

ChainBuilder ESB

Visual Enterprise Integration™

Version 1.2 M3 – 2008/02/05

ETL User Guide



©Copyright 2008
Bostech Corporation
2800 Corporate Exchange Drive
Suite 260
Columbus, OH 43231

Acknowledgements

This document contains proprietary information that is the property of Bostech Corporation. Any reproduction, disclosure, or transfer of this document or the information contained herein without the express written consent of Bostech Corporation is strictly prohibited.

The use of the information contained in this document and the implementation of any of its techniques are the sole responsibility of the client and depend on the client's ability to evaluate the information and implement it into the client's operational environment.

Except for any express written warranties made by it, Bostech Corporation makes no warranties or representations with respect to any information contained herein, whether express, implied, statutory, or otherwise, in fact or in law, including without limitation, any implied warranties of merchantability or fitness for a particular purpose; and in no event shall Bostech Corporation be liable for any special, consequential, indirect, punitive, or exemplary damages in connection with the use of the information contained herein. The information contained in this document is subject to change at any time without notice.

Trademarks

The following trademarks and acknowledgments apply to the information presented in this manual:

- ChainBuilder is a registered trademark of Bostech Corporation.
- Adobe and Acrobat Reader are registered trademarks of Adobe, Inc.
- Java is a registered trademark of Sun Microsystems, Inc.
- Windows (NT, 2000, XP, and Server 2003), .NET Framework, Internet Information Services (IIS) are registered trademarks of Microsoft Corporation.

Credits

The following third-party products are used within the ChainBuilder product, and acknowledgments apply to the information presented in this manual:

- Acrobat Reader is created and licensed by Adobe, Inc.
- This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)
- This product includes software developed by Eclipse (<http://www.eclipse.org/>)

Table of Contents

Table of Contents	3
Overview	4
Prerequisites	4
ChainBuilder ESB Community	4
Introduction	5
Configuration Wizards	6
Create JDBC Driver file	6
Create Connection	7
Create a data source map	8
Create an ETL Config File (consumer)	12
Create an ETL Config File (provider)	13
Custom component configuration	15
General Usage Steps	17

Overview

The ETL component available in ChainBuilder ESB Professional provides a helpful graphical interface that gives powerful access to extract data from and load data into a database.

Prerequisites

The user should have installed ChainBuilder ESB Professional and should have read the Getting Started Guide.

ChainBuilder ESB Community

ChainForge.net is the internet's premier destination to share ChainBuilder and JBI knowledge with your peers.

Join the ChainBuilder ESB Community:

<http://www.chainforge.net/community>

As a member you can view content and contribute to a Forum:

<http://www.chainforge.net/community/forums.html>

Read ChainBuilder ESB related Blogs:

<http://www.chainforge.net/blogs>

Introduction

ETL is an acronym for Extract, Transform and Load which generally refers to performing data extraction and/or loading to a relational database. The ChainBuilder ESB ETL Component allows the user to define connections to a particular database and generate an XML schema that can be used to extracting data from a database or loading data into a database.

When extracting data from a database, the ETL component will perform one or more select statements to create an XML message. This XML model of the data can be hierarchical to show the relationship between a master table and subordinate tables with foreign keys, or it can be configured as a flat layout that is the result of a joined select. Multiple extraction methods are provided to select specific rows (i.e. select and update a column, select and delete the row, select based on time interval in a column, etc).

When loading data into a database, the ETL component will receive an XML message that contains the data and will insert or update the affected rows in the database. In this case, the supplied schema can be used as the target in a transformation, so that the user can map data from any format into the target database.

Several new wizards have been added to the product which allow for configuration and testing of the database connection. The ETL component itself is delivered as a custom component, and added to the Component Flow Editor through the Custom binding component. The various setup wizards must be completed prior to setting up the custom component.

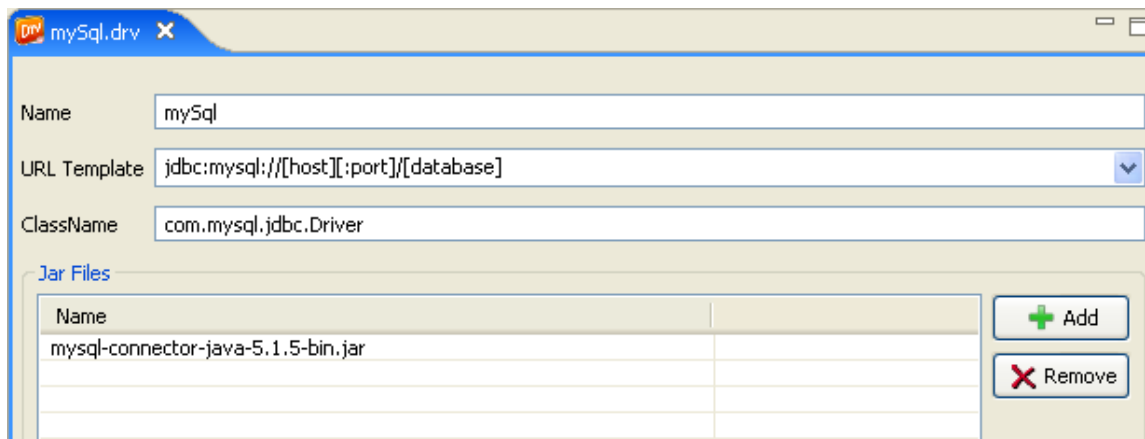
Configuration Wizards

Create JDBC Driver file

Purpose: The driver is a generic configuration for a particular database install, and references the JDBC drivers for that install. For example, the Driver is a reference to an install of MySQL (which may contain multiple databases in it)

Usage:

1. Right-click on src/database/drivers and select New -> JDBC Driver File
2. Name the driver
3. Select (from the pull-down menu) a URL template. Leave the template as it is – do not edit it. The class name should be automatically populated.
4. Use the add button to locate the JDBC driver file(s) required for the particular database.

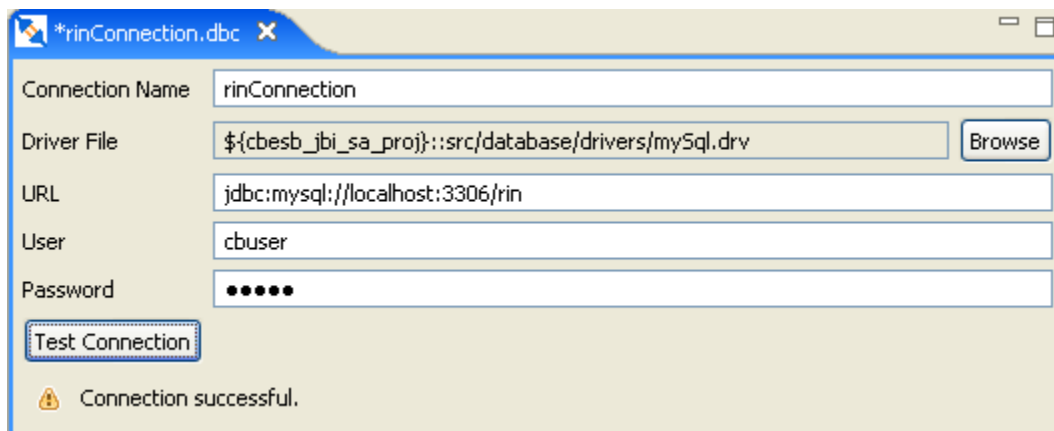


Create Connection

Purpose: The connection is used to define a JDBC connection to a specific instance of a database. The connection is based on a driver (which must already be created) and references that driver to connect to a particular database defined within that install. The database must exist and be accessible to create the connection.

Usage:

1. Right-click on src/database/connections and select New -> Connection file
2. Browse to select the driver file created in the above step
3. Edit the URL (change the [] references to real values)
4. Enter the User and Password
5. Test the connection: It is necessary to successfully test the database connection at this point, as the next step requires a successful connection.

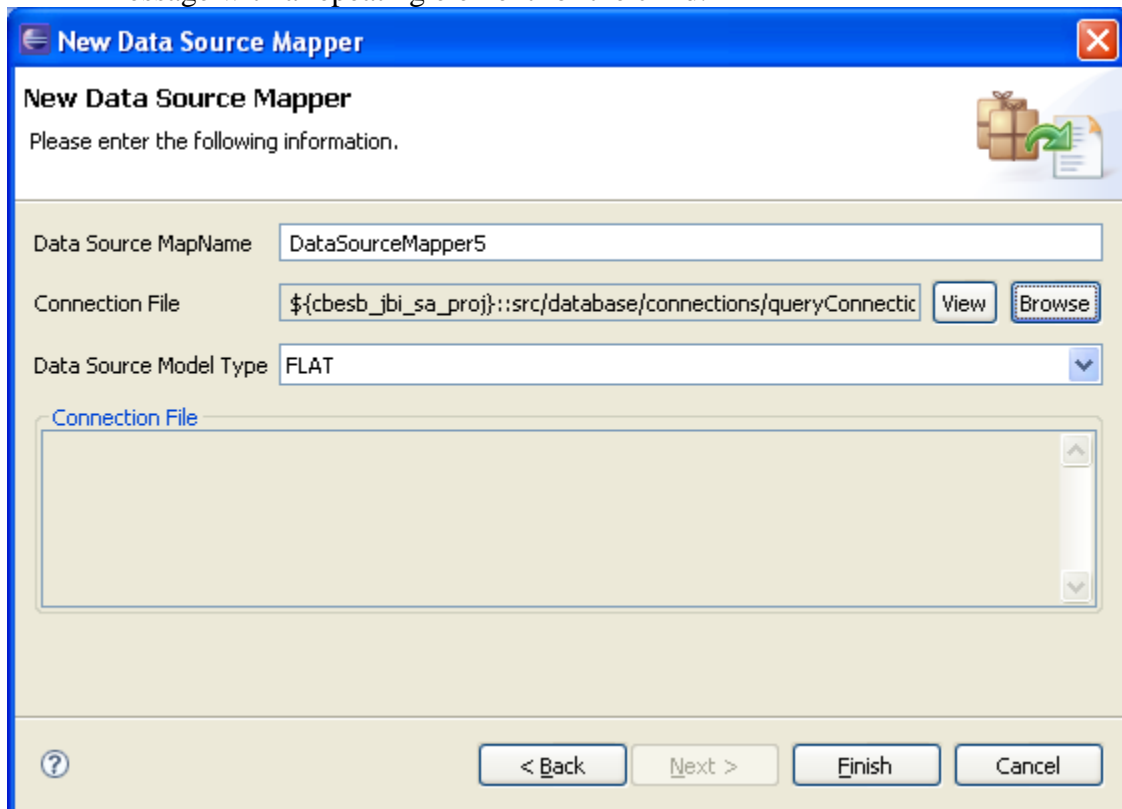


Create a data source map

Purpose: The Data Source mapper defines the XML format that will be used to extract or load data into or out of the database. The Data Source Mapper is tied to a particular Connection.

Usage:

1. Right-click on src/database/dataSourceMaps and select New -> Data Source Mapper File
2. Name the file
3. Select the connection file
4. Select the model type
 - a. Flat: used when only referencing a single table or referencing multiple tables without foreign key relationships (or when a sibling relationship is preferred in the generated schema). In the case of a table that has multiple references in a sibling, this will result in multiple messages being sent – one for each different sibling.
 - b. Hierarchical: used when multiple tables are connected by foreign keys and a parent-child relationship is desired in the generated xml schema. In the case of a parent table that has multiple references in a child, this will result in a single message with a repeating element for the child.



New Data Source Mapper

Please enter the following information.

Data Source MapName: DataSourceMapper5

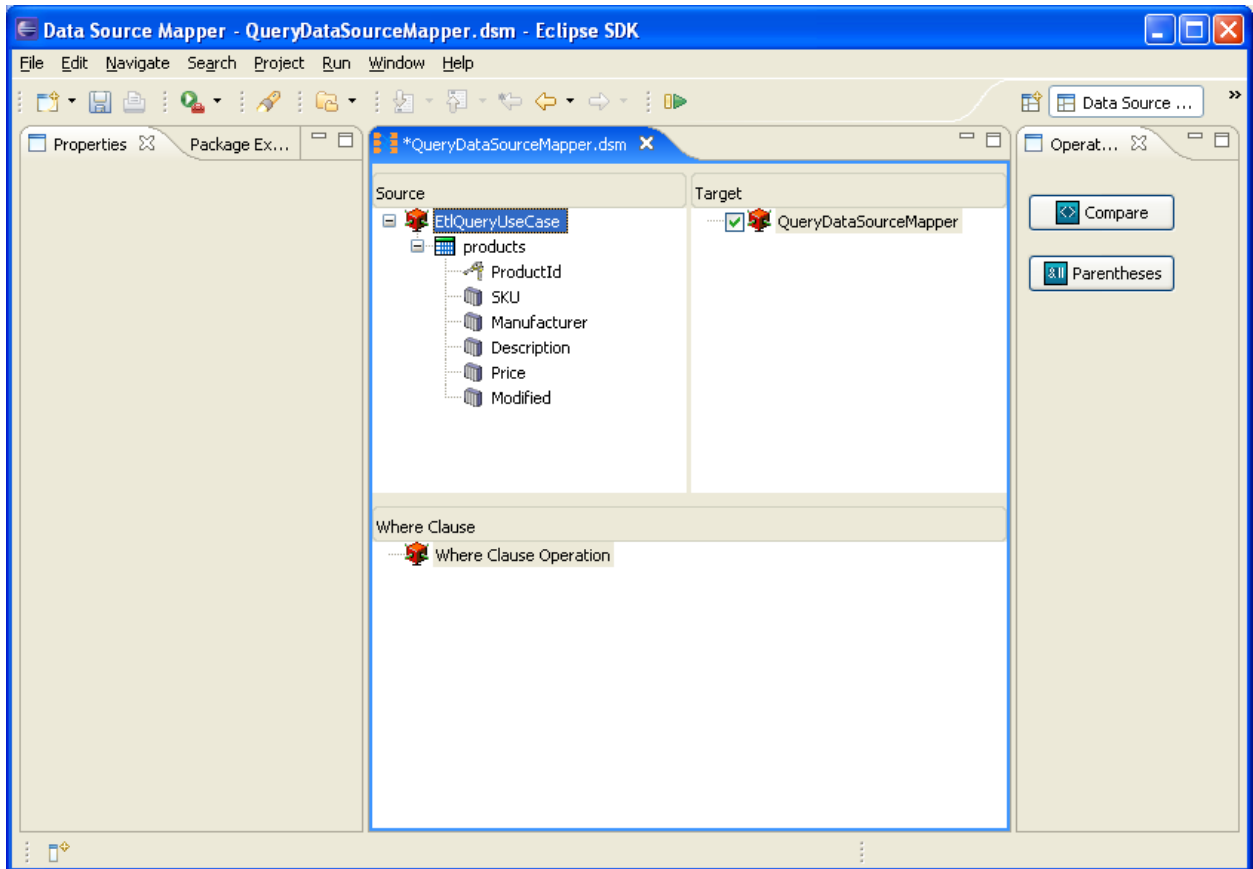
Connection File: \${cbesb_jbi_sa_proj}::src/database/connections/queryConnectic [View] [Browse]

Data Source Model Type: FLAT

Connection File

[?] < Back Next > Finish Cancel

5. Click Finish – this will open the Data Source Mapper



- a. Properties: contains detail properties for a selected Where clause operation
 - b. Source: details tables that exist in the selected Connection
 - c. Target: builder for the xml format
 - d. Operation Palette: contains buttons for available Where clause operations
 - i. Compare: create a new where clause
 - ii. Parentheses: used for grouping when creating complex where clauses
6. To configure the Target (this will determine the generated schema), drag a table (only drag a table, not a field) from Source to the red box on the target side. This will create the first table. Additional tables can be added by dragging from Source to a table that has already been added to Target. Based on the previously selected Flat or Hierarchical mode, one of two dialog boxes will be presented. In both cases, it is necessary to click on fields on the left (source) and right (target) to define foreign key relationships, then click Add to add that mapping to the Operation list. The Hierarchical option additionally requires the user to enter a Group name, which will be used as a repeating element in the resultant schema.
- a. Flat

Add Relationship Dialog.

Column Name	Type	Column Name	Type
phone.Id	INTEGER	person.Id	INTEGER
phone.PersonId	INTEGER	person.FirstName	VARCHAR
phone.Type	VARCHAR	person.LastName	VARCHAR
phone.Number	INTEGER	person.Title	VARCHAR
		person.Processed	VARCHAR

Operation

Compare:AND phone.PersonId = person.Id

b. Hierarchical

Add Relationship Dialog.

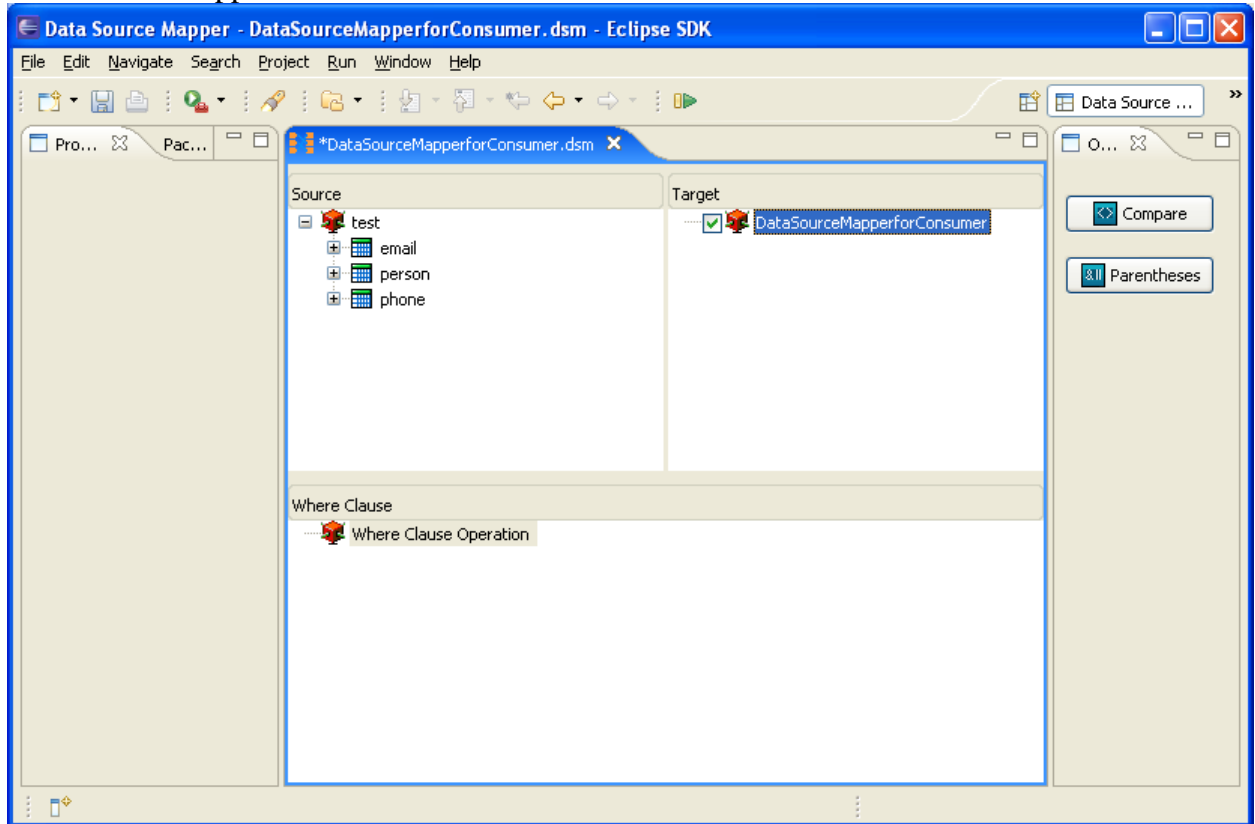
Group Name

Column Name	Type	Column Name	Type
email.Id	INTEGER	person.Id	INTEGER
email.PersonId	INTEGER	person.FirstName	VARCHAR
email.Type	VARCHAR	person.LastName	VARCHAR
email.Address	VARCHAR	person.Title	VARCHAR
		person.Processed	VARCHAR

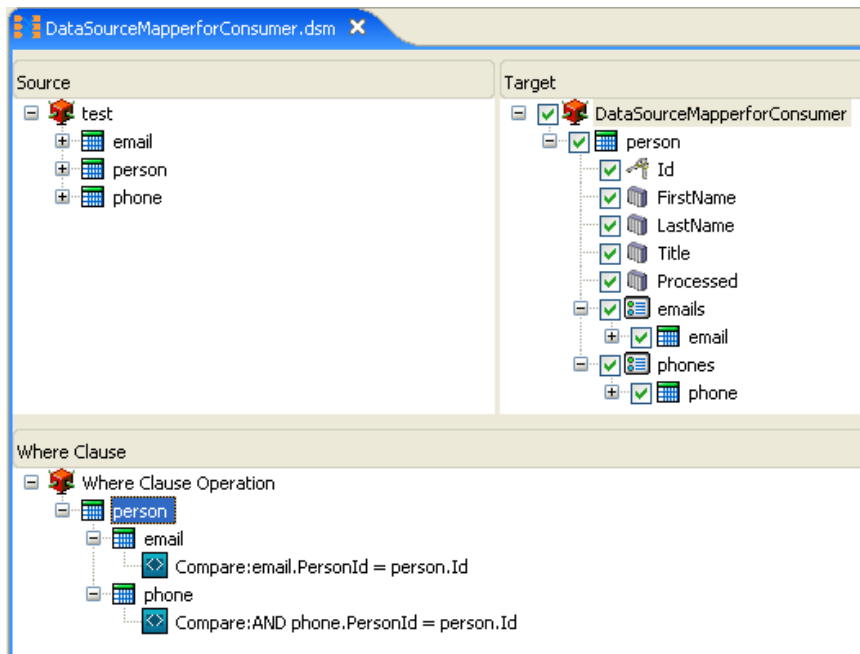
Operation

Compare:AND email.PersonId = person.Id

Data Source Mapper initial view



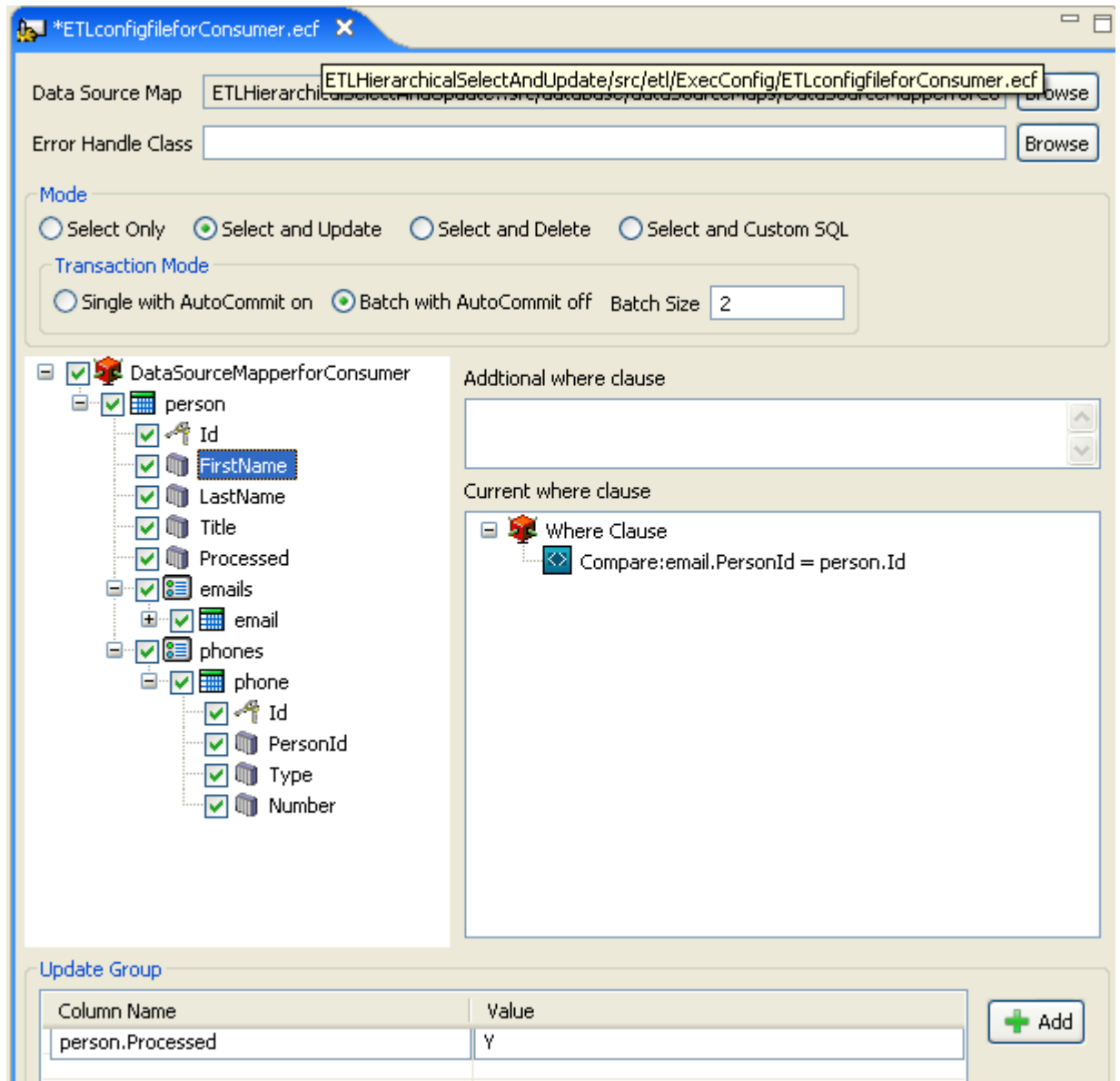
7. The Where Clause can be configured by using the Compare, then dragging fields (from source or target) down, and can be grouped by the parenthesis.



Create an ETL Config File (consumer)

Purpose: Defines the action to be taken on the defined data map

- a. Right-click on src/etl/ExecConfig and select New -> ETL Config File
- b. Select Mode
 - i. Select Only: Only a select (or multiple selects) are performed. This is usually only used when selecting on a data/time range.
 - ii. Select and Update: After a row is selected, one or more fields in that row can be updated to prevent the row from being selected again later.
 - iii. Select and Delete: After a row is selected, the row is deleted from the table
 - iv. Select and Custom SQL: After a row is selected, a user entered SQL statement is executed. This could be used if a temp table is used to identify when new rows are inserted. The custom SQL could be used to delete the row from the temp table, but leave the real data tables unmodified.
- c. If Mode is not Select Only, select Transaction Mode
 - i. Single with AutoCommit on: The driver's AutoCommit is set to True. The transaction will not be rolled back even if there is an error.
 - ii. Batch with AutoCommit off: The driver's AutoCommit is set to False. A batch of database updates will be committed or rolled back depending on if there is an error.
 1. Batch Size: set the batch size, or leave blank for infinite (entire results from select statement treated as one batch)
- d. Additional Where Clause: Used to restrict which rows are gathered. Drag a field from the left panel to the Additional Where Clause box and complete the where clause manually, for example by adding = 'Y' to restrict the query to those cases where the selected field = Y.
- e. Update Group: Used only with Select and Update to define the field that should be updated after the select has completed. Click the Add button, then drag a field to the Column Name and enter a value in the Value column.

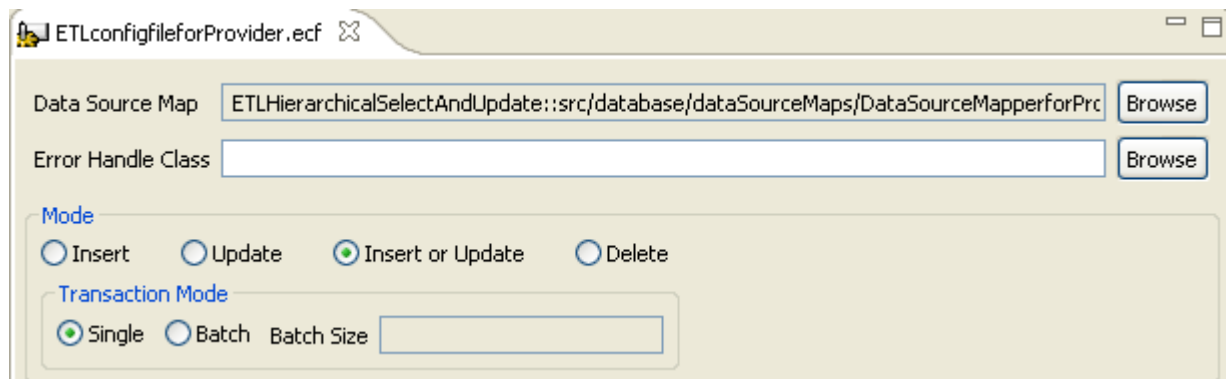


Create an ETL Config File (provider)

Purpose: Defines the action to be taken on the defined data map

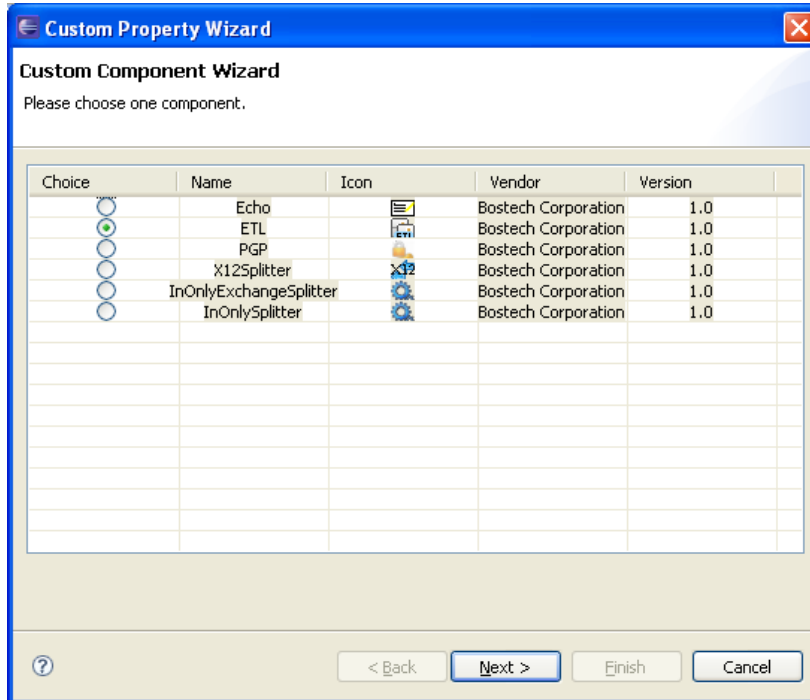
- f. Right-click on src/etl/ExecConfig and select New -> ETL Config File
- g. Select Mode
 - i. Insert: All messages received are treated as an insert statement. If the data already exists in the database, the operation will fail.
 - ii. Update: All messages received are treated as an update statement. If a matching record does not exist in the database, the operation will fail.
 - iii. Insert or Update: Each message is treated as an insert, if there is already a matching record in the database, the statement is changed to an update and re-executed.
 - iv. Delete: Each message is treated as a delete statement

- h. Select Transaction Mode
 - i. Single: Database operations from one MessageExchange are treated as a single transaction that will be committed or rolled back depending on if there is an error.
 - ii. Batch: Database operations from a batch of MessageExchange are treated as a single transaction that will be committed or rolled back depending on if there is an error.
 - 1. Batch Size: Specify the size of the batch. If it is one, a transaction for each MessageExchange received will be committed or rolled back. If the Batch Size is not specified, the batch is controlled by the metadata in received MessageExchange.
- i. Additional Where Clause: Used to restrict which rows are gathered. Drag a field from the left panel to the Additional Where Clause box and complete the where clause manually, for example by adding = 'Y' to restrict the query to those cases where the selected field = Y.
- j. Update Group: Used only with Select and Update to define the field that should be updated after the select has completed. Click the Add button, then drag a field to the Column Name and enter a value in the Value column.

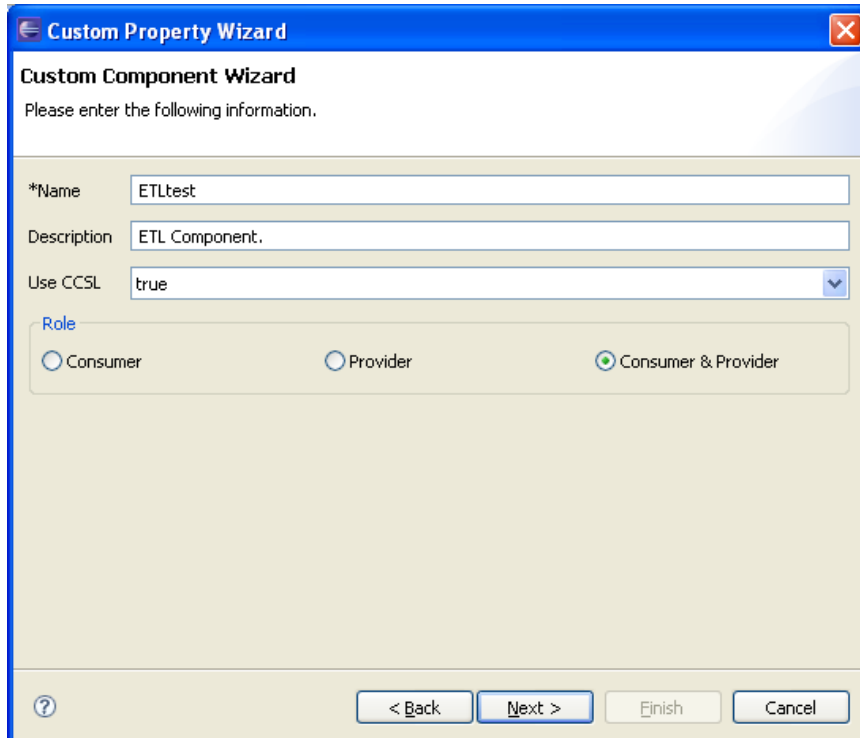


Custom component configuration

The ETL component is added to the component flow editor as a custom component. The user must first select which custom component to use from the list. Select ETL and click Next.



The first screen in the ETL wizard allows the user to name the component and select the Role.



Based on the selected Role, the user will have either a Consumer screen, a Provider screen, or both, followed by the standard CCSL Consumer/Provider screens.

The screenshot shows the 'Custom Property Wizard' window with the 'Consumer Operations' tab selected. The window title is 'Custom Property Wizard'. Below the title bar, the text 'Consumer Operations' is displayed, followed by the instruction 'Please select Exec config file.'.

The 'Default MEP' is set to 'in-out' in a dropdown menu. The '*Exec_config_file' field contains the path '\$\{cbesb_jbi_sa_proj\}::src/etl/ExecConfig/ETLconfigfileforConsumer.ecf' and has a 'Browse...' button next to it.

The 'Trigger Method' section has two radio buttons: 'Scan Interval' (selected) and 'Schedule'. The 'Scan Interval' field is set to '5000'. The 'Schedule' section contains a table with two columns: 'Type' and 'Detail'. The table is currently empty. To the right of the table are three buttons: 'New', 'Edit', and 'Remove'.

At the bottom of the window, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. A help icon (?) is located in the bottom left corner.

The screenshot shows the 'Custom Property Wizard' window with the 'Provider Operations' tab selected. The window title is 'Custom Property Wizard'. Below the title bar, the text 'Provider Operations' is displayed, followed by the instruction 'Please select Exec config file.'.

The '*Exec_config_file' field contains the path '\$\{cbesb_jbi_sa_proj\}::src/etl/ExecConfig/ETLconfigfileforProvider.ecf' and has a 'Browse...' button next to it.

At the bottom of the window, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. A help icon (?) is located in the bottom left corner.

In both cases, the user must select an existing ETL Config file. In the consumer screen, the user may also modify the default MEP.

General Usage Steps

For a typical use case of the ETL component, the user will perform steps in the following order:

1. create database and tables
2. create database driver
3. create database connection
4. create data source map
5. create ETL config file
6. configure component flow editor