. If your Logical Host continues to run, additional records could append to your output file, depending on your specific settings.

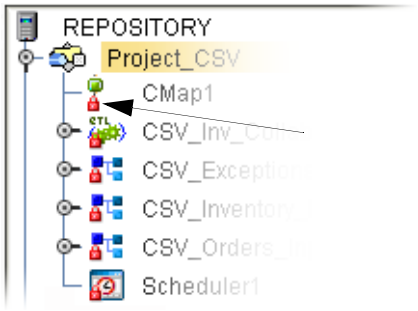
Press the **Deactivate** button to stop the processing (See Figure 56.)

Figure 62 Output Table

```
ORDER_NUM,RECEIVED_DATE,BOOKSELLER_NAME,ISBN_NUM,QTY_SHORT
10001,09-26-2004,Adamson Publishing,0-4545-2210-1,200
10002,04-04-2004,Firestone Livres,0-4545-2310-7,40
10003,05-11-2004,Hardcourt Publishing,0-4545-3366-5,160
10004,06-10-2004,Moonves Books,0-4545-3535-3,20
10005,10-17-2004,Ural Russian Literary,0-4545-5413-8,30
```

You can delete records from your output file using a text editor such as WordPad. When you delete records, be sure to retain your heading row.

Figure 63 Checkout Components



5.5 Working with the eTL eInsight Sample Project

5.5.1 Project Overview

The Business Process used in this scenario has been added to demonstrate the eTL interface to eInsight. The Business Process counts the number of exception orders added to the target table and displays a status message.

The eTL Collaboration portion of this project captures “exception” orders (orders that cannot be filled because of insufficient inventory).

The Connectivity Map portion of this project links message and data Services to produce two output files.

The deployment portion of this project runs the project to produce two output files: **Msg_Out** (showing the count and status), and **Output_file** (showing the exception records and the quantity short).

5.5.2 Business Problem Description

The business problem is defined in **Business Problem Description** on page 74. As in that scenario, this scenario will also:

- write the exception orders, showing the quantity short.

In addition, an eInsight Business Process is added to:

- write a count of the exception orders.
- write the status message.

5.5.3 Sample Data

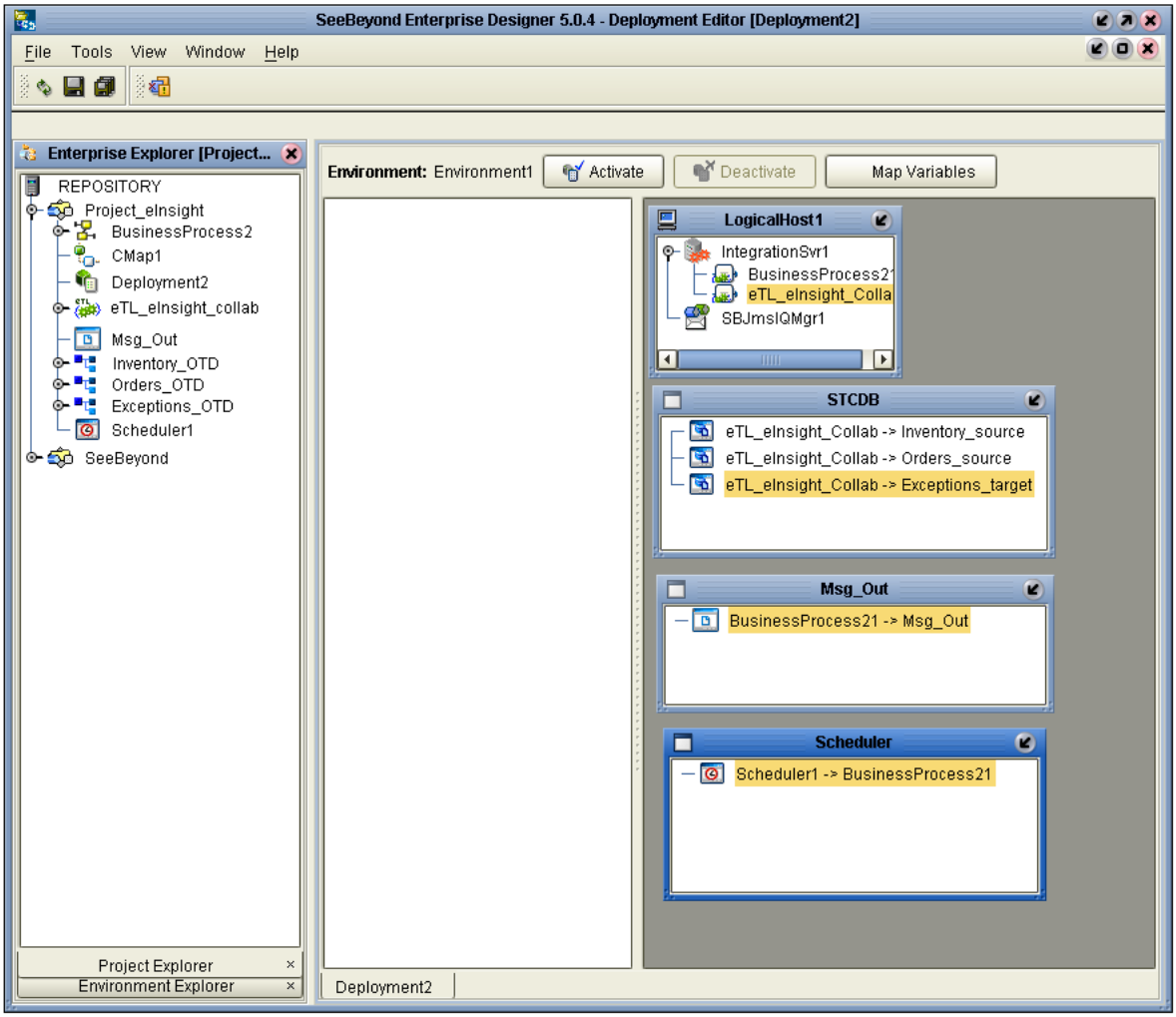
The sample data is the same as defined in **Business Problem Description** on page 74. In this eTL/eInsight scenario you create an additional text output file (**Msg_Out**) that shows the status and the record count of exceptional orders.

5.5.4 Creating The Environment and Activating the Deployment Profile

An Environment is a collection of physical resources and their configurations that are used to host Project components.

When you deploy your **Project_eInsight** it should look similar to the following figure.

Figure 64 Deployment Editor - Project_eInsight



5.5.5 Starting the Bootstrap and Management Agent

The Bootstrap process executes your Project and begins the process of polling your input data. See **Starting the Bootstrap and Management Agent** on page 78.

5.5.6 Running The Project

When you deploy your Project, a Deployment file is created and picked up by the Logical Host. See **Running The Project** on page 79.

The Management Agent processes your input files and writes to **CSV_EXCEPTIONS_OUTPUT.CSV**. If your Logical Host continues to run, additional records could append to your output file, depending on your specific settings.

Press the **Deactivate** button to stop the processing.

Figure 65 Output Table

```
ORDER_NUM,RECEIVED_DATE,BOOKSELLER_NAME,ISBN_NUM,QTY_SHORT
10001,09-26-2004,Adamson Publishing,0-4545-2210-1,200
10002,04-04-2004,Firestone Livres,0-4545-2310-7,40
10003,05-11-2004,Hardcourt Publishing,0-4545-3366-5,160
10004,06-10-2004,Moonves Books,0-4545-3535-3,20
10005,10-17-2004,Ural Russian Literary,0-4545-5413-8,30
```

The output is the same as in **Linking Connectivity Map Components** on page 31.

You can delete records from your output file using a text editor such as WordPad. When you delete records, be sure to retain your heading row.

In addition to the CSV output, you will also have a message (text) file that was created in your Business Process. If your project ran successfully, your output should look similar to the following figure.

Figure 66 Messages Out

```
'Status'Success'Count'5
```

Chapter 6

Managing Runtime eTL Projects

This chapter describes how to manage runtime eTL Projects. Once you have built and deployed an eTL Project, you can monitor it using the Enterprise Manager. This chapter includes information about monitoring eTL Projects as well as reconfiguring deployed Projects.

In This Chapter

6.1 Reconfiguring Deployed Projects

This section describes how you reconfigure the logical and physical properties of deployed Projects that have already been deployed. The logical properties are configured in the Connectivity Map.

6.1.1 Reconfiguring Logical Project Properties

To reconfigure a currently deployed Project, you change the configuration and then reactivate the Deployment Profile. If you also made changes to the logical properties in the Connectivity Map, apply the changes to the Logical Host as described in the next section.

The procedure below describes how you reconfigure the logical Project properties.

To configure logical eWay properties

1. Configure the logical properties in the Connectivity Map as described the *eGate Integrator User's Guide* and if appropriate, the eWay User's Guide.
2. In the **Project Explorer** tab, double-click the Deployment Profile for the Project.
3. Click **Reactivate**.

6.1.2 Reconfiguring Physical Project Properties

To reconfigure a currently deployed Project, you change the configuration and then apply the changes to the Logical Host as described below. If you also made changes to

the logical properties in the Connectivity Map, you must also reactivate the Deployment Profile as described in the *eGate Integrator User's Guide.*

The procedure below describes how you reconfigure the physical eWay properties.

To reconfigure physical eWay properties

1. Configure the physical properties in the Environment as described in *eGate Integrator User's Guide* and if appropriate, the eWay User's Guide.
2. In the **Environment Explorer** tab, right-click the Logical Host that contains this eWay.
3. Click **Apply**.

6.2 Monitoring eTL Collaborations

You monitor eTL Collaborations with the Enterprise Manager. For more information using the Enterprise Manager, refer to the *eGate Integrator System Administration Guide* and the *eGate Integrator User's Guide*.

6.3 Log Files and Alerts

eTL alerts are logged in the Logical Host log file. For information about this log file, and how to change the logging level in Enterprise Manager, refer to the *eGate Integrator System Administrator Guide*.

Index

O

P

R

S

T

U

W