Getting Started

INFORMATICA® POWERMART® 5
INFORMATICA® POWERCENTER® 5
(VERSION 5.1)

Informatica PowerMart/PowerCenter Getting Started
Version 5.1.0
June 2001

Table of Contents

List of Figures

List of Tables

Preface

Welcome to PowerMart and PowerCenter, Informatica's integrated suite of software products that deliver an open, scalable solution addressing the complete life cycle for data warehouse and analytic application development. Both PowerMart and PowerCenter combine the latest technology enhancements for reliably managing data repositories and delivering information resources in a timely, usable manner.

The metadata repository coordinates and drives a variety of core functions including extraction, transformation, loading, and management. The Informatica Server can extract large volumes of data from multiple platforms, handle complex transformations, and support high-speed loads. PowerMart and PowerCenter can simplify and accelerate the process of moving data warehouses from development to test to full production.

About Informatica Documentation

The complete set of printed documentation for PowerMart and PowerCenter includes the following books:

- ***Designer Guide.*** Provides information needed to use the Designer. Includes information to help you create mappings, mapplets, and transformations. Also includes a description of the transformation datatypes used to process and transform source data.
- ***Getting Started.*** Provides basic tutorials for getting started. Also contains documentation about the sample repository.
- ***Installation and Configuration Guide.*** Provides information needed to install and configure the Informatica tools, including details on environment variables and database connections.
- ***Metadata Reporter Guide.*** Provides information on how to install and use the web-based Metadata Reporter to generate reports on the metadata in PowerMart and PowerCenter repositories.
- ***Repository Guide***. Provides information needed to administer the repository using the Repository Manager or the *pmrep* command line program. Includes details on functionality available in the Repository Manager, such as creating and maintaining repositories, folders, users, groups, and permissions and privileges.
- ***Session and Server Guide.*** Provides information to help you create and run sessions in the Server Manager, as well as administer the Informatica Server. Also contains information on tuning sessions and the Informatica Server for better performance.
- ***Transformation Language Reference.*** Provides syntax descriptions and examples for each transformation function provided with PowerMart and PowerCenter.
- ***Troubleshooting Guide.*** Provides error messages that you might encounter while using PowerMart or PowerCenter. Each error message includes one or more possible causes and actions that you can take to correct the condition.

Documentation available with our other products includes:

- **Informatica® Metadata Exchange SDK *User Guide.*** Provides information about the second generation of Metadata Exchange interfaces for PowerMart and PowerCenter repositories.
- **Informatica® PowerChannel™ *User Guide*.** Provides information on how to transport compressed and encrypted data through a secure channel.
- **PowerCenter™ Integration Server for BW *Installation Guide.*** Provides information to install and configure PCISBW for loading source data into a SAP Business Warehouse. It also contains basic instructions for loading into BW.
- **PowerConnect™ for IBM® DB2® *Installation Guide.*** Provides information on setting up a mainframe environment to work with PowerCenter, including connectivity issues for mainframe and DB2 databases.

- **PowerConnect™ for IBM® MQSeries®** ***User and Administrator Guide.*** Provides information to install PowerConnect for IBM MQSeries, build mappings, extract data from message queues, and load data to message queues.
- **PowerConnect™ for PeopleSoft®** ***User and Administrator Guide.*** Provides information to install PowerConnect for PeopleSoft, extract data from PeopleSoft, build mappings, and run sessions to load PeopleSoft source data into your data warehouse.
- **PowerConnect™ for SAP™ R/3®** ***User and Administrator Guide.*** Provides information to install PowerConnect for SAP R/3, extract data from SAP R/3, build mappings, and run sessions to load SAP R/3 data into a data warehouse.
- **PowerConnect™ for Siebel®** ***User and Administrator Guide.*** Provides information to install PowerConnect for Siebel, extract data from Siebel systems, build mappings, and run sessions to load Siebel source data into a data warehouse.
- **PowerPlug™** ***User Guide.*** Provides information on how to use PowerPlug to extract metadata from leading data modeling tools and import it into PowerMart/PowerCenter repositories through Informatica Metadata Exchange SDK.
- **PowerBridge™ for Hyperion® Essbase® Server** ***User Guide.*** Provides information on how to use PowerBridge to export multi-dimensional metadata from PowerMart/PowerCenter repositories into the Hyperion Integration Server through Informatica Metadata Exchange SDK.

About this Book

Getting Started is written for the IS developers and software engineers who are responsible for implementing a data warehouse. It provides a tutorial to help first-time users learn how to use PowerCenter. *Getting Started* assumes you have knowledge of your operating systems, relational database concepts, and the database engines, flat files, or mainframe systems in your environment. The guide also assumes you are familiar with the interface requirements for your supporting applications. It also provides information on using the sample repository included for use with PowerMart/PowerCenter. The sample repository contains sample objects and mappings to help you read web logs, sort file data, and call Perl functions from mappings.

The online help and Adobe Acrobat versions of this book contain material for PowerMart 5.1/ PowerCenter 5.1. The printed version of this book contains material for PowerMart 5.0/ PowerCenter 5.0. To view printed material for the new features in PowerMart 5.1/PowerCenter 5.1, refer to the PowerMart/PowerCenter *Supplemental Guide.*

The material in this book is available for online use.

About PowerCenter and PowerMart

This guide contains information about both PowerMart and PowerCenter. The documentation explicitly mentions software features that differ between the two products.

If You Are Using PowerCenter

With PowerCenter, you receive all product functionality, including the ability to register multiple servers, share metadata across repositories, and partition data.

A PowerCenter license lets you create a single repository that you can configure as a global repository, the core component of a data warehouse.

When this guide mentions a PowerCenter Server, it is referring to an Informatica Server with a PowerCenter license.

If You Are Using PowerMart

This version of PowerMart includes all features except distributed metadata, multiple registered servers, and data partitioning. Also, the various options available with PowerCenter (such as PowerCenter Integration Server for BW, PowerConnect for IBM DB2, PowerConnect for IBM MQSeries, PowerConnect for SAP R/3, PowerConnect for Siebel, and PowerConnect for PeopleSoft) are not available with PowerMart.

When this guide mentions a PowerMart Server, it is referring to an Informatica Server with a PowerMart license.

Document Conventions

This guide uses the following formatting conventions:

If you see...	It means...
italicized text	The word or set of words are especially emphasized.
boldfaced text	Emphasized subjects.
`italicized monospaced text`	This is the variable name for a value you enter as part of an operating system command. This is generic text that should be replaced with user-supplied values.
Note:	The following paragraph provides additional facts.
Tip:	The following paragraph provides suggested uses.
Warning:	The following paragraph notes situations where you can overwrite or corrupt data, unless you follow the specified procedure.
`monospaced text`	This is a code example.
`bold monospaced text`	This is an operating system command you enter from a prompt to execute a task.

Other Informatica Resources

In addition to the product manuals, Informatica provides these other resources:

- *Informatica Webzine*
- Informatica web site
- Informatica Developer Network
- Informatica Technical Support

Accessing the Informatica Webzine

The Informatica Documentation Team delivers an online journal, the *Informatica Webzine*. This journal provides solutions to common tasks, conceptual overviews of industry-standard technology, detailed descriptions of specific features, and tips and tricks to help you develop data warehouses. You can access the webzine at:

http://www.informatica.com/webzine

The *Informatica Webzine* is a password-protected site. Informatica has an online registration form for login accounts to its webzine and web support. To register for an account, go to the following URL:

http://www.informatica.com/support/webpaswdreq.html

If you have any questions, please email webzine@informatica.com.

To better serve your needs, the Informatica Documentation Team welcomes all comments and suggestions. You can send comments and suggestions to:

documentation@informatica.com

Visiting the Informatica Web Site

You can access Informatica's corporate web site at http://www.informatica.com. The site contains information about Informatica, its background, upcoming events, and locating your closest sales office. You will also find product information, as well as literature and partner information. The services area of the site includes important information on technical support, training and education, and implementation services.

Visiting the Informatica Developer Network

The Informatica Developer Network is a web-based forum for third-party software developers. You can access the Informatica Developer Network at http://devnet.informatica.com. The site contains information on how to create, market, and support customer-oriented add-on solutions based on Informatica's interoperability interfaces.

Obtaining Technical Support

There are many ways to access Informatica technical support. You can call or email your nearest Technical Support Center listed below or you can use our WebSupport Service.

Both WebSupport and our Customer Site require a user name and password. To receive a user name and password, please contact us at support@informatica.com or call 650-687-6300.

North America / South America	Africa / Asia / Australia / Europe
Informatica Corporation 3350 W. Bayshore Rd. Palo Alto, CA 94303 Phone: 650.687.6300 Fax: 650.687.0040 Hours: 6 a.m. - 7 p.m./ p.s.t email: support@informatica.com	**Informatica Software Ltd.** 6 Waltham Park Waltham Road, White Waltham Maidenhead, Berkshire SL6 3TN Phone: UK +44 870 606 1525 Germany +49 1805 702 702 Switzerland +41 800 81 80 70 Fax: +44 1628 511 411 Hours: 9 a.m. - 5 p.m./ g.m.t email: support_uk@informatica.com

New Features and Enhancements

This section describes new features and enhancements to PowerMart and PowerCenter.

Version 5.1

The following list describes new features and enhancements in PowerMart 5.1/PowerCenter 5.1.

Performance Enhancements

- **High precision decimal arithmetic.** The Informatica Server optimizes data throughput to increase performance of sessions using the Enable Decimal Arithmetic option.
- **To_Decimal and Aggregate functions.** The Informatica Server uses improved algorithms to increase performance of To_Decimal and all aggregate functions such as percentile, median, and average.
- **Cache management.** The Informatica Server uses better cache management to increase performance of Aggregator, Joiner, Lookup, and Rank transformations.
- **Partition sessions with sorted aggregation.** You can partition sessions with Aggregator transformation that use sorted input. This improves memory usage and increases performance of sessions that have sorted data.

Relaxed Data Code Page Validation

When enabled, the Informatica Client and Informatica Server lift code page selection and validation restrictions. You can select any supported code page for source, target, lookup, and stored procedure data.

Designer Features and Enhancements

- **Debug mapplets.** You can debug a mapplet within a mapping in the Mapping Designer. You can set breakpoints in transformations in the mapplet.
- **Support for slash character (/) in table and field names.** You can use the Designer to import source and target definitions with table and field names containing the slash character (/). This allows you to import SAP BW source definitions by connecting directly to the underlying database tables.

Server Manager Features and Enhancements

- **Continuous sessions.** You can schedule a session to run continuously. A continuous session starts automatically when the Load Manager starts. When the session stops, it restarts immediately without rescheduling. Use continuous sessions when reading real time sources, such as IBM MQSeries.

- **Partition sessions with sorted aggregators.** You can partition sessions with sorted aggregators in a mapping.
- **Register multiple servers against a local repository.** You can register multiple PowerCenter Servers against a local repository.

Version 5.0

The following list describes new features and enhancements in PowerMart 5.0/PowerCenter 5.0.

Designer Features and Enhancements

- **Debugger.** You can debug a mapping from the Mapping Designer. Using the Debugger Wizard, choose to run the Debugger against an existing session in debug mode or create a debug session. Before you debug a mapping, set breakpoints in the mapping where you want the Informatica Server to evaluate data and error conditions. The Debugger pauses when a breakpoint evaluates to true, and you can change transformation output data.
- **Double-click folder icon to open the folder and tool.** When you double-click a folder icon in the Navigator window, the Designer opens the folder and displays the last tool that was active within that folder.
- **Find in Workspace tool.** You can search for text strings in tables, columns, or ports across all open objects in the workspace.
- **Full screen view.** You can view the workspace over the entire screen. The Designer hides the menu, the Navigator and Output windows, and the title bar to maximize the workspace window.
- **Invalid mapping icon.** The Designer displays a new icon for invalid mappings in the Navigator window.
- **Lookup cache.** You can use a dynamic lookup cache in a Lookup transformation. When you use a dynamic lookup cache, the Informatica Server updates the lookup cache during the session run. Also, you can share persistent lookup caches within and across mappings by naming the cache files.
- **Mapping parameters and variables.** Create mapping parameters and variables in a mapplet or mapping to make mappings more flexible. Mapping parameters and variables represent values in mappings and mapplets. Use mapping parameters and variables to reduce the overhead of creating multiple mappings when only certain attributes of a mapping need to change.
- **New Designer shortcut keys.** New shortcut keys allow you to cut, copy, paste, and delete rows when you edit a repository object.
- **New workspace look.** You can select two colors to create a color gradient for the title bars of repository objects. Other components of the workspace, such as links, arrows, and tool title, also have a sharper, three-dimensional look.
- **Object import and export.** You can export objects from a repository and import objects into a repository. When you export a repository object, the Designer or Server Manager

creates an XML file describing the repository metadata. Use the XML file to import repository objects into a repository. You can export and import sources, targets, transformations, mapplets, mappings, and sessions.

- **Preview data.** The Designer allows you to preview data for relational sources, flat file sources, and relational targets.
- **Retain relationships and source or target definition descriptions when re-importing a source or target definition.** When you re-import a source or target definition, you have the option of retaining primary and foreign key relationships and source or target descriptions.
- **Resizable Expression Editor.** You can resize the Expression Editor by dragging the corner of the dialog box.
- **Router transformation.** The Designer allows you to use a Router transformation to test data based on multiple conditions. The Router transformation allows you to route groups of data to a transformation or a target.
- **Save mappings and mapplets under a different name.** The Designer allows you to save mappings and mapplets under a different name within the same folder.
- **XML sources and targets.** You can use XML data as a source or target. You can create source and target definitions from an XML, DTD, or XML schema file and use them in mappings. When you run a session, you can read data from an XML source file and validate it against a DTD. You can also write data from any type of source to an XML file.

Informatica Server Enhancements

- **Expanded *pmcmd* capability.** You can use the command line program *pmcmd* to specify a parameter file when you start a session or batch. This allows you to change the values of session parameters, and mapping parameters and variables at runtime.
- **Parallel data processing.** Available with PowerCenter only. If you run the Informatica Server on a symmetric multi-processing system, you can use multiple CPUs to process a session concurrently. You configure partitions in the session properties based on source qualifiers. The Informatica Server reads, transforms, and writes partitions of data in parallel for a single session.
- **Process session data using threads.** The Informatica Server runs on two processes: the Load Manager process and the DTM process. The DTM process creates threads to perform session tasks, such as reading, transforming, and writing data as well as performing pre-and post-session operations.

Metadata Reporter

The Metadata Reporter is a web-based application that enables you to run reports against repository metadata. With the Metadata Reporter, you can access information about your repository without having knowledge of SQL, the transformation language, or the underlying tables in the repository.

Repository Manager Features and Enhancements

- **Compare Folders.** You can compare the contents of folders in the same repository or across different repositories.
- **Copy and Replace Folder Wizard.** You can maintain shortcuts from local to global repositories when you copy and replace folders. When copying folders, you can select the shared folder to associate shortcuts.
- **Display user privileges.** The Manage Privileges dialog box displays all user and group privileges so you can assign and revoke privileges by selecting the appropriate option.
- ***pmrep*.** A command-line utility for Windows NT/2000 that allows you to perform repository tasks and change session-related parameters. Using *pmrep*, you can connect to, back up, delete, and restore a repository. Additional *pmrep* commands allow you to update database connectivity information for a repository, update email address information for session notification email, and update server variables.

Server Manager Features and Enhancements

- **Additional email variables.** With new email variables, you can configure post-session email to include information, such as the mapping used during the session.
- **Additional server variables.** You can configure several new server variables when registering a server. Server variables such as $PMSuccessEmailUser allow you to configure sessions without hard-coding information.
- **Copy session.** You can copy sessions across folders and repositories using the Copy Session Wizard in the Server Manager.
- **Enhanced user security.** You can set read, write, and execute permissions for users and groups to restrict access to database, FTP, and external loader connections.
- **Global validation of sessions.** You can validate a specified set of sessions in a folder or validate all of the invalid sessions in a folder.
- **Read-only privileges for sessions.** You can set user and group permissions and privileges that allow users to view, but not edit, session properties.
- **Session parameters.** Create session parameters to reuse sessions. You can create parameters for database connections, reject files, and file sources and targets. You can also use a built-in session parameter for the session log. Define session parameters in the session parameter file.
- **Sybase IQ 12 External Loader.** You can use Sybase IQ 12 external loaders to load multibyte data. Null characters and delimiters can be up to four bytes long.
- **Teradata External Loader.** You can use Teradata external loaders to load single-byte data. The Teradata external loader uses ASCII-based single-byte code pages.

Transformation Language Enhancements

You can use two new built-in system variables with PowerMart and PowerCenter:

- **SESSSTARTTIME.** Returns the system date value on the machine hosting the Informatica Server when the server initializes the session. You can use SESSSTARTTIME within any function that accepts transformation date/time datatypes using the transformation language.
- **$$$SessStartTime.** Returns the system date value on the machine hosting the Informatica Server when the server initializes the session. You can use $$$SessStartTime at the mapping level in the filter condition, user-defined join, or SQL override of a source qualifier. You can also use $$$SessStartTime in session-level overrides for mapping attributes.

You can also use four new variable functions with mapping variables:

- **SetCountVariable.** Counts the number of evaluated rows and increment or decrement a mapping variable for each row.
- **SetMaxVariable.** Evaluates the value of a mapping variable to the higher of two values.
- **SetMinVariable.** Evaluates the value of a mapping variable to the lower of two values.
- **SetVariable.** Sets the value of a mapping variable to a specified value.

Chapter 1

Product Overview

This chapter covers the following topics:

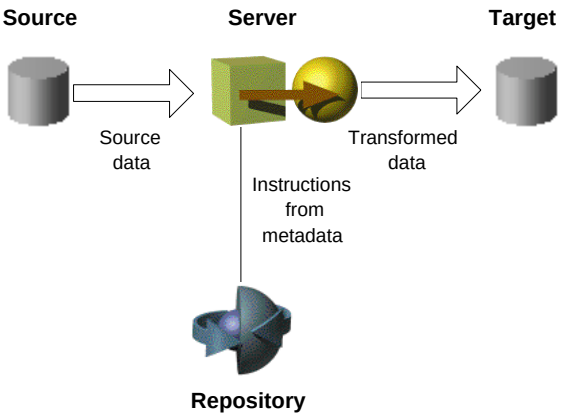
Introduction

PowerMart and PowerCenter provide an environment that allows you to load data into a centralized location, such as a datamart, data warehouse, or operational data store (ODS). You can extract data from multiple sources, transform the data according to business logic you build in the client application, and load the transformed data into file and relational targets. Informatica provides the following integrated components:

- **Informatica repository.** The Informatica repository is at the center of the Informatica suite. You create a set of metadata tables within the repository database that the Informatica applications and tools access. The Informatica Client and Server access the repository to save and retrieve metadata.
- **Informatica Client.** Use the Informatica Client to manage users, define sources and targets, build mappings and mapplets with the transformation logic, and create sessions to run the mapping logic. The Informatica Client has three client applications: Repository Manager, Designer, and Server Manager.
- **Informatica Server.** The Informatica Server extracts the source data, performs the data transformation, and loads the transformed data into the targets.

Figure 1-1 illustrates the architecture of PowerMart and PowerCenter:

Figure 1-1. PowerMart/PowerCenter Architecture



Sources

PowerMart and PowerCenter access the following sources:

- **Relational.** Oracle, Sybase, Informix, IBM DB2, Microsoft SQL Server, and Teradata.
- **File.** Fixed and delimited flat file, COBOL file, and XML.
- **Extended.** If you use PowerCenter, you can purchase additional PowerConnect products to access business sources such as PeopleSoft, SAP R/3, Siebel, and IBM MQSeries.

- **Mainframe.** If you use PowerCenter, you can purchase PowerConnect for IBM DB2 for faster access to IBM DB2 on MVS.
- **Other.** Microsoft Excel and Access.

Note: The Designer imports relational sources, such as Microsoft Excel, Microsoft Access, and Teradata using ODBC and native drivers.

For more information about sources, see "Working with Sources" in the *Designer Guide*.

Targets

PowerMart and PowerCenter can load data into the following targets:

- **Relational.** Oracle, Sybase, Sybase IQ, Informix, IBM DB2, Microsoft SQL Server, and Teradata.
- **File.** Fixed and delimited flat files and XML.
- **Extended.** If you use PowerCenter, you can purchase an integration server to load data into SAP BW. You can also purchase PowerConnect for IBM MQSeries to load data into IBM MQSeries message queues.
- **Other.** Microsoft Access.

You can load data into targets using ODBC or native drivers, FTP, or external loaders.

For more information about targets, see "Working with Targets" in the *Designer Guide*.

Repository

The Informatica repository is a set of tables that stores the metadata you create using the Informatica Client tools. You create a database for the repository, and then use the Repository Manager to create the metadata tables in the database.

You add metadata to the repository tables when you perform tasks in the Informatica Client application such as creating users, analyzing sources, developing mappings or mapplets, or creating sessions. The Informatica Server reads metadata created in the Client application when you run a session. The Informatica Server also creates metadata such as start and finish times of a session or session status.

When you use PowerCenter, you can develop global and local repository to share metadata:

- **Global repository.** The global repository is the hub of the domain. Use the global repository to store common objects that multiple developers can use through shortcuts. These objects may include operational or application source definitions, reusable transformations, mapplets, and mappings.
- **Local repositories.** A local repository is within a domain that is not the global repository. Use local repositories for development. From a local repository, you can create shortcuts to objects in shared folders in the global repository. These objects typically include source definitions, common dimensions and lookups, and enterprise standard transformations. You can also create copies of objects in non-shared folders.

You can connect to a repository, back up, delete, or restore repositories using *pmrep*, a command line program. For more information on *pmrep*, see "Using pmrep" in the *Repository Guide*.

For more information about global repositories, see "PowerCenter Repositories" in the *Repository Guide*.

You can view much of the metadata in the Repository Manager. You can also run metadata reports in the Repository Manager or through the Metadata Reporter. For more information about the Metadata Reporter, see the *Metadata Reporter Guide*.

Informatica Client

The Informatica Client is comprised of three applications that you use to manage the repository, design mappings, mapplets, and create sessions to load the data.

- **Repository Manager.** Use the Repository Manager to create and administer the metadata repository. You can create repository users and groups, assign privileges and permissions, manage folders and locks, and print Crystal Reports containing repository data.
- **Designer.** Use the Designer to create mappings that contain transformation instructions for the Informatica Server. Before you can create mappings, you must add source and target definitions to the repository. The Designer has five tools that you use to analyze sources, design target schemas, and build source-to-target mappings:
  - **Source Analyzer.** Import or create source definitions.
  - **Warehouse Designer.** Import or create target definitions.
  - **Transformation Developer.** Develop reusable transformations to use in mappings.
  - **Mapplet Designer.** Create sets of transformations to use in mappings.
  - **Mapping Designer.** Create mappings that the Informatica Server uses to extract, transform, and load data.
- **Server Manager.** Use the Server Manager to create, schedule, execute, and monitor sessions. You create a session based on a mapping in the repository and schedule it to run against an Informatica Server. You can view scheduled and running sessions for each Informatica Server in the domain. You can also access details about those sessions.

Install the client tools on a Microsoft Windows machine. For more information about installation requirements, see "Installation and Configuration Overview" in the *Installation and Configuration Guide*.

Informatica Server

The Informatica Server reads mapping and session information from the repository. It extracts data from the mapping sources and stores the data in memory while it applies the transformation rules that you configure in the mapping. The Informatica Server loads the transformed data into the mapping targets.

You can install the Informatica Server on a Windows NT/2000 or UNIX server machine. For more information about installation requirements, see "Installation and Configuration Overview" in the *Installation and Configuration Guide*.

You can communicate with the Informatica Server using *pmcmd*, a command line program. For more information, see "Using pmcmd" in the *Session and Server Guide*.

Connectivity

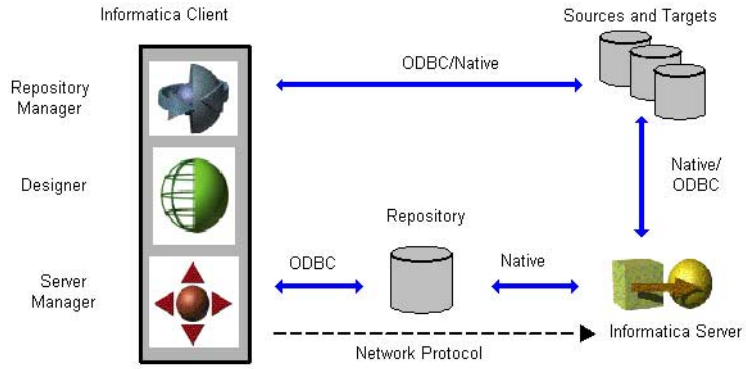
PowerMart and PowerCenter use the following types of connectivity:

- Network Protocol
- Native Drivers
- ODBC

The Informatica Client uses ODBC and native drivers to connect to source, target, and repository databases. The Server Manager and the Informatica Server use TCP/IP or IPX/SPX to communicate with each other. The Informatica Server uses native drivers to connect to the databases to move data. You can optionally use ODBC to connect the Informatica Server to databases. The Informatica Client uses ODBC and native drivers to connect to source, target, and repository databases.

Figure 1-2 shows the connectivity used within the PowerMart/PowerCenter architecture:

Figure 1-2. Connectivity Overview



Metadata Reporter

PowerMart and PowerCenter provide the Metadata Reporter, a web-based application that allows you to run reports against repository metadata. It gives you insight into your repository that enhances your ability to analyze and manage your repository efficiently.

The Metadata Reporter provides a number of reports, including reports on executed sessions, lookup table dependencies, mappings, and source or target schemas.

You can install the Metadata Reporter on a server with any type of operating system running a compatible web server. You can run the Metadata Reporter from a browser on any workstation, even without the Informatica tools installed on the workstation. For more information, see the *Metadata Reporter Guide*.

Using the Repository Manager

Use the Repository Manager to administer your repositories. The Repository Manager allows you to navigate through multiple folders and repositories, and perform the following tasks:

- **Perform repository maintenance.** You can create, copy, restore, upgrade, backup, and delete repositories. With a global repository, you can register and unregister local repositories. You can import and export repository connection information in the registry and edit repository connection information.
- **Implement repository security.** You can create, edit, and delete repository users and user groups. You can assign and revoke repository privileges and folder permissions.
- **Perform folder functions.** You can create, edit, copy, and delete folders. All the work you perform in the Designer is stored in folders. If you want to share metadata, you can configure a folder to be shared.
- **View metadata.** You can analyze sources, targets, mappings, and shortcut dependencies, search by keyword, and view the properties of repository objects.
- **Customize the Repository Manager.** You can add, edit, and remove repositories in the Navigator, view or hide windows.
- **Run repository reports.** You can run repository reports such as the Source to Target Dependency report or the Session report. You can also add and remove customized reports.

For more information about the repository and the Repository Manager, see the *Repository Guide.*

Repository Manager Windows

The Repository Manager can display the following windows:

- **Navigator.** Displays all objects that you create in the Repository Manager, the Designer, and the Server Manager. It is organized first by repository, then by folder and folder version. Viewable objects include sources, targets, dimensions, cubes, mappings, mapplets, transformations, sessions, and batches. You can also view folder versions and business components.
- **Main.** Provides properties of the object selected in the Navigator window. The columns in this window change depending on the object selected in the Navigator window.
- **Dependency.** Shows dependencies on sources, targets, mappings, and shortcuts for objects selected in either the Navigator or Main window.
- **Output.** Provides the output of tasks executed within the Repository Manager, such as creating a repository.

Figure 1-3 shows the windows in the Repository Manager:

Figure 1-3. Repository Manager Windows

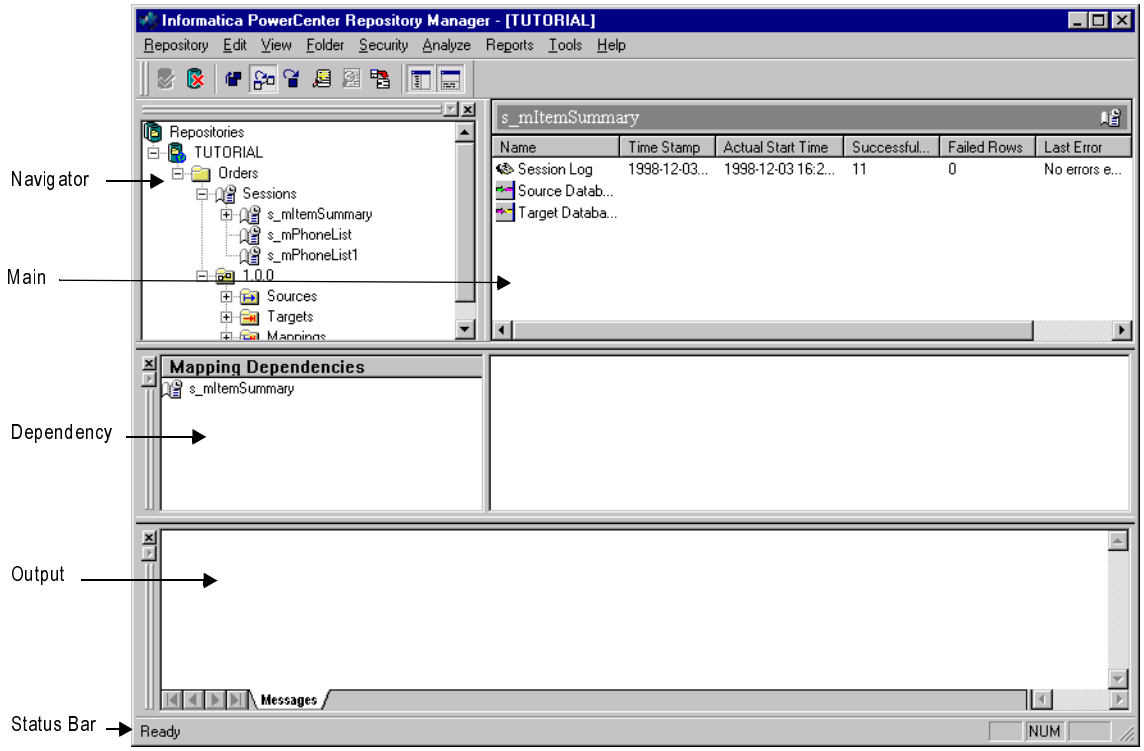
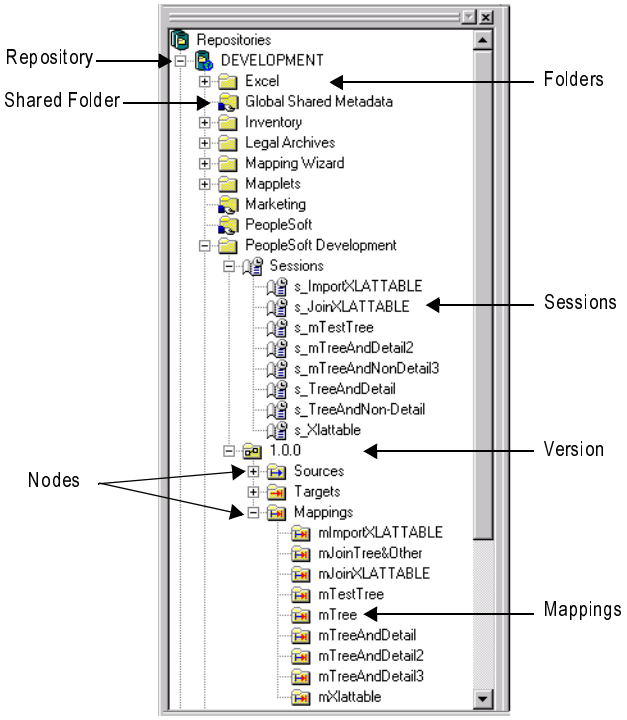


Figure 1-4 shows the Repository Manager Navigator:

Figure 1-4. Repository Manager Navigator



Repository Objects

You create repository objects using the Repository Manager, Designer, and Server Manager client tools. You can view the following objects in the Navigator window of the Repository Manager:

- **Source definitions.** Definitions of database objects (tables, views, synonyms) or files that provide source data.
- **Target definitions.** Definitions of database objects or files that contain the target data.
- **Multi-dimensional metadata.** Target definitions that are configured as cubes and dimensions.
- **Mappings.** A set of source and target definitions along with transformations containing business logic that you build into the transformation. These are the instructions that the Informatica Server uses to transform and move data.
- **Reusable transformations.** Transformations that you can use in multiple mappings.
- **Mapplets.** A set of transformations that you can use in multiple mappings.
- **Session and batches.** Sessions and batches store information about how and when the Informatica Server moves data. Each session corresponds to a single mapping. You can group several sessions together in a batch.

The Design Process

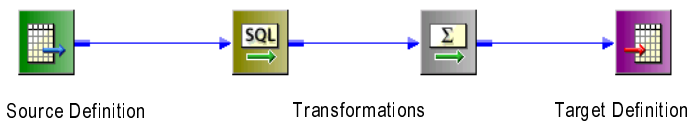
The goal of the design process is to create mappings that depict the flow of data between sources and targets, including changes made to the data before it reaches the targets. However, before you can create a mapping, you must first create or import source and target definitions. You might also want to create reusable objects such as reusable transformations or mapplets. For a list of objects you create in the Design process, see "Repository Objects" on page 9.

Perform the following design tasks in the Designer:

1. **Import source definitions.** Use the Source Analyzer to connect to the sources and import the source definitions.
2. **Create or import target definitions.** Use the Warehouse Designer to define relational, flat file, or XML targets to receive data from sources. You can import target definitions from a relational database, or you can manually create a target definition.
3. **Create the target tables.** If you add a target definition to the repository that does not exist in a relational database, you need to create target tables in your target database. You do this by generating and executing the necessary SQL code within the Warehouse Designer.
4. **Design mappings.** Once you have source and target definitions in the repository, you can create mappings in the Mapping Designer. A mapping is a set of source and target definitions linked by transformation objects that define the rules for data transformation. A transformation is an object that performs a specific function in a mapping, such as looking up data or performing aggregation.
5. **Create mapping objects.** Optionally, you can create reusable objects for use in multiple mappings. Use the Transformation Developer to create reusable transformations. Use the Mapplet Designer to create mapplets. A mapplet is a set of transformations that may contain sources and transformations.
6. **Debug mappings.** Use the Mapping Designer to debug a valid mapping to gain troubleshooting information about data and error conditions.
7. **Import and export repository objects.** You can import and export repository objects, such as sources, targets, transformations, mapplets, and mappings to archive or share metadata.

Figure 1-5 shows a sample mapping with source and target definitions and transformations:

Figure 1-5. Sample Mapping



For more information about the Designer, see the *Designer Guide*.

Designer Windows

You can display the following windows in the Designer:

- **Navigator.** Connect to repositories, and open folders within the Navigator. You can also copy objects and create shortcuts within the Navigator.
- **Workspace.** Open different tools in this window to create and edit repository objects such as sources, targets, mapplets, transformations, and mappings.
- **Output.** View details about tasks you perform, such as saving your work or validating a mapping.
- **Status bar.** Displays the status of the operation you perform.
- **Overview.** An optional window to simplify viewing a workspace that contains a large mapping or multiple objects.
- **Instance data.** View transformation data while you run the Debugger to debug a mapping.
- **Target data.** View target data while you run the Debugger to debug a mapping.

Figure 1-6 shows the default Designer windows:

Figure 1-6. Designer Windows

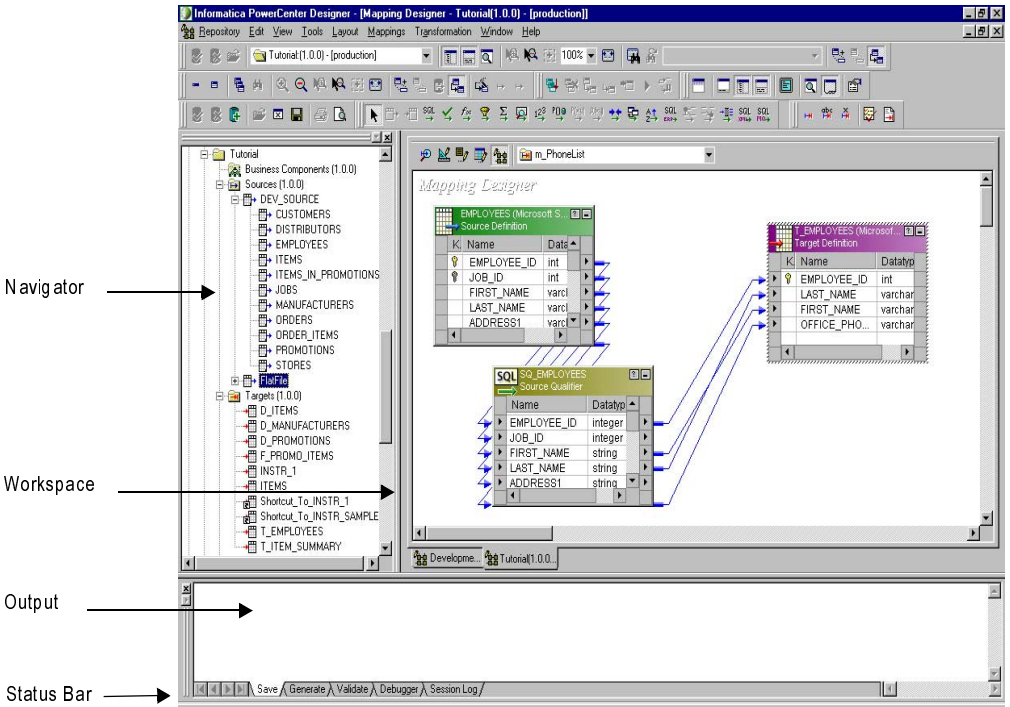


Figure 1-7 shows the Overview window:

Figure 1-7. Overview Window

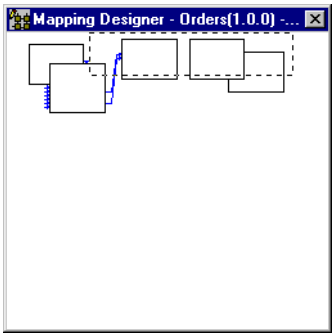
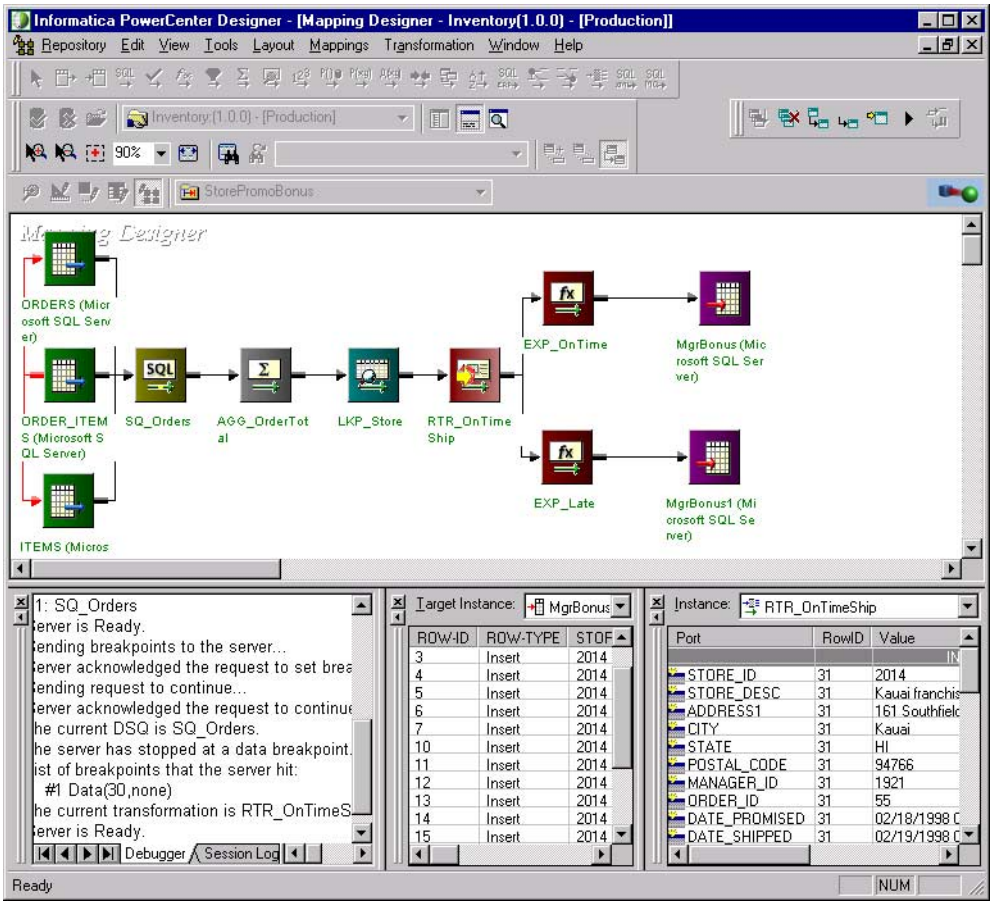


Figure 1-8 shows the Debugger window in the Designer:

Figure 1-8. Debugger Window



Loading Data

When you complete a mapping, you can use the Server Manager client application to create and schedule a session. During a session run, the Informatica Server uses the mapping logic to extract data, perform transformations, and load data to the target. When you create a session, the Informatica Server stores session information in the repository. When the session start time arrives, the Informatica Server retrieves the session information from the repository to extract, transform, and load the data.

Server Manager

Use the Server Manager to create, schedule, monitor, edit, copy, and abort sessions. You can group multiple sessions to run as a single unit, known as a batch. When you create a session, you select a valid mapping and configure other settings such as connections, error handling, and scheduling. You may also be able to override some transformation properties.

When you monitor sessions, the Server Manager displays status such as scheduled, completed, and failed sessions. It also displays some errors encountered while running the session. You can find a complete log of errors in the session log and server log files.

Before you create a session, you must configure the following connection information:

- **Informatica Server connection.** Register the Informatica Server with the repository before you can start it or create a session to run against it.
- **Database connections.** Create connections to source and target systems.
- **Other connections.** If you want to use external loaders or FTP, you configure access within the Server Manager.

For more information about configuring the Server Manager, see "Configuring the Server Manager" in the *Session and Server Guide.*

You can set the following properties when you create a session:

- **Informatica Server.** If you use PowerCenter, you can select an Informatica Server to run a session.
- **Source and target location.** Select a connection or specify a path for the source and target data.
- **Scheduling information.** Schedule the session to run on demand or on a repeating schedule.
- **Error handling.** Configure error handling parameters that determine how the Informatica Server behaves when it encounters errors.
- **Post-session email.** Send post-session email dependent on success and failure of session.
- **Pre- and post-session scripts.** Run shell commands before or after the session.

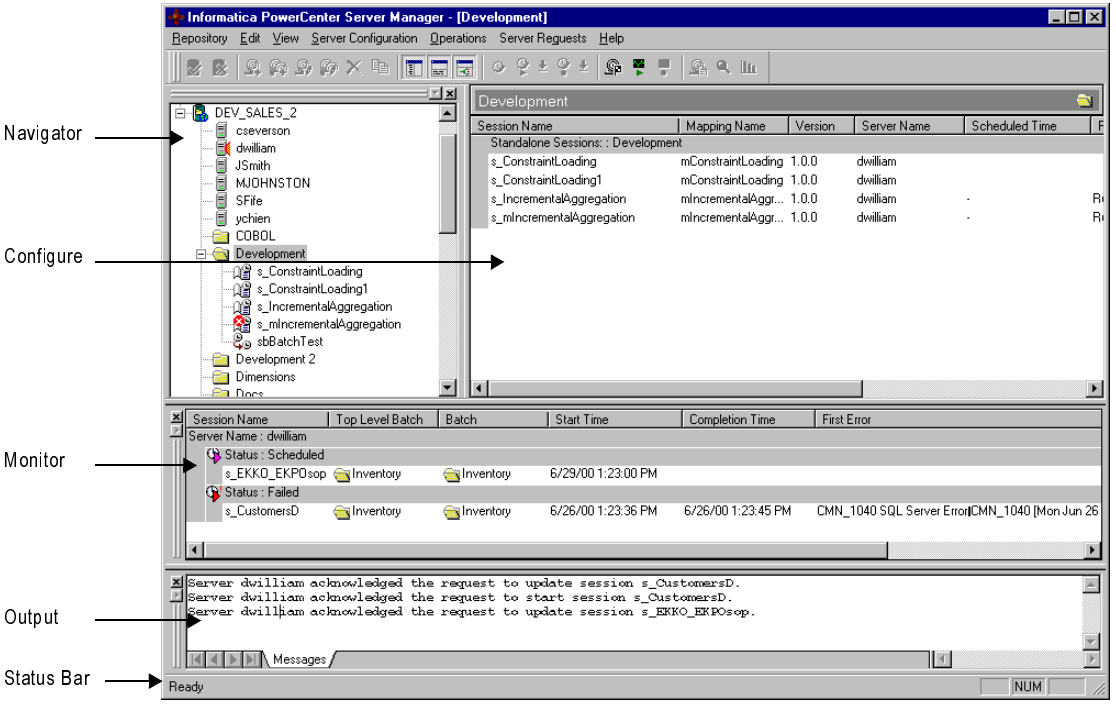
Server Manager Windows

The Server Manager displays the following windows:

- **Navigator.** View and select configured sessions.
- **Configure.** Create and edit sessions.
- **Monitor.** View information about running and completed sessions.
- **Output.** View messages from the Informatica Server.
- **Status.** Displays the status of the operation you perform.

Figure 1-9 shows the Server Manager windows:

Figure 1-9. Server Manager Windows



The Informatica Server

The Informatica Server can achieve high performance using symmetric multi-processing systems. The Informatica Server can start and run multiple sessions concurrently. It can also concurrently process partitions within a single session. When you partition a session, the Informatica Server creates multiple database connections to a single source and extracts a separate range of data for each connection, according to the properties you configure.

The Informatica Server manages the scheduling and execution of sessions and batches. It moves data from sources to targets based on mapping and session information stored in the repository.

The Informatica Server uses the following processes to run a session:

- **The Load Manager process.** Starts the session, creates the DTM process, and sends post-session email when the session completes.
- **The DTM process.** Creates threads to initialize the session, to read, write, and transform data, and to handle pre- and post-session operations.

Running a Session

When you schedule a session in the Server Manager, the Informatica Server stores the session name and execution time in shared memory. When a session starts, the Informatica Server connects to the repository to retrieve all session metadata to read, transform, and write data.

After the Load Manager performs validations for the session, it creates the DTM process. The DTM then creates the master thread, which creates and manages all other threads.

The master thread creates one mapping thread, one pre-session thread, and one post-session thread for each session. It creates a set of reader, transformation, and writer threads for each partition. When you partition a session, the threads for a source execute concurrently.

The reader thread extracts source data from relational tables, ERP objects, and flat files. As the reader reads data, it fills a buffer from session shared memory. When the reader fills a buffer, it passes it to the transformation thread.

The transformation thread transforms data received from buffers filled by the reader. It moves data from transformation to transformation using transformation caches when necessary.

The writer receives filled buffers from the transformation thread. It writes to the target database and commits data based on the commit type configured in the session properties.

For more information about the Informatica Server, see “Understanding the Server Architecture” in the *Session and Server Guide*.

Getting Started

Before you can begin using PowerMart or PowerCenter, you must create the environment and perform the following administration tasks to allow access to the repository and the Informatica Server:

1. **Configure the sources.** If you extract from relational sources, ask the database administrator to create user profiles with read access. This allows you to import source definitions into the repository and access the sources at runtime.

   If you extract from file sources, the files must be accessible to the Informatica Server and Client machines.

2. **Configure the targets.** Ask the database administrator to create user profiles with read and write access. This allows you to import target definitions into the repository and write to the targets at runtime.

   If the target database does not exist, create it using the database administration tools included with your RDBMS. After you create the target database, you can use the Designer to design and create target tables.

   For flat file targets, you need a target directory large enough to process the resulting files.

3. **Choose globalization settings and data movement modes.** The data movement mode you use depends on whether you want the Informatica Server to process single-byte data or multibyte character data. You select code pages for the repository, Informatica Client and Server.

4. **Create repository database.** Create a database for the repository. Users accessing the repository database need full rights in that database. If you upgrade the repository for a new version, you need database rights to drop or modify these tables.

5. **Install the Informatica Client.** Install the client software on a machine that accesses the sources, targets, and repository databases, as well as the Informatica Server.

6. **Install and configure the Informatica Server.** Install the Informatica Server on a Windows NT/2000 or UNIX system that accesses the sources, targets, and the repository database.

7. **Configure connectivity.** Configure network, native, and ODBC connectivity. Create ODBC data sources to connect to the Informatica Clients to the sources, targets, and repository databases. You must also have network connections between all databases and Informatica Servers.

8. **Create the repository.** After you configure source, target, and repository databases, and create the ODBC data sources needed to connect to them, you can create the metadata repository. Use the Repository Manager to create the metadata repository. The Repository Manager connects to the repository database and runs the SQL needed to create the repository tables. All the objects you create with PowerMart or PowerCenter are stored as metadata in the repository.

9. **Create repository users and groups.** Create groups and user profiles, then assign privileges and permissions that determine tasks that users can perform.

10. **Register the Informatica Server.** Before you can start the Informatica Server you must register the Informatica Server so the Server Manager can direct the Informatica Server to the repository.

For more information about these tasks, see the *Installation and Configuration Guide*.

Chapter 2

Tutorial Lessons 1-5

This chapter walks you through the process of creating a data warehouse. Topics include:

Overview

This tutorial walks you through the process of creating a data warehouse. The tutorial teaches you how to:

- Create a repository.
- Create users and groups.
- Add source definitions to the repository.
- Create targets and add their definitions to the repository.
- Map data between sources and targets.
- Instruct the Informatica Server to write data to targets.

In general, you can set your own pace for completing the tutorial. However, Informatica recommends completing an entire lesson in one sitting, since each lesson builds on a sequence of related tasks. After you finish the tutorial, you will have mastered the basics of creating a data warehouse.

For additional information, case studies, and updates on using Informatica products, see the Informatica on-line journal, the *Informatica Webzine*. You can access the webzine at http://www.informatica.com/webzine.

Before You Begin

Before starting the lessons in the basic tutorial, you must complete the preliminary tasks. For more information see “Getting Started” on page 16.

For a description of the process of creating a data warehouse, see “The Design Process” on page 10 and “Loading Data” on page 13.

Lesson 1: Creating a Repository

The first step in designing a data warehouse is creating the repository. Each repository contains metadata, information about repository objects, such as sources and targets. Repository metadata also contains instructions on how and when to read source data, transform it, and write it to targets. In the simplest possible repository, the metadata contains these descriptions:

- A single source, such as a table in a relational database
- A source qualifier transformation
- One target, perhaps another table in a different database

The metadata also includes mappings, identifying how each column in a target receives data from one or more columns in the source. For example, the mapping might specify that the CUSTOMER_ID column in a target table receives data from the CUSTOMER_ID column in a source table. The CUSTOMER_NAME column in the target might receive data from two columns, LAST_NAME and FIRST_NAME, two strings that are concatenated before writing this modified information to the target. You need to configure the Informatica Server to concatenate these strings before sending them to the target.

Metadata can include other types of information as well. For example, the metadata might contain transformation details that specify how the Informatica Server changes information before writing it to the target database. Common changes include:

- Formatting date values.
- Calculating the sum of all values found in a particular column.
- Replacing a numeric ID of a customer with their name stored in a separate table.

During this tutorial, you use three different tools:

- **The Repository Manager.** Use to create and maintain repositories.
- **The Designer.** Use to import sources, design targets, create mappings, and develop transformations.
- **The Server Manager.** Use to instruct the Informatica Server how and when to move data according to the instructions in mappings you create with the Designer.

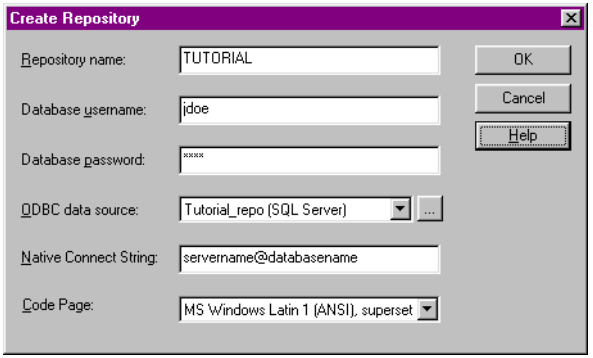
The first step in building a data warehouse is creating a repository in the Repository Manager.

To create the repository:

1. Launch the Repository Manager by choosing Programs-PowerCenter (or PowerMart) Client-Repository Manager from the Start Menu.
2. In the Repository Manager, choose Repository-Create Repository.

   Note: You must be running the Repository Manager in Administrator mode to see the Create Repository option on the menu. Administrator mode is the default when you install the program.

3. In the Create Repository dialog box, specify the name of the new repository, as well as the parameters needed to connect to the repository database through ODBC.



4. Enter TUTORIAL as the name of the new repository.

   Enter the username and password, followed by the name of the ODBC data source used for the repository and the native connect string.

   Table 2-1 lists the native connect string syntax to use for different database platforms:

Table 2-1. Native Connect String Syntax for Database Platforms

Database Platform	Native Connect String	Example
IBM DB2	*dbname*	mydatabase
Informix	*dbname@servername*	mydatabase@informix
Microsoft SQL Server	*servername@dbname*	sqlserver@mydatabase
Oracle	*dbname*.world (same as TNSNAMES entry)	oracle.world
Sybase	*servername@dbname*	sambrown@mydatabase

   For example, if your database platform is Sybase, the database name is TUTORIAL, and it runs on the MY_SERVER database server, you would enter MY_SERVER@TUTORIAL.

5. Select a code page for the repository.

   The repository code page defines the character set associated with the repository. After you create the repository, you cannot change the repository code page. The repository code page must be compatible with the Informatica Client and Server. The Repository Manager displays only code pages compatible with the Informatica Client. For more information on code page compatibility, see "Code Pages" in the *Installation and Configuration Guide*.

6. Click OK.

   If a repository already exists in the database, the Repository Manager warns you that if you want to continue, the new repository will replace the old one. However, since this is

the first time you have added a repository to the repository database, you should not see this warning.

7. Click OK when prompted to create the repository.

   The Repository Manager runs the DDL scripts to set up the repository. The results of this process displays in the Output window. You can open and close the Output window from the View menu.

8. Click OK to close the dialog box.

   Note: If you are using PowerCenter, you are prompted to make the repository a global repository. Click No to create a local repository. For details, see "PowerCenter Repositories" in the *Repository Guide*.

Creating Repository Users and Groups

You can create a repository user profile for everyone working in the repository, each with a separate username and password. You can also create user groups and assign each user to one or more groups. Then, grant repository privileges to each group, so users in the group can perform tasks within the repository (such as use the Designer or create sessions).

The repository user profile is not the same as the database user profile. While a particular user might not have access to a database as a *database* user, that same person can have privileges to a repository in the database as a *repository* user.

Informatica tools include two basic types of security:

- **Privileges.** Repository-wide security that controls which task or set of tasks a single user or group of users can access.
- **Permissions.** Security assigned to individual folders within the repository.

You can perform various tasks for each privilege.

Table 2-2 lists the most common tasks associated with each privilege:

Table 2-2. Repository Privileges

Privilege	Description
Use Designer	Can edit metadata, import and export objects in the Designer, with read and write permission at the folder-level.
Browse Repository	Can browse repository contents through the Repository Manager, add and remove reports, import, export, or remove the registry, and change user password.
Create Sessions and Batches	Can create, import, export, modify, start, stop, and delete sessions and batches through the Server Manager with folder-level read, write, and execute permissions. Can configure some connections used by the Informatica Server.

Table 2-2. Repository Privileges

Privilege	Description
Session Operator	Can use the command line program (*pmcmd*) to start sessions and batches. Can start, view, monitor, and stop sessions or batches with folder-level read permission and the Create Sessions and Batches privilege using the Server Manager.
Administer Repository	Can create, upgrade, back up, delete, and restore repositories. Can create and modify folders, create and modify users and groups, and assign privileges to users and groups.
Administer Server	Can configure connections to the Informatica Server, start and stop the Informatica Server through the Server Manager and *pmcmd*.
Super User	Can perform all tasks across all folders in the repository, including unlocking locks and managing global object permissions.

Privileges depend on your group membership. Every repository user belongs to at least one group. For example, the user who administers the repository belongs to the Administrators group, while developers belong to the Designers group. By default, you receive the privileges assigned to your group. While it is most common to assign privileges by group, the repository administrator, who has either Super User *or* Administer Repository privileges, can also grant privileges to individual users.

As an administrator you can perform the following tasks:

- Create groups
- Assign privileges to groups
- Create users and assign them to groups

When you create a repository, the Repository Manager creates two default groups: Administrators and Public. Within the Administrators group, the Repository Manager creates two default users: Administrator (password *Administrator*) and a username equivalent to the database username (and password). The first time you log into a new repository, you use the Administrator user profile. You can then change the password and begin adding new users and groups. You cannot delete the Administrator user.

When you assign group memberships, user privileges are affected in following ways:

- Each user inherits all group privileges.
- Each user belongs to one or more groups and inherit the privileges of each group.
- Each user gains and loses privileges when moved from one group to another.

However, if a group is removed from the repository, any users in the group retain the privileges they inherited from the group until you revoke them for the individual users.

In this tutorial, you create two groups:

- **Managers.** Repository users that lead data warehouse developers and administer the Informatica Server. They are super users.
- **Designers.** Repository users that create mappings and run sessions.

To create these users, you need to connect to the repository.

If you are already connected to the repository, disconnect and connect again to log in as the Administrator.

To connect to the repository:

1. Launch Repository Manager, and click the Connect button.

   A list of all repositories appears in the Navigator.

2. Double-click the TUTORIAL repository.

3. Enter Administrator as the username, and Administrator as the password.

4. Click Connect.

   The dialog box expands to enter additional information.

5. Select the ODBC data source corresponding to the repository database.

6. Enter the database username and password needed to connect to this database.

   Enter the native connect string for your database. See Table 2-1 on page 22 for the native connect string for your database platform.

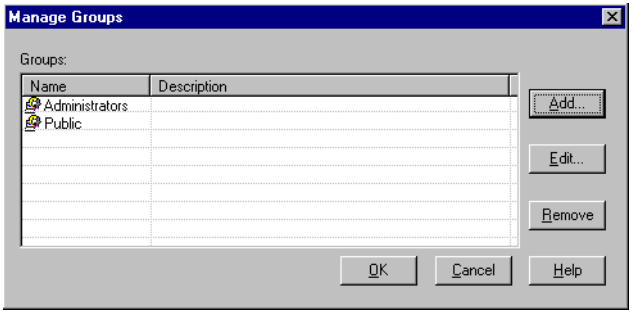
7. Click Connect.

   You are now connected to the repository as the user Administrator.

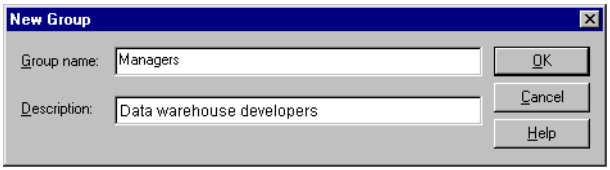
To create the Managers group:

1. Select the repository in the Navigator, and choose Security-Manage Groups.

   The Manage Groups dialog box lists all existing groups in the repository. Since you have not created any new groups yet, only the two default groups, Administrators and Public, appear in the list. You cannot edit or remove these groups.

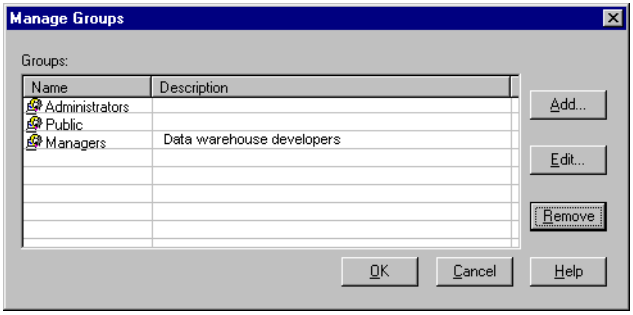


2. Click Add.



3. Type Managers for the name of the new group, and Data warehouse developers as the description.
4. Click OK.

   The new group, Managers, appears in the list.



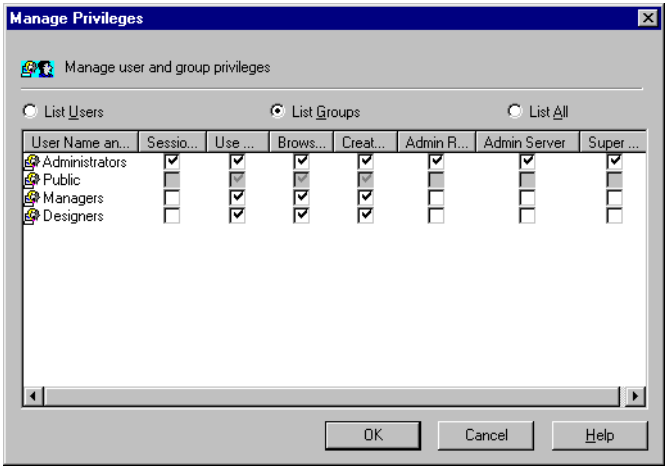
5. Add another new group with Designers as the group name, and Repository developers as the description.
6. Click OK to return to the Repository Manager.

   The next step is to grant privileges to these two groups.

To assign privileges to the Managers group:

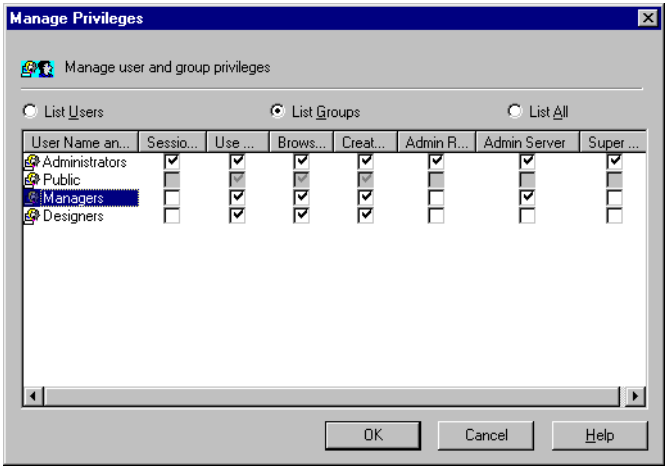
1. Choose Security-Manage Privileges.

The Manage Privileges dialog box displays the privileges currently assigned to users and groups.



The Administrators group currently has the Administer Server privilege.

2. Assign the Administer Server privilege to the Managers group.



Next, assign privileges to the Designers group.

3. Look at the Use Designer column on the Privileges list.

   Both the Managers and the Designers groups have this default privilege. The following default privileges are granted to each new group and new user:

   - Use Designer
   - Create Sessions and Batches
   - Browse Repository

4. Assign the Session Operator privilege to the Designers group.

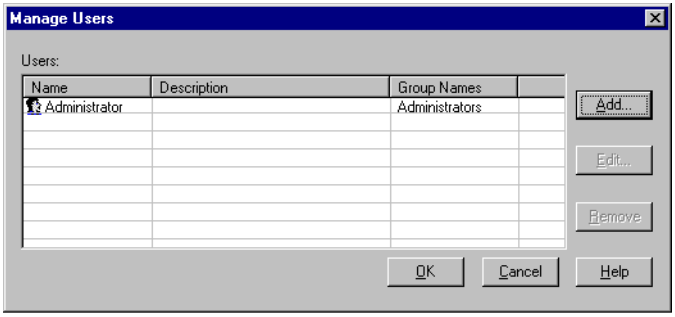
The Designers group now can create a session in any folder for which they have read permission.

The final step is to create a new user, jdoe, and add this user to the Administrators group. You will use this user profile throughout the rest of this tutorial.

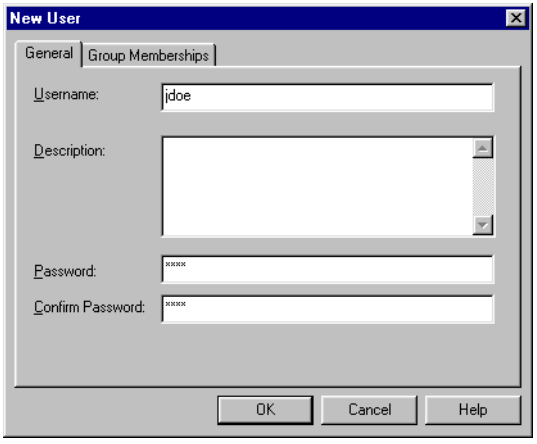
To create a new user:

1. Choose Security-Manage Users.

   A new dialog box appears, listing all the users in the repository.

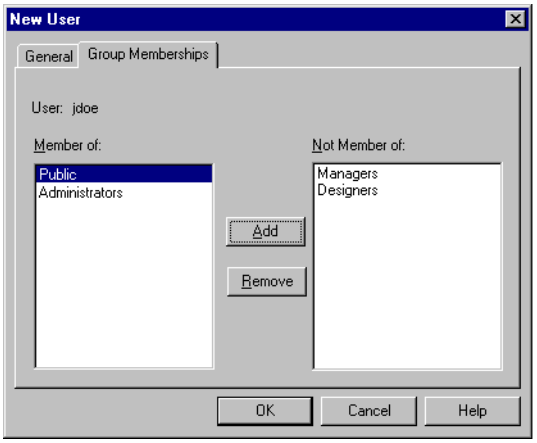


2. Click Add.



3. In the New User dialog box, enter jdoe as the Username.
4. In both the Password and Confirm Password fields, enter jdoe.
5. Click the Group Memberships tab.

6. Select Administrators, the group for jdoe.



7. Click OK. The user, jdoe, now has all the privileges associated with the Administrators group.

Creating a Folder

Folders provide a way to organize and store all metadata in the repository, including mappings, schemas, and sessions. Folders are designed to be flexible, to help you organize your data warehouse logically. Each folder has a set of properties you can configure to define how users access the folder. For example, you can create a folder that allows all repository users to see objects within the folder, but not to edit them. Or you can create a folder that allows users to share objects within the folder.

Shared Folders

When you create a folder, you can configure it as a shared folder. Shared folders allow users to create shortcuts to objects in the folder. If you have reusable transformation that you want to use in several mappings or across multiple folders, you can place the object in a shared folder. For example, you may have a reusable Expression transformation that calculates sales commissions. You can then use the object in other folders by creating a shortcut to the object.

Folder Permissions

Permissions allow repository users to perform tasks within a folder. With folder permissions, you can control user access to the folder, and the tasks you permit them to perform.

Folder permissions work closely with repository privileges. Privileges grant access to specific tasks while permissions grant access to specific folders with read, write, and execute qualifiers. However, any user with the Super User privilege can perform all tasks across all folders in the repository. Folders have the following types of permissions:

- **Read permission.** Allows you to view the folder as well as objects in the folder.

- **Write permission.** Allows you to create or edit objects in the folder.
- **Execute permission.** Allows you to execute or schedule a session or batch in the folder.

Permission Levels

You can grant folder permissions on the following levels of security:

- **Owner.** The owner of the folder.
- **Owner's Group.** Each user in the owner's repository group. If the owner belongs to more than one group, you must select one of those groups for the owner's group.
- **Repository.** All groups and users in the repository.

Each permission level includes the permissions of the level above it. For more information on assigning folder-level permissions, see "Repository Security" in the *Repository Guide*.

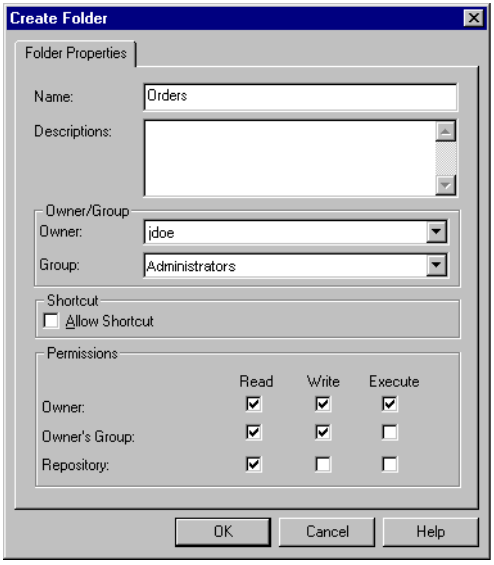
Folders in this Tutorial

For this tutorial, you create a single folder, Orders. The metadata in this folder contains information about orders, customers, and products.

To create a new folder:

1. Choose Folder-Create.
2. Enter Orders as the name of the folder.
3. Select jdoe as the owner of the folder.

   Because jdoe is a member of the Administrators group, the Owner's Group defaults to that group. Users that belong to more than one group must select a group from the list.



4. Click OK to create the folder.

   The new folder now appears as part of the TUTORIAL repository.

5. Exit the Repository Manager.

What Comes Next

Now that you have created the repository, you can start adding source and target definitions to it. However, you first need to create the actual source tables that contain the sample data.

Lesson 2: Importing Source and Target Definitions

In this lesson, you create source tables and add source definitions to the repository. You also create target tables and target definitions to the repository.

Creating Source Tables

With most data warehouses, you already have existing source tables or flat files. In this tutorial, however, you run a SQL script in the Warehouse Designer to create sample source tables. The SQL script creates sources with ASCII 7 bit table names and data.

To create the sample tables:

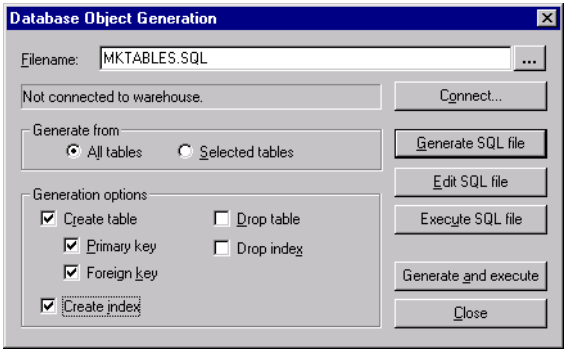
1. Launch the Designer, double-click the icon for the TUTORIAL repository, and log into the repository.

   Use the jdoe/jdoe user profile to open the connection.

2. Double-click the Orders folder.

3. Choose Tools-Warehouse Designer to switch to the Warehouse Designer.

4. Choose Targets-Generate/Execute SQL.



   The Database Object Generation dialog box appears. This dialog box gives you several options for creating tables.

5. Click the Connect button.

6. Select the ODBC data source you created for connecting to the source database (not the repository).

7. Enter the database username and password and click the Connect button.

   You now have an open connection to the source database. You know that you are connected when the Connect button reads Disconnect, and the ODBC name of the source database appears in the dialog box.

8. Make sure the Output window is open at the bottom of the Designer.

If it is not open, click the Toggle Output Window button on the View toolbar, or choose View-Output.

9. Click the Browse button to find the SQL file. The SQL file is installed in the Tutorial folder in the program installation directory.

10. Select the SQL file appropriate to the database platform you are using. Click Open.

Platform	File
Informix	SMPL_INF.SQL
Microsoft SQL Server	SMPL_MS.SQL
Oracle	SMPL_ORA.SQL
Sybase SQL Server	SMPL_SYB.SQL
DB2	SMPL_DB2.SQL

Alternatively, you can type in the file name and path of the .SQL file.

Note: Database objects created in Informix databases have shorter names than those created in other types of databases. For example, the name of the table ITEMS_IN_PROMOTIONS, created in these scripts, is shortened to ITEMS_IN_PROMO.

11. Click Execute SQL file.

    The database now executes the SQL script to create the sample source database objects. While the script is running, the Output window displays the progress.

12. When the script completes, click Disconnect and then click Close.

Creating Source Definitions

Now that you have added the source tables containing sample data to the source database, you are ready to add the definitions of these sources to your metadata. The repository contains a description of source tables, not the actual records contained in them. After you add these source definitions to the repository, you can use them in a mapping.

To analyze the sample sources:

1. In the Designer, click the Source Analyzer button on the workbook, or choose Tools-Source Analyzer.

2. Double-click the Orders folder to view its contents.

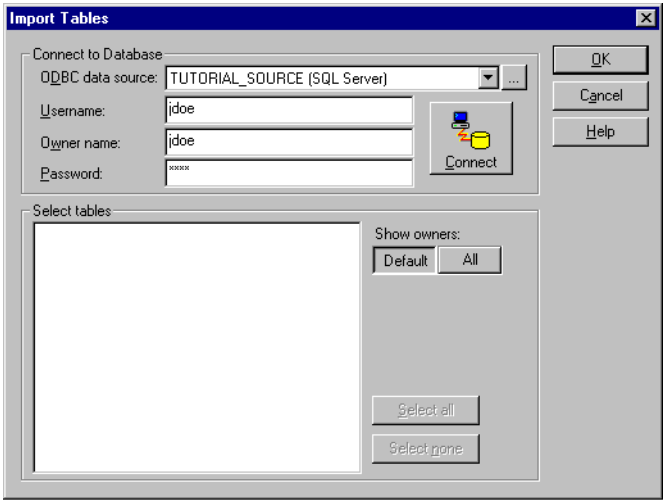
   Every folder contains sections for sources, targets, schemas, mappings, and reusable transformations.

3. Choose Sources-Import from Database.

4. Select the ODBC data source to access the database containing the source tables.

5. Enter the username and password for connecting to this database, as well as the name of the source table owner.

   Note: In Oracle, the owner name is the same as the username. Make sure that the owner name is in all caps (for example, JDOE).

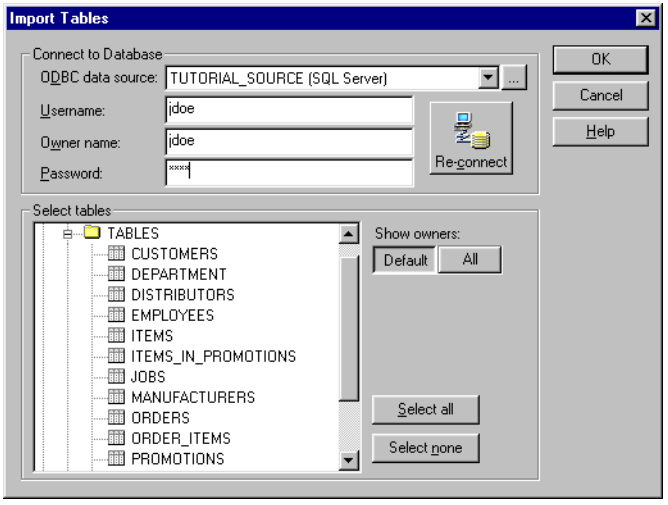


6. Click Connect.

7. In the Select tables list, expand the database owner and the TABLES heading.

   Alternatively, you can click the All button to see all tables in the source database.

   You should now see a list of all the tables you created by running the SQL script.

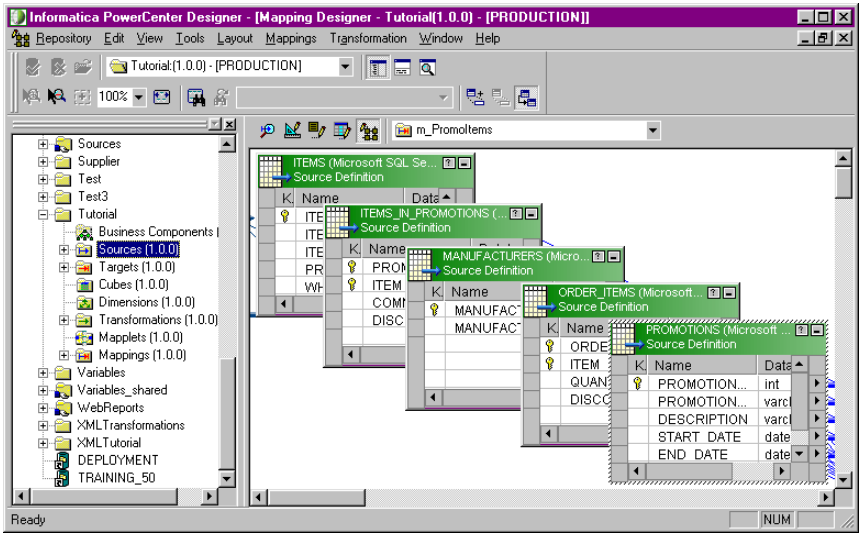


8. Click the CUSTOMERS table, hold down the Shift key, and then click the STORES table.

You may need to scroll down the list of tables to reach STORES. You should now have all the tables selected. If not, click the Select All button.

9. Click OK.

   The Designer displays the newly imported sources in the workbook. You can select Layout-Zoom Out to fit all the definitions in the workbook.



A new entry appears under the Sources node in the Orders folder. This new entry has the same name as the ODBC data source to access the sources you just imported. If you double-click the name of the ODBC data source, you see the list of all the imported sources.

Viewing Source Definitions

You can view details for each source definition.

To view a source definition:

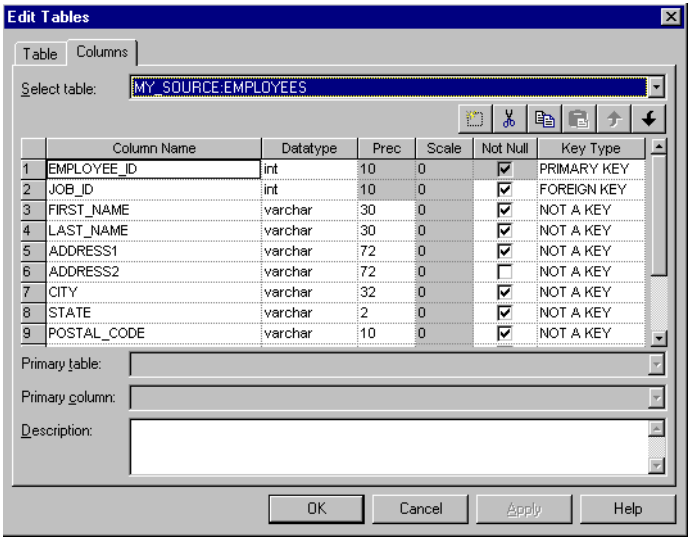
1. Double-click the title bar of the source definition for the EMPLOYEES table.

   The Edit Tables dialog box opens and displays all the properties of this source definition. The Table tab shows the name of the table, business name, owner name, and the database type. You can add a comment in the Description section.

   Note: To change the source table name, click Rename.

2. Click the Columns tab.

The Columns tab displays the column descriptions for the source. You can modify the source definition, change or delete columns. Any changes you make in this dialog box affect the source definition, not the source.



3. Click Cancel to close the dialog box.

Creating Target Definitions

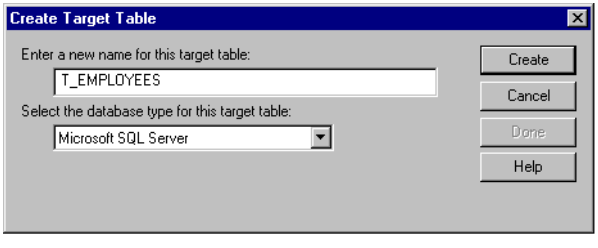
The next step is to create the metadata for the target tables in the repository. The actual table that the target definition describes does not exist yet.

Target definitions can describe targets that already exist, or those you plan to create later. You can modify targets before adding them to the database or flat file system. After creating a target definition, run the necessary SQL code, or create the new flat file.

Create a simple target table that is a copy of the EMPLOYEES table you created in the source database.

To create the first target definition:

1. In the Designer, switch to the Warehouse Designer.
2. Choose Targets-Create.



3. In the Create Target Table dialog box, enter T_EMPLOYEES, select the correct database platform, and click Create.

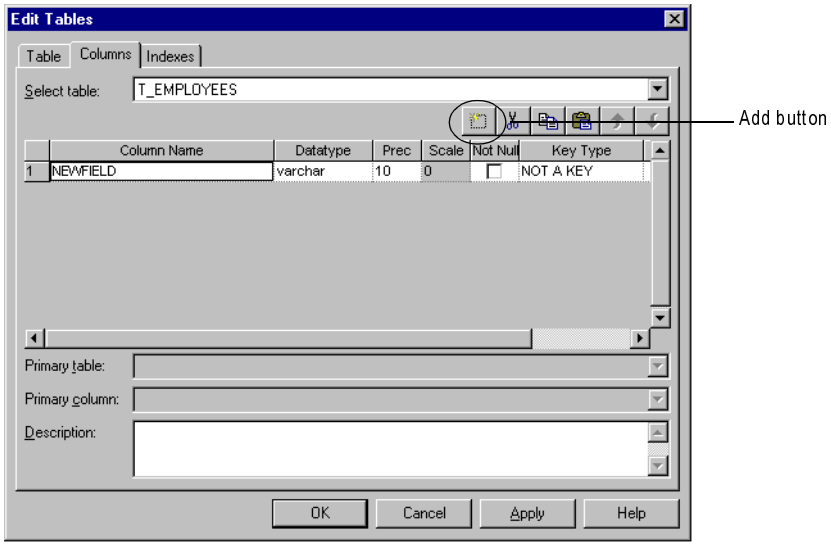
   A new target, T_EMPLOYEES, appears in the workbook. It does not contain any columns yet, so you add columns to the target in the next steps.

4. Click Done to close the Create Target dialog box, then double-click the title bar of T_EMPLOYEES.

   The Edit Tables dialog box appears.

5. Click the Columns tab, then click the Add button.

   A new column appears in the list with the default name NEWFIELD. You can give this column a different name, as well as make other changes to it.



6. Select the column name and enter EMPLOYEE_ID.

7. Select Integer for the column datatype.

   If your database does not have an Integer datatype, use Number(p,s) with a precision of 15 and a scale of 0.

8. In the Key Type section, designate this column as a primary key.

   The primary key cannot accept null values. The Designer automatically selects Not Null and disables the Not Null option. You now have a column ready to receive data from the EMPLOYEE_ID column in the EMPLOYEES source table. Next, you add some more columns to the table. Click the Add button and configure them as you did with EMPLOYEE_ID.

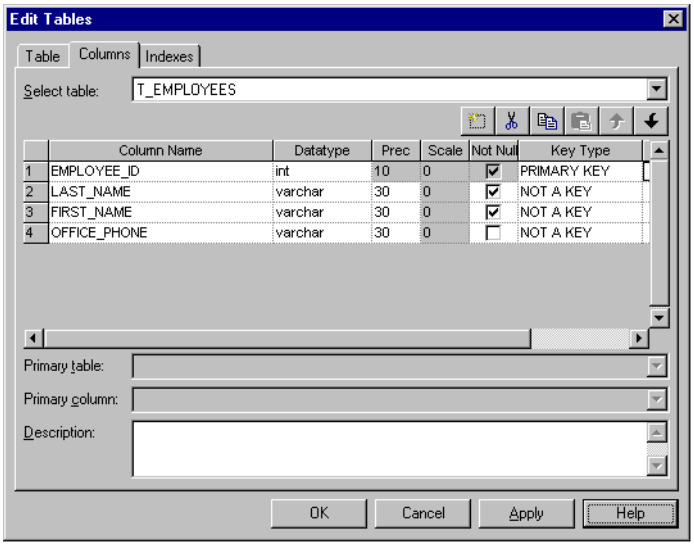
   Note: If you want to add a business name for any column, scroll to the right and enter it.

9. Add the following columns to T_EMPLOYEES:

Name	Datatype	Precision	Not Null
LAST_NAME	Varchar*	30	Yes
FIRST_NAME	Varchar*	30	Yes
OFFICE_PHONE	Varchar*	30	No

** Use Varchar2 in Oracle*

Do not modify the Key settings for these columns. When you finish, the target is configured as follows:



10. Click OK to close the dialog box.

11. Choose Repository-Save.

Creating Targets

You can use the Warehouse Designer to run an existing SQL script to create target tables.

Note: When the Designer generates SQL script to create a target table, the DDL code can start with instructions to drop a table with a name that you assign to the new target. This can occur if you select the Drop Table option to modify targets. If the target database already contains tables before you start designing your data warehouse, make sure that an existing table does not contain the same name as the target that you plan to create. If the table exists, you lose the existing table and data.

To create the target T_EMPLOYEES table:

1. Select the target definition for T_EMPLOYEES.
2. Choose Targets-Generate/Execute SQL.

   The dialog box to run the SQL script appears.
3. In the Filename field, enter `C:\[your installation directory]\MKT_EMP.SQL`

   If you installed the client software in a different location, enter the appropriate drive letter and directory.
4. If you are connected to the source database from the previous lesson, click Disconnect, then Connect.
5. Select the ODBC data source to connect to the target database.
6. Enter the necessary username and password, then click Connect.
7. Select the Create Table, Drop Table, and Primary Key options.
8. Click the Generate and Execute button.

   The Designer runs the DDL code needed to create T_EMPLOYEES. If you want to review the actual code, click Edit SQL file to open the MKT_EMP.SQL file.
9. Click Close to exit.

What Comes Next

In the next lesson, you design the dataflow between sources and targets, specifying how each column in the target table receives data from a counterpart column in the source table.

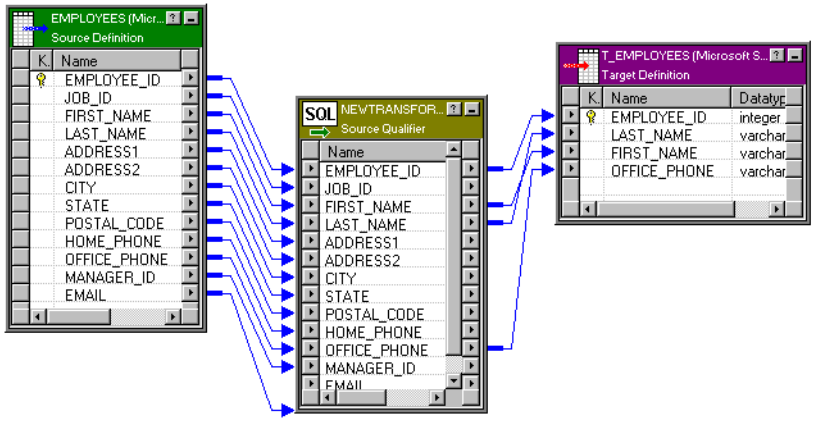
Lesson 3: Creating a Pass-Through Mapping

In the previous lessons, you added source and target definitions to your repository. You generated and ran the SQL code to create source and target tables.

The next step is to create a mapping to depict the flow of data between sources and targets. To create and edit mappings, you use the Mapping Designer tool in the Designer. The mapping interface in the Designer is *component-based*, meaning that it shows you every step in the process of moving data between sources and targets. In addition, *transformations* depict how the Informatica Server modifies data before it loads a target.

Figure 2-1 illustrates a mapping between a source and a target with a Source Qualifier transformation:

Figure 2-1. Simple Mapping



In your first mapping, you map each column in the source EMPLOYEES table to its counterpart in the target T_EMPLOYEES table.

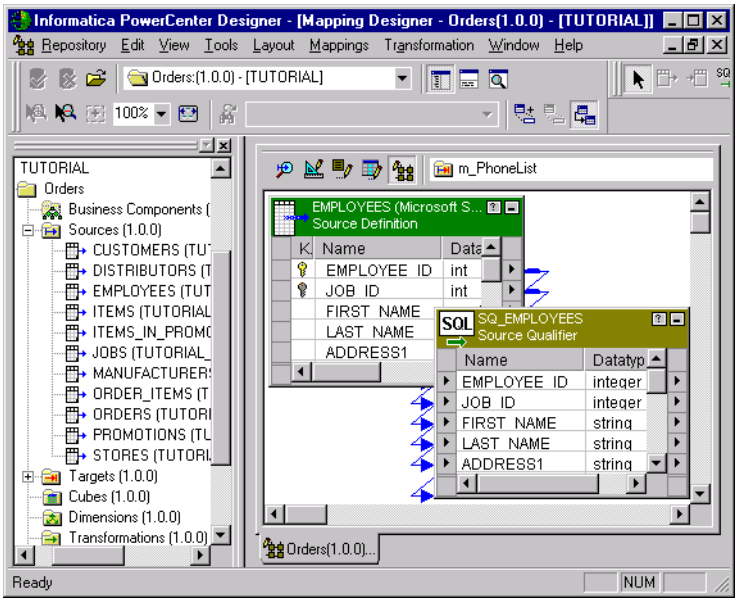
To create a mapping:

1. Switch to the Mapping Designer.
2. Choose Mappings-Create.

   While the workspace may appear blank, in fact it contains a new mapping without any sources, targets, or transformations.
3. In the Mapping Name dialog box, enter m_PhoneList as the name of the new mapping and click OK.

   The naming convention for mappings is m_*MappingName*.
4. In the Navigator, under the TUTORIAL repository and Orders folder, click the Sources node to view source definitions added to the repository.

5. Click the icon representing the EMPLOYEES source and drag it into the workbook.



The source definition appears in the workspace. The Designer automatically connects a Source Qualifier transformation to the source definition. After you add the target definition, you connect the Source Qualifier to the target.

6. Click the Targets icon in the Navigator to open the list of all target definitions.

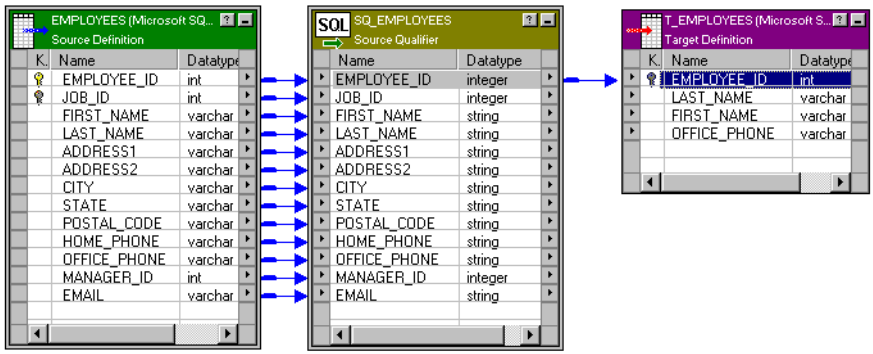
7. Click and drag the icon for the T_EMPLOYEES target into the workspace.

   The target definition appears. The final step is connecting the Source Qualifier to this target definition.

To connect the Source Qualifier to the target definition:

1. Click once in the middle of the EMPLOYEE_ID row in the Source Qualifier. Hold down the mouse button, and drag the cursor to the EMPLOYEE_ID row in the target. Then release the mouse button.

An arrow (called a connector) now appears between the two columns.



2. Repeat step 1 to map the LAST_NAME, FIRST_NAME, and OFFICE_PHONE columns between the Source Qualifier transformation and the T_EMPLOYEES target table.

   If you accidentally connect the wrong columns, select the connector and press the Delete key.

3. Choose Repository-Save.

To arrange the mapping:

1. Choose Layout-Arrange.
2. In the Select Targets dialog box, select the T_EMPLOYEES table and click OK.

   The Designer now rearranges the source, Source Qualifier, and target from left to right, making it easier to see how one column maps to another. However, you still may not be able to see all the columns in the Source Qualifier and the source.

3. Drag the lower edge of the source and Source Qualifier windows until all columns display.

If you examine the mapping, you see that data flows from the source definition to the Source Qualifier transformation to the target definition through a series of input and output ports.

The source provides information, so it contains only output ports, one for each column. Each output port is connected to a corresponding input port in the Source Qualifier. The Source Qualifier transformation contains both input and output ports. The target only contains input ports.

When you design mappings containing different types of transformations, you can configure transformation ports as inputs, outputs, or both. You can rename ports and change their datatypes.

What Comes Next

In Lesson 4, you will create a more complex mapping to learn how to modify data before it reaches the target.

Lesson 4: Creating a Mapping with Aggregate Values

In this lesson, you create a mapping that contains a source, transformations, and a target.

A transformation is any part of a mapping that generates or modifies data. Every mapping includes a Source Qualifier transformation, representing all the columns of information read from a source and temporarily stored by the Informatica Server. In addition, you can add transformations such as a calculating sum, looking up a value, or generating a unique ID that modify information before it reaches the target.

Table 2-3 lists the transformations displayed in the Transformations toolbar in the Designer:

Table 2-3. Transformations Descriptions

Transformation	Description
Advanced External Procedure	Calls a procedure in a shared library or in the COM layer of Windows NT/2000.
Aggregator	Performs aggregate calculations.
ERP Source Qualifier	Represents the data that the Informatica Server reads from an ERP source when it runs a session. This transformation is available with PowerConnect products.
Expression	Calculates a value.
External Procedure	Calls a procedure in a shared library or in the COM layer of Windows NT/2000.
Filter	Filters records.
Input	Defines mapplet input rows. Available only in the Mapplet Designer.
Joiner	Joins records from different databases or flat file systems.
Lookup	Looks up values.
MQ Source Qualifier	Represents the data that the Informatica Server reads from an IBM MQSeries message queue when it runs a session. This transformation is available with PowerConnect for IBM MQSeries.
Normalizer	Normalizes records, including those read from COBOL sources.
Output	Defines mapplet output rows. Available only in the Mapplet Designer.
Rank	Limits records to a top or bottom range.
Router	Routes data into multiple transformations based on a group expression.
Sequence Generator	Generates primary keys.
Source Qualifier	Represents the rows that the Informatica Server reads from a relational or flat file source when it runs a session.

Table 2-3. Transformations Descriptions

Transformation	Description
Stored Procedure	Calls a stored procedure.
Update Strategy	Determines whether to insert, delete, update, or reject records.
XML Source Qualifier	Represents the rows that the Informatica Server reads from an XML source when it runs a session.

Figure 2-2 shows the Transformation toolbar:

Figure 2-2. Transformation Toolbar

For many transformations, you can also create a reusable transformation that serves as a template for other transformations.

In this lesson, you create a mappping with a source, a new target, and these transformations:

- **Lookup transformation.** Finds the name of a manufacturer.
- **Aggregator transformation.** Calculates the maximum, minimum, and average price of items from each manufacturer.
- **Expression transformation.** Calculates the average profit of items, based on the average price.

Creating a New Target

Before creating the mapping, you need to design a new target that holds summary data about products from various manufacturers. This table includes the maximum and minimum price for products from a given manufacturer, an average price, and an average profit.

To design the new target:

1. In the Designer, choose Window-New Window.
2. In the new workbook, click the tools icons to switch to the Warehouse Designer.
3. Choose Targets-Create.
4. Enter T_ITEM_SUMMARY as the name of the new target table, select the correct database platform, and click Create.
5. Double-click the title bar of the new target.

6. Add the following columns to T_ITEM_SUMMARY:

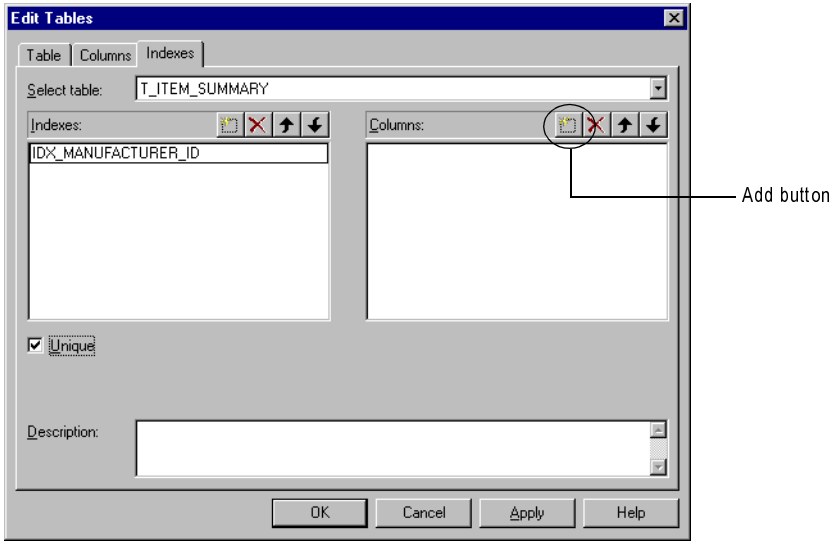
Column	Datatype	Precision	Not Null	Key Type
MANUFACTURER_ID	Integer		Selected	Primary key
MANUFACTURER_NAME	Varchar	72		
MAX_PRICE	Money		Selected	
MIN_PRICE	Money		Selected	
AVG_PRICE	Money		Selected	
AVG_PROFIT	Money		Selected	

If the Money datatype does not exist in your database, use Number (15,2) or Decimal (15,2). If the Integer datatype does not exist, use Number (10,0) or Double.

7. Click Apply.
8. Select the Indexes tab to add an index to the target table.

   If your target exists in an Oracle database, skip to step 14, since you cannot add an index to a column that already has the PRIMARY KEY constraint added to it.

9. In the Indexes section, click Add.
10. Enter IDX_MANUFACTURER_ID as the name of the new index, then press Enter.
11. Select the Unique index option.



12. In the Columns section, click Add.
13. Select MANUFACTURER_ID and click OK.
14. Click OK to close this dialog box, then choose Repository-Save.

The next step is to generate and execute the SQL script to create this new target table.

To create T_ITEM_SUMMARY:

1. Select the table T_ITEM_SUMMARY, then choose Targets-Generate/Execute SQL.
2. In the Database Object Generation dialog box, connect to the target database.
3. Choose Generate from Selected tables, and select the Create Table, Primary Key, Create Index options.

   Leave the other options unchanged.
4. Click Generate and Execute.

   The Designer runs the SQL script to create T_ITEM_SUMMARY.
5. Click Close.

Creating a New Mapping

The next step is to create a mapping that populates this new target. This mapping performs the following tasks:

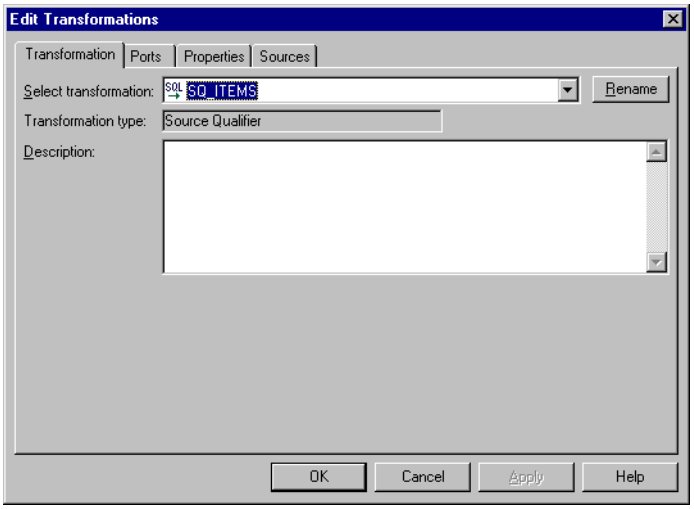
- Finds the most expensive and least expensive item in your inventory for each manufacturer.
- Calculates the average price and profitability of all items from a given manufacturer.

To provide this information, the mapping has to perform both simple and aggregate calculations. For example, you use the MIN and MAX functions to find the most and least expensive items from each manufacturer.

To create the new mapping:

1. Switch from the Warehouse Designer to the Mapping Designer.
2. Choose Mappings-Create.
3. When prompted to close the current mapping, click Yes.
4. In the Mapping Name dialog box, enter m_ItemSummary as the name of this new mapping.
5. From the list of sources in the TUTORIAL folder, click and drag the ITEMS table into the mapping.

6. Double-click the title bar of the Source Qualifier.



7. In the Edit Transformations dialog box, click Rename.
8. Enter SQ_AllData as the new name of the Source Qualifier transformation and click OK.
9. From the Targets node, click and drag the T_ITEM_SUMMARY target into the mapping.

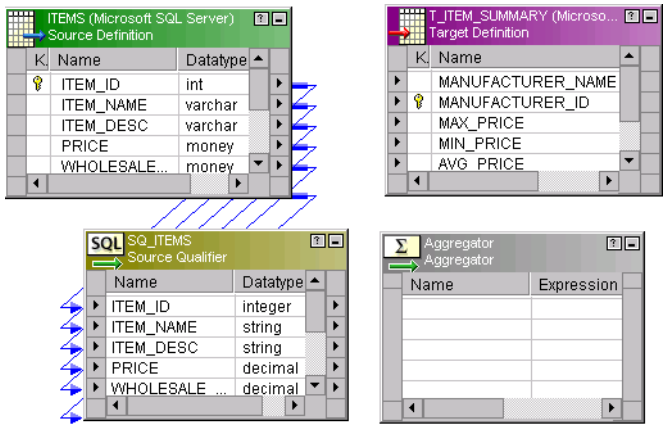
Creating an Aggregator Transformation

One of the first requirements in the new mapping is to add an Aggregator transformation to calculate the average, maximum, and minimum prices of items from each manufacturer.

To add the Aggregator transformation:

1. Click the Aggregator button on the Transformations toolbar.
2. Click in the space between the Source Qualifier and the T_ITEM_SUMMARY target.

The Designer adds a new Aggregator transformation to the mapping.



3. Choose Layout-Link Columns.

   When you drag column data from one transformation to another, the Designer copies the port description and links the original port to its copy. The port you copy must have both an input and an output.

   If you choose Layout-Copy Columns, every port you drag is copied, not linked.

4. From the Source Qualifier, click and drag the PRICE column into the Aggregator transformation.

   A copy of the PRICE column now appears in the new Aggregator transformation. The new column has the same name and datatype as the original version.

   The Aggregator transformation now receives data from the PRICE column. You need this information to calculate the maximum, minimum, and average product price for each manufacturer.

5. Click and drag the MANUFACTURER_ID column into the Aggregator transformation.

   You need another input, MANUFACTURER_ID, to provide the information for the equivalent of a GROUP BY statement. By adding this second input, you can define the groups (in this case, manufacturers) for the aggregate calculation. This organizes the data by manufacturer.

6. Double-click the title bar of the transformation.

7. Change the name of this transformation to AGG_PriceCalculations.

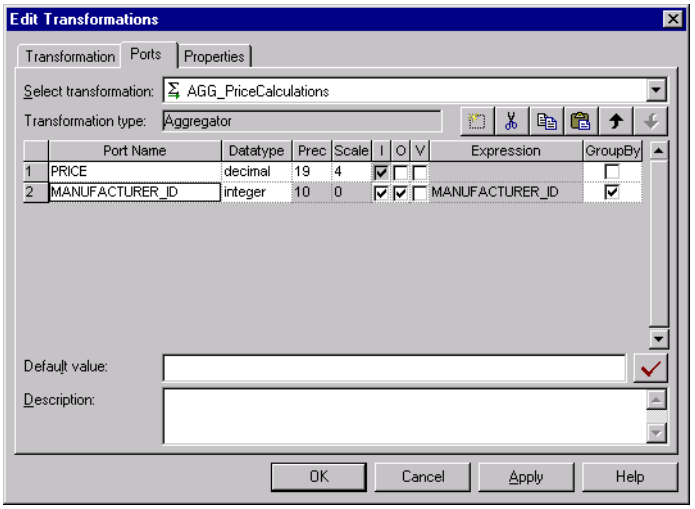
   The naming convention for Aggregator transformations is AGG_*TransformationName.*

8. Click the Ports tab.

9. Clear the Output (O) box for PRICE.

   You want to use this port as an input (I) only, not as an output (O). Later, you use data from PRICE to calculate the average, maximum, and minimum prices.

10. Select the Group By option for the MANUFACTURER_ID column.

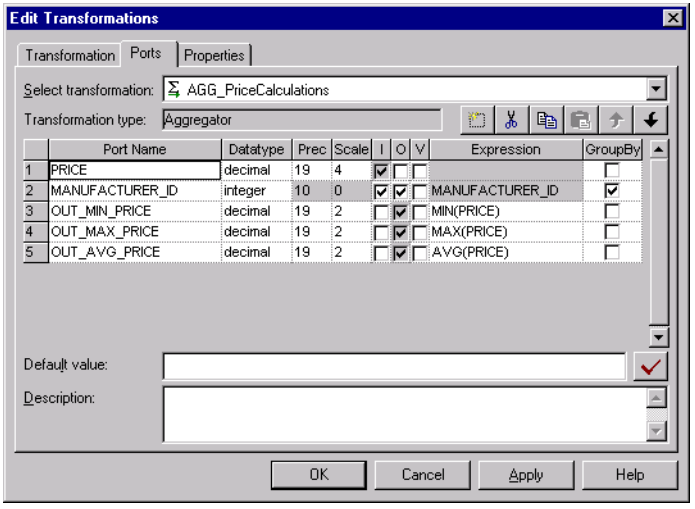


11. Click Add three times to add three new ports.

12. Configure the ports:

Name	Datatype	Precision	Scale	I	O	V
OUT_MIN_PRICE	Decimal	19	2	No	Yes	No
OUT_MAX_PRICE	Decimal	19	2	No	Yes	No
OUT_AVG_PRICE	Decimal	19	2	No	Yes	No

You can also select each port and click the Up and Down buttons to position the output ports after the input ports in this list.



13. Click Apply to save your changes to this point.

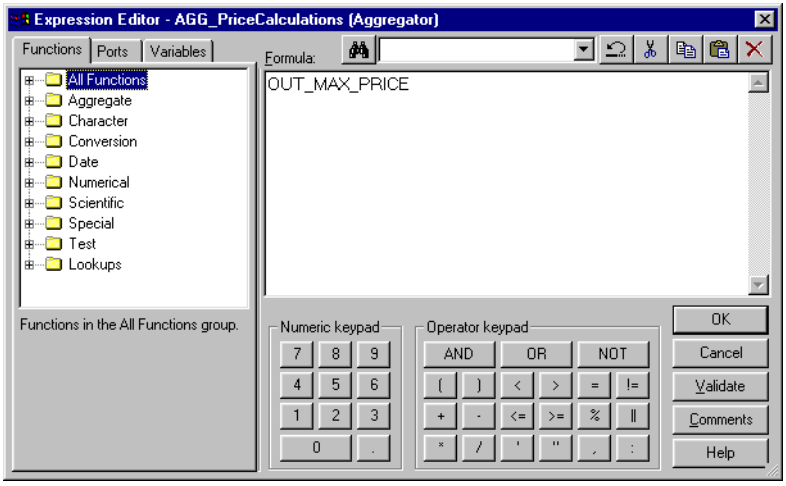
The next step is to enter the expressions for all three output ports, using the functions MAX, MIN, and AVG to perform aggregate calculations.

To enter the first aggregate calculation:

1. Click the Expression field of the OUT_MAX_PRICE column.

   A small button appears in the right corner of the field.

   Click the button to open the Expression Editor.



   The Formula section of the Expression Editor displays the expression as you develop it. You use other sections of this dialog box to select the input ports to provide values for an expression, enter literals and operators, and select functions to use in the expression. If you declared mapping parameters or variables in the mapping, you can select them from the variables tab and use them in the expression.

2. Select the text OUT_MAX_PRICE and press the Delete key.

3. Double-click the Aggregate heading in the Functions section of the dialog box.

   A list of all group functions now appears.

4. Double-click the Max function on the list.

   The MAX function now appears in the window where you enter the expression. To perform the calculation, you need to add a reference to an input port that provides data for the expression.

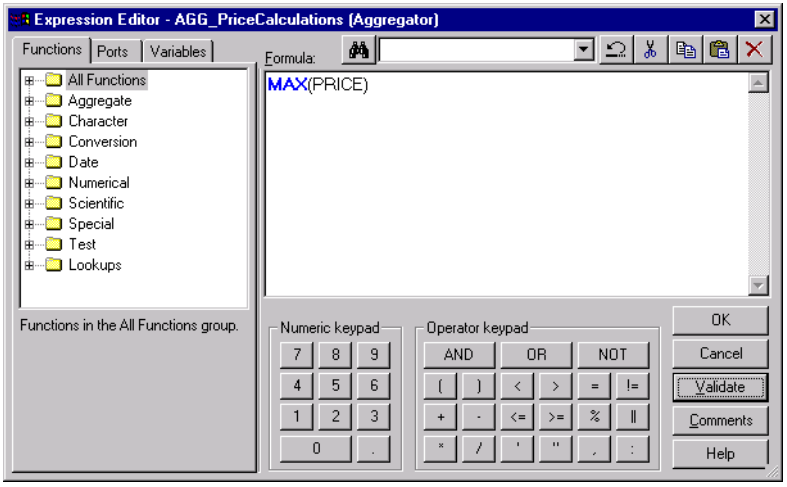
5. Move the cursor between the parentheses next to MAX.

6. Click the Ports tab.

   This section of the Expression Editor displays all the ports from all transformations appearing in the mapping.

7. Double-click the PRICE port appearing beneath AGG_PriceCalculations.

   A reference to this port now appears within the expression. The final step is to validate the expression.



8. Click Validate.

   If you followed the steps in this portion of the lesson, the Designer informs you that the expression parsed successfully. The syntax you entered has no errors.

9. Click OK to close the message box from the parser, then click OK again to close the Expression Editor.

To enter the remaining aggregate calculations:

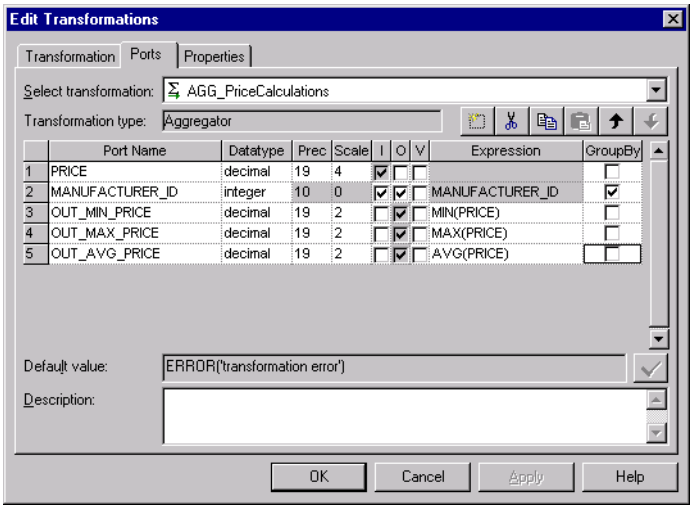
1. Using the technique you just learned, enter and validate the following expressions for the other two output ports:

Port	Expression
OUT_MIN_PRICE	MIN(PRICE)
OUT_AVG_PRICE	AVG(PRICE)

   Both MIN and AVG appear in the list of Aggregate functions, along with MAX.

   Note that you can type this syntax, instead of using the point and click features of the Expression Editor. However, use the point and click method to avoid typing mistakes.

2. Click OK to close the Edit Transformations dialog box.



3. Choose Repository-Save.

   You may notice an error message in the Output window at this point. Ignore it, since it just indicates that you have not connected targets to transformations yet.

Creating an Expression Transformation

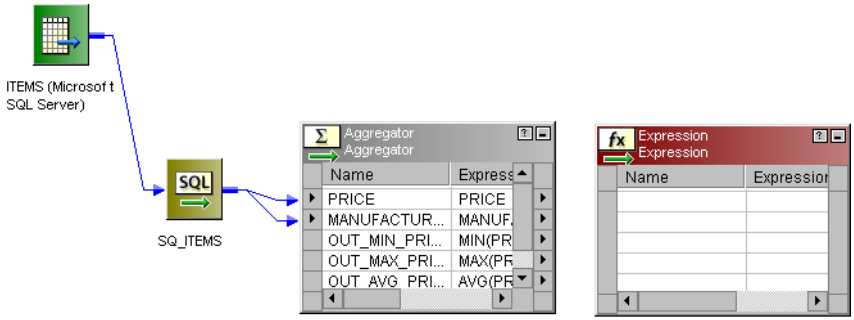
Now that you have calculated the highest, lowest, and average prices for items, the next step is to calculate the average profitability of items from each manufacturer. While such calculations are normally more complex, you simply multiply the average price by 0.2 (20%).

To add this information to the target, you need to create an Expression transformation that takes the average price of items from a manufacturer, performs the calculation, then passes the result along to the target. As you develop your own transformations, you connect transformations, using the output of one transformation as an input for others.

To add an Expression transformation:

1. Click the Expression button on the Transformations toolbar.
2. Click in the area to the right of the Aggregator transformation.

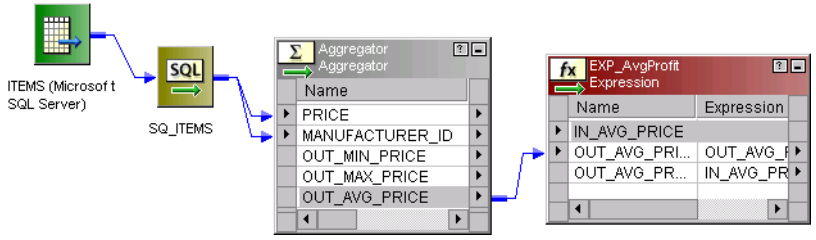
The new transformation appears when you release the mouse button.



3. Double-click the title bar of the Expression transformation.
4. In the Edit Transformations dialog box, change the name of this transformation to EXP_AvgProfit.
5. Add a new input port, IN_AVG_PRICE, and a new output port, OUT_AVG_PROFIT, both using the Decimal datatype with precision of 19 and scale of 2.
6. Enter the following expression for OUT_AVG_PROFIT:

   ```
   IN_AVG_PRICE * 0.2
   ```

7. Validate the expression.
8. Close the Expression Editor, then close the Edit Transformations dialog box.
9. Connect OUT_AVG_PRICE from the Aggregator to the new input port.



10. Choose Repository-Save.

Locating Additional Data

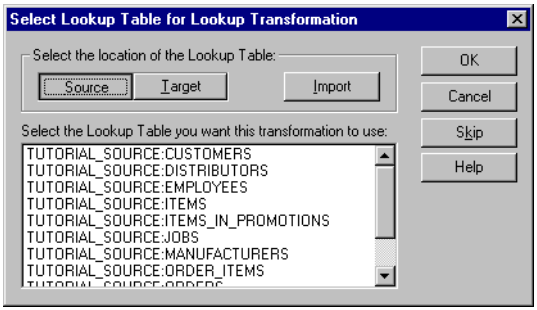
The last step is to find each manufacturer name in the MANUFACTURERS table, using the Lookup transformation.

To add the Lookup transformation:

1. Click the Lookup button on the Transformations toolbar.
2. Click the mapping to create the new transformation.

A dialog box prompts you to identify the source or target to provide data for the lookup. When you run a session, the Informatica Server must access the lookup source.

3. Click Source.



4. Select the MANUFACTURERS table from the scrolling list and click OK.

   The Designer now adds the transformation. Notice that the Lookup transformation has the same columns as in the MANUFACTURERS source definition you created in the Source Analyzer.

   Since you already created a source definition for the MANUFACTURER table, you do not have to import the source definition into the repository at this point. You can use source and target definitions in the repository to identify a lookup source for the Lookup transformation. Alternatively, using the Import feature, you can open a connection to a database, and designate some table or view in that database as the lookup source.

5. Double-click the title bar of the transformation.

6. Name this transformation LKP_Manufacturers.

7. Add a new input port, IN_MANUFACTURER_ID, using the same datatype as MANUFACTURER_ID.

   This port receives MANUFACTURER_ID values. When the transformation receives a new value through this port, it looks up the matching value in a record from MANUFACTURERS. The Lookup transformation queries and stores the contents of the table before the rest of the transformation runs, so it is performing the join through a local copy of the table that it has cached.

8. Click the Condition tab, then click the Add button.

   An entry for the first condition in the lookup appears. Each row in this dialog box represents one condition in the WHERE clause that the Informatica Server generates when querying records.

9. Verify the following settings for the condition:

Lookup Table Column	Operator	Transformation Port
MANUFACTURER_ID	=	IN_MANUFACTURER_ID

Note: If the datatypes (including precision and scale) of these two columns do not match, you cannot validate the mapping.

10. Click the Properties tab.

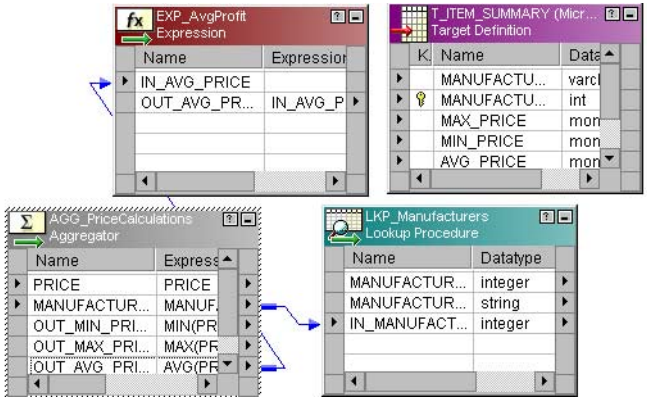
    Do not change any settings in this section of the dialog box. You can modify how the Lookup transformation queries and caches values.

11. Click OK.

    You now have a Lookup transformation that reads values from the MANUFACTURERS table and can perform lookups using values passed through the IN_MANUFACTURER_ID input port. The final step is to connect this Lookup transformation to the rest of the mapping.

12. Choose Layout-Link Columns.

13. Connect the output port MANUFACTURER_ID from the Aggregator transformation to the input port IN_MANUFACTURER_ID in the Lookup transformation.



14. Choose Repository-Save.

Connecting the Target

You have set up all the transformations needed to modify data before writing to the target. So far, you have performed the following tasks:

- Created a target.
- Created a mapping.
- Added transformations (and calculations).

The final step is to connect to the target.

To connect to the target:

1. Click and drag the following output ports to the corresponding input ports in the target:

Transformation	Output Port	Input Port
Lookup	MANUFACTURER_ID	MANUFACTURER_ID
Lookup	MANUFACTURER_NAME	MANUFACTURER_NAME
Aggregator	OUT_MIN_PRICE	MIN_PRICE
Aggregator	OUT_MAX_PRICE	MAX_PRICE
Aggregator	OUT_AVG_PRICE	AVG_PRICE
Expression	OUT_AVG_PROFIT	AVG_PROFIT

2. Choose Repository-Save.
3. Close the Designer.

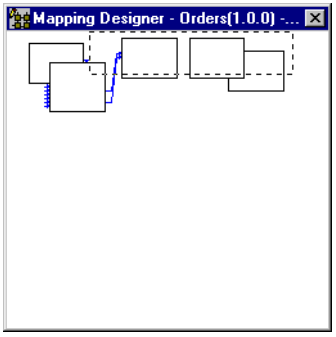
Using the Overview Window

When you create a mapping with multiple transformations, you may not be able to see the entire mapping at once due to the size of the mapping. Use the Overview window to navigate around the workbook containing this mapping.

To use the Overview window:

1. Choose View-Overview Window.

   A window appears, displaying a smaller version of the mapping.



2. Click and drag the dotted square (the viewing rectangle) within this window.

   As you move the viewing rectangle, your perspective on the mapping changes.

What Comes Next

You have two mappings:

- **m_PhoneList.** A pass-through mapping that reads employee names and phone numbers.
- **m_ItemSummary.** A more complex mapping that performs simple and aggregate calculations as well as lookups.

You can now create a session for each mapping in the Server Manager.

Lesson 5: Creating Sessions and Batches

Sessions are a set of instructions that tell the Informatica Server how and when to move data from sources to targets. You create a session for each mapping that you want the Informatica Server to run. The Informatica Server uses the instructions configured in the session to move data from sources to targets. You create and maintain sessions in the Server Manager.

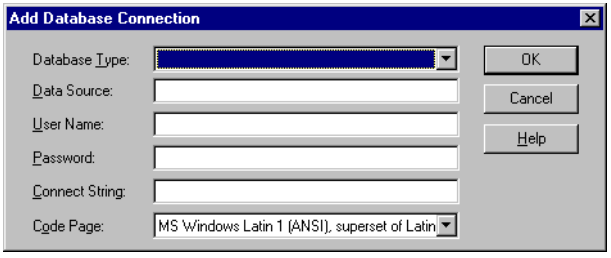
Configuring the Informatica Server and the Server Manager

Before you can create a session, you need to configure the Server Manager to communicate with the Informatica Server.

First, you need to provide the Informatica Server with the information it needs to connect to the source and target databases.

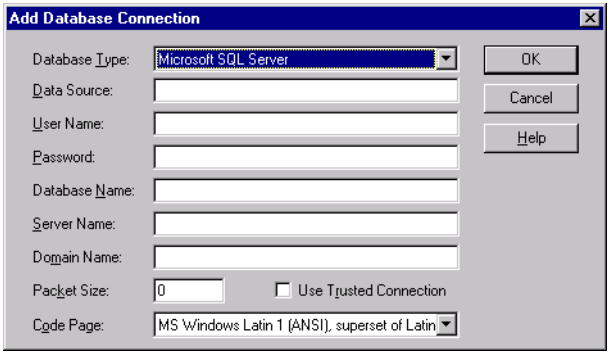
To define database connections:

1. Launch the Server Manager.
2. Select the TUTORIAL repository in the Navigator, then choose Repository-Connect.
3. Enter the username and password (jdoe/jdoe) for connecting to it, and click Connect.
4. Choose Server Configuration-Database Connections.
5. In the Database Connections dialog box, click Add.



6. Select your Database Type from the list.

The dialog box now expands, adding options appropriate to the selected database platform.



7. In the Data Source field, enter TUTORIAL_SOURCE as the name of the database connection.

   The Informatica Server uses this database name as a reference to one database connection.

8. Enter the username and password, as well as the name of the ODBC data source to connect to the database.

9. Select a code page for the database connection.

   The source code page must be a subset of the Informatica Server code page and the target code page.

10. Enter any additional information to connect to this database and click OK.

    TUTORIAL_SOURCE now appears in the list of registered database connections.

11. Repeat the above steps to create another database connection called TUTORIAL_TARGET for the target databases.

    The target code page must be a superset of the Informatica Server code page and the source code page.

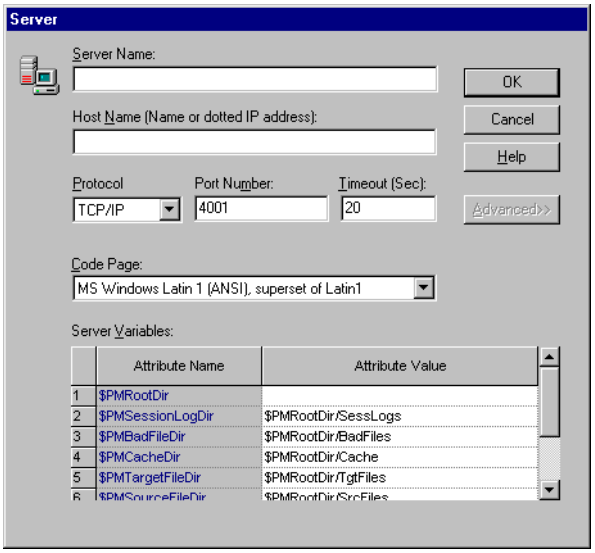
When you finish, TUTORIAL_SOURCE and TUTORIAL_TARGET appear in the list of registered database connections.

12. Click Close.

You can now configure the Server Manager to communicate with the Informatica Server.

To configure the Server Manager to communicate with the Informatica Server:

1. Make sure the Informatica Server is running.
2. While connected to the TUTORIAL repository in the Server Manager, choose Server Configuration-Register Server.



3. In the Informatica Server dialog box, enter the Server Name and the Host Name (or address) where your Informatica Server is located.
4. Select the network protocol (IPX/SPX or TCP/IP) to access the Informatica Server across your network and click Select. If you are using TCP/IP, you also need to enter the port number the Informatica Server is using.

   Ignore the timeout setting for now.
5. Select a code page for the Informatica Server.

   The code page must be identical to the code page used by the operating system on which the Informatica Server is running.
6. Enter a directory for the Informatica Server variable attribute $PMRootDir.

   $PMRootDir is the root directory used by all other Informatica Server variables. Server variables are default directories you want the Informatica Server to use. It is recommended you enter the directory where the Informatica Server is installed, since all of the necessary subdirectories are created there when you install the Informatica Server.

If you enter a different directory, you must make sure a directory exists for each Informatica Server variable.

7. Click OK.

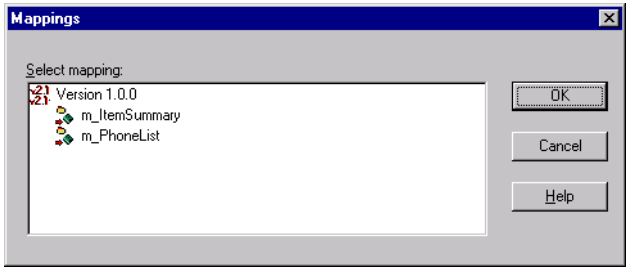
You have now finished configuring the Server Manager to communicate with the Informatica Server, and indicated how to connect to the source, target, and repository databases. The next step is to create a session for the mapping m_PhoneList.

Creating the First Session

The first session you create uses the simple mapping m_PhoneList. Later, you add a second session to run the second mapping, m_ItemSummary.

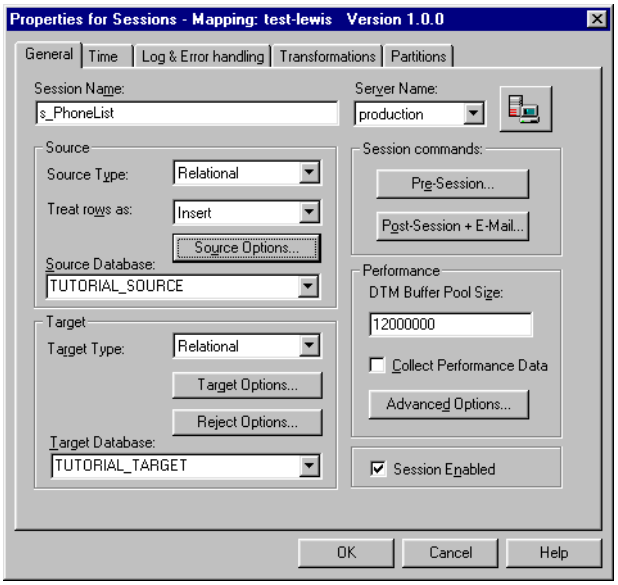
To create the session:

1. In the Navigator, double-click the folder containing the m_PhoneList mapping.
2. Click the Add Session button, or choose Operations-Add Sessions.
3. Select the mapping m_PhoneList and click OK.



The Session Wizard now appears.

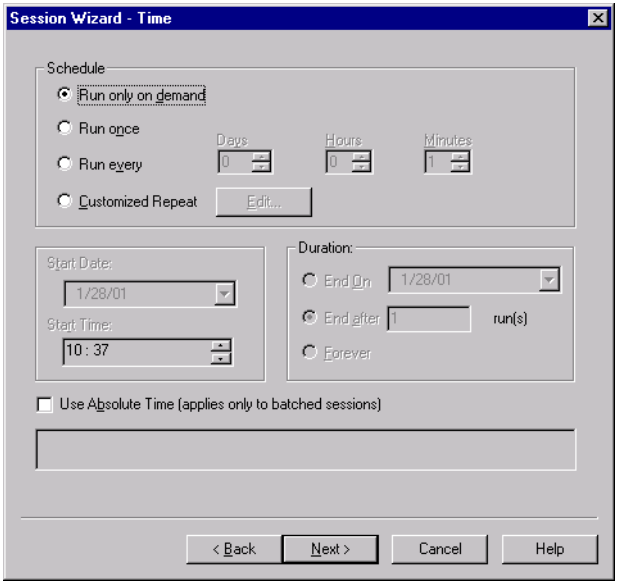
4. Enter s_PhoneList as the session name.



5. In the Source section, select Relational and TUTORIAL_SOURCE for your Source Database.

6. In the Target section, select Relational and TUTORIAL_TARGET for your Target Database. These are the only options you need to select here.

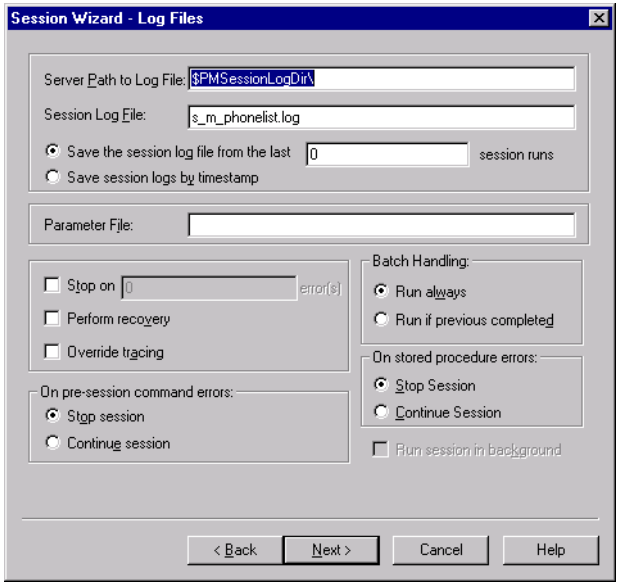
7. Click Next.



8. For the schedule, select Run Only on Demand.

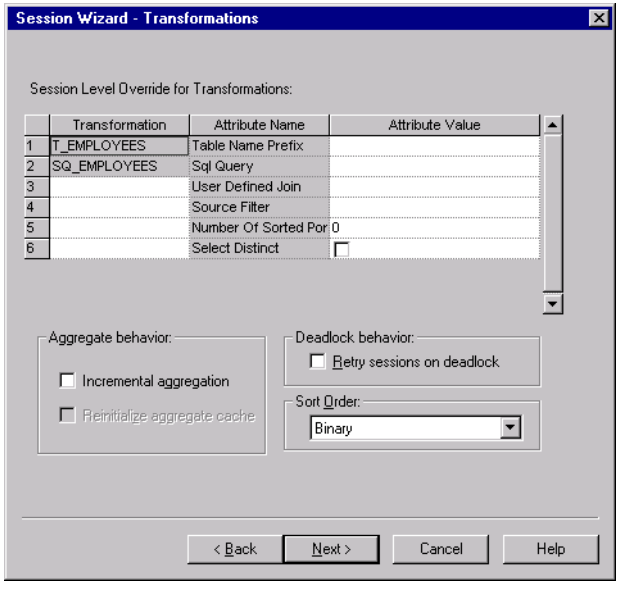
You can set various intervals (for example, every week or every six hours) for running the session. If you want to view additional options, select Customized Repeat and click Edit.

9. Click Next.



You can enter the name and directory for session log files and set error handling options on the log file page. Accept the default settings.

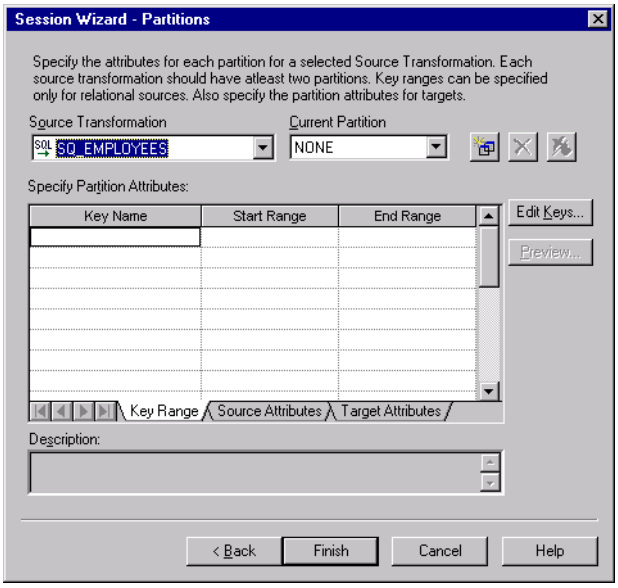
10. Click Next.



This page allows you to override transformation attributes for the session. Do not override transformation attributes for this session.

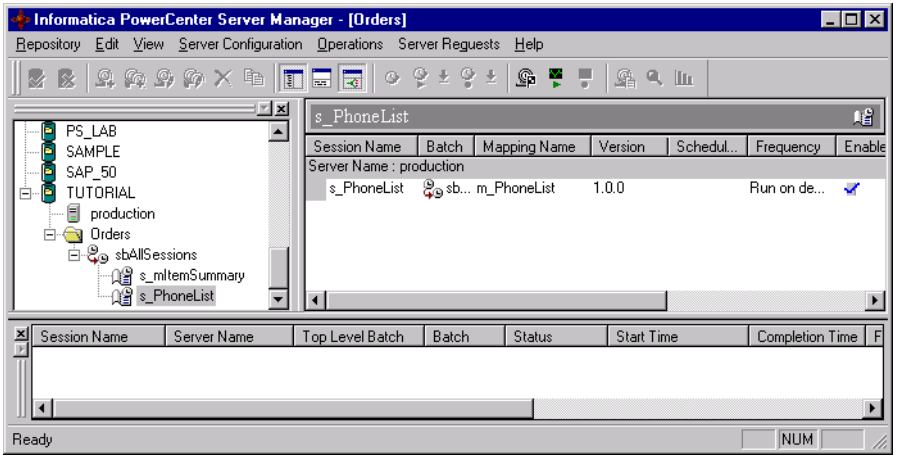
Select one of the sort orders associated with the Informatica Server code page. For English data, use the Binary sort order. For more information about sort orders and code pages, see "Globalization Overview" in the *Installation and Configuration Guide*.

11. Click Next.



This page allows you to add and configure partitions based on source qualifiers. For details on configuring partitions, see "Partitioning Data" in the *Session and Server Guide*.

12. Click Finish.



The Server Manager now prompts the Repository to store the description of the new session. You can now monitor and run the session.

Monitoring Sessions

As you work with sessions in the Server Manager, use the following menu options to poll and view Informatica Server activity:

- **Server Configuration-Monitor.** Use this option to toggle the monitor window.
- **Server Requests.** Use this option to Start Polling, Stop Polling, or to select Session Overview. When polling is enabled, the Server Manager automatically polls the Informatica Server and displays the current status of any jobs in the Monitor window. The Session Overview displays a quick snapshot of the latest activity.

At several points during the tutorial, you can turn the monitor on and off.

To monitor the Informatica Server:

1. Select the Informatica Server in the Navigator. Choose Server Configuration-Monitor.

   This opens the Monitor window (and splits the screen) to show a snapshot of all sessions running on the selected Informatica Server. The session is listed in the upper section of the screen. If other sessions were scheduled, the lower window on the screen would list them.

2. To see a refreshed view of the session status, choose Server Requests-Session Overview.

   If the session completes before you refresh the view, the Server Manager indicates when the Informatica Server finishes running the session.

Running the Session

After you create a session, you can run it to move the data from the source to the target.

To run the session:

1. In the Server Manager, click the session to select it.
2. Click the Start A Session/Batch button on the toolbar, or choose Server Requests-Start.

   The session returns the following results:

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	OFFICE_PHONE
1921	Nelson	William	415-541-5145
1922	Page	Ian	415-541-5145
1923	Osborne	Lyle	415-541-5145
1928	DeSouza	Leo	415-541-5145
2001	S. MacDonald	Ira	415-541-5145
2002	Hill	Andy	415-541-5145

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	OFFICE_PHONE
2003	Sawyer	Monisha	415-541-5145
2006	St. Jean	Bender	415-541-5145
2100	Johnson	Teddy	415-541-5145
2102	Steadman	Ono	415-541-5145
2103	Markowitz	John	415-541-5145
2109	Centre	Tom	415-541-5145

(12 rows affected)

Creating the Second Session

You can now create a session for the mapping, m_ItemSummary.

To create the second session:

1. In the Server Manager, switch back to configure mode by selecting a folder in the Navigator.
2. Open the Orders folder.
3. Click the Add Session button, or choose Operations-Add Sessions.
4. Select the mapping m_ItemSummary and click OK.
5. Complete the Session Wizard and enter the following settings:

Session Property	Setting
Name	s_ItemSummary
Source Database	TUTORIAL_SOURCE
Target Database	TUTORIAL_TARGET
Interval	Run Only On Demand

6. Click Finish to close the Session Wizard.

Now that you have two sessions, you can organize them in the same batch. When you run the batch, the Informatica Server runs all sessions within the batch, either simultaneously or in sequence.

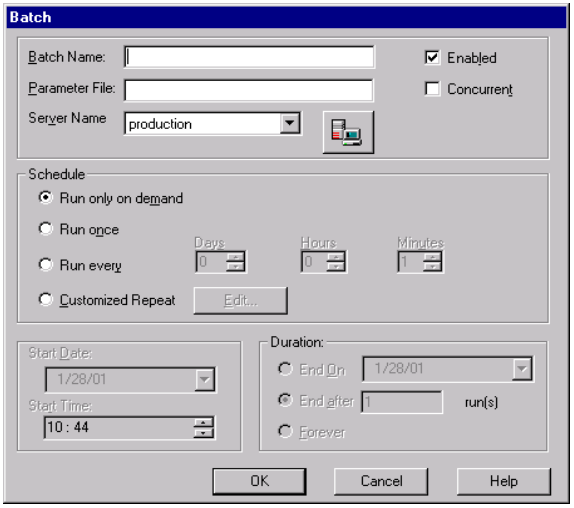
Creating a Batch

Batches provide a way to group sessions for either serial or parallel execution by the Informatica Server. When you place sessions in a batch, the batch schedule overrides the session schedule by default.

To create the batch:

1. In the Server Manager, click the Add Batch button, or choose Operations-Add Batch.

   The Batch dialog box appears, where you can select batch settings.



2. Enter bs_AllSessions as the batch name.

   bs_*BatchName* is the naming convention for sequential batches. For a concurrent batch, the convention is bc_*BatchName*.

3. Schedule the batch to run on demand.

4. Click OK.

   The new batch appears in the same list as the two sessions as you created. The next step is to move these sessions into the batch.

Grouping Sessions into Batches

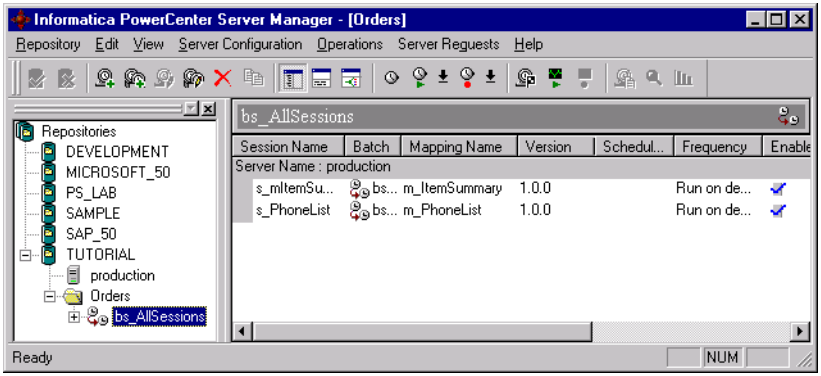
You can group sessions in a batch to improve performance or to ensure that targets load in a set order.

To group sessions into the batch:

1. Click and drag the s_PhoneList session into the batch.

2. Click and drag s_ItemSummary into the batch.

The Server Manager notifies you that the sessions are now part of the batch. In the list of sessions and batches, the sessions now appear within the batch.



Note: Since both sessions move a small amount of data, you do not see any performance loss if you run the two sessions simultaneously. By default, the sessions run consecutively, in order from top to bottom as they appear in the list of sessions within the batch.

Adding Sessions to Batches

If you want to change the order in which the sessions run, you can click and drag the sessions to change the order in which they appear in the batch list. You can configure the batch to run both sessions simultaneously.

To configure both sessions to run concurrently:

1. Double-click the name of the batch.

   The Batch dialog box reappears.

2. Check the Concurrent option.

3. Change the name of the batch to bc_AllSessions.

4. Click OK.

   The Server Manager notifies you that the Informatica Server has saved this change to the batch in the repository.

Running the Batch

After you create the batch and add sessions to it, you can run the batch.

To run the batch:

1. Click the batch to select it, then click the Start a Session/Batch button, or choose Server Requests-Start.

2. Choose Start Polling from the Server Requests menu to view the progress of these sessions.

The results from running the s_ItemSummary session are as follows:

MANUFACTURER_ID	MANUFACTURER_NAME	MAX_PRICE	MIN_PRICE	AVG_PRICE	AVG_PROFIT
100	Nike	365.00	169.95	261.24	52.25
101	OBrien	188.00	44.95	134.32	26.86
102	Mistral	390.00	70.00	200.00	40.00
103	Spinnaker	70.00	29.00	52.98	10.60
104	Head	179.00	52.00	98.67	19.73
105	Jesper	325.00	34.95	133.65	26.73
106	Acme	195.00	56.95	143.65	28.73
107	Medallion	235.00	19.95	98.65	19.73
108	Sportstar	280.00	18.00	149.00	29.80
109	WindJammer	430.00	395.00	412.50	82.50
110	Monsoon	280.00	280.00	280.00	56.00

(11 rows affected)

What Comes Next

In Lesson 6, you can create a mapping with Filter, Stored Procedure, and Sequence Generator transformations.

Chapter 3

Tutorial Lesson 6

This lesson in the tutorial walks you through the process of creating complex mappings.

Lesson 6: Creating a Complex Mapping

In previous lessons, you use the Source Qualifier, Expression, Aggregator, and Lookup transformations in mappings. In this lesson, you learn how to use the Stored Procedure, Filter, and Sequence Generator transformations.

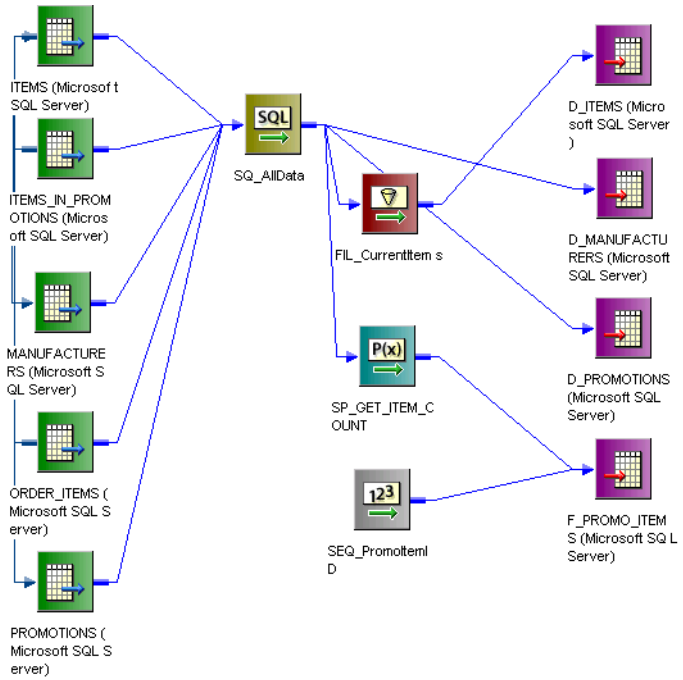
You use a Filter transformation to filter extraneous data, such as discontinued items in the ITEMS table.

To generate unique IDs for target tables, you use the Sequence Generator transformation.

Stored Procedure transformations call stored procedures and capture their return values.

Figure 3-1 displays the complex mapping you create in this lesson using Stored Procedure, Filter, and Sequence Generator transformations:

Figure 3-1. Complex Mapping



Creating Targets

Before creating the mapping, create the following target tables:

- F_PROMO_ITEMS, a fact table of promotional items.
- D_ITEMS, D_PROMOTIONS, and D_MANUFACTURERS, the dimensional tables.

Note: Fact and dimension tables are described in "Creating Cubes and Dimensions" in the *Designer Guide*.

To design the new targets:

1. In the Designer, choose Window-New Window.
2. In the new workbook, switch to the Warehouse Designer. Note that you are still working in the Orders folder at this point. To clear your workspace, select all objects in the workspace, then choose Edit-Clear.
3. Choose Targets-Create.
4. In the Create Target Table dialog box, enter F_PROMO_ITEMS as the name of the new target table, select the database type, and click Create.
5. Repeat step 4 to create the other tables needed for this schema: D_ITEMS, D_PROMOTIONS, and D_MANUFACTURERS. When you have created all these tables, click Done.
6. Double-click the title bar of each new target, and add these columns to the appropriate table:

   F_PROMO_ITEMS

Column	Datatype	Precision	Not Null	Key
PROMO_ITEM_ID	Integer	NA	Not Null	Primary Key
FK_ITEM_ID	Integer	NA		Foreign Key
FK_PROMOTION_ID	Integer	NA		Foreign Key
FK_MANUFACTURER_ID	Integer	NA		Foreign Key
NUMBER_ORDERED	Integer	NA		
DISCOUNT	Money	default		
COMMENTS	Varchar	default		

Note: For F_PROMO_ITEMS, you are including foreign key columns that correspond to the primary keys in each of the following tables.

D_ITEMS

Column	Datatype	Precision	Not Null	Key
ITEM_ID	Integer	NA	Not Null	Primary Key
ITEM_NAME	Varchar	72		
PRICE	Money	default		

D_PROMOTIONS

Column	Datatype	Precision	Not Null	Key
PROMOTION_ID	Integer	NA	Not Null	Primary Key
PROMOTION_NAME	Varchar	72		
DESCRIPTION	Varchar	default		
START_DATE	Datetime	default		
END_DATE	Datetime	default		

D_MANUFACTURERS

Column	Datatype	Precision	Not Null	Key
MANUFACTURER_ID	Integer	NA	Not Null	Primary Key
MANUFACTURER_NAME	Varchar	72		

The next step is to generate and execute the SQL script to create each of these new target tables.

To create the tables:

1. Select all the table definitions.
2. Choose Targets-Generate/Execute SQL.
3. In the Database Object Generation dialog box, connect to the target database.
4. Choose Generate from Selected Tables, and select the options for creating the tables and generating keys.
5. Click Generate and Execute.

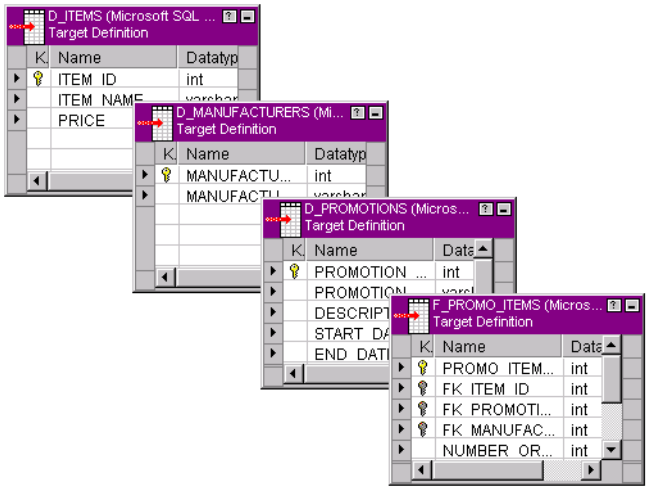
   The Designer now runs the SQL script needed to create the table.
6. Click Close, then choose Repository-Save.

Creating the New Mapping

Create a new mapping for the new targets. The mapping filters out discontinued items, calls a stored procedure to find how many of each item customers have ordered, and generates a unique ID for each row in the fact table.

To create the new mapping:

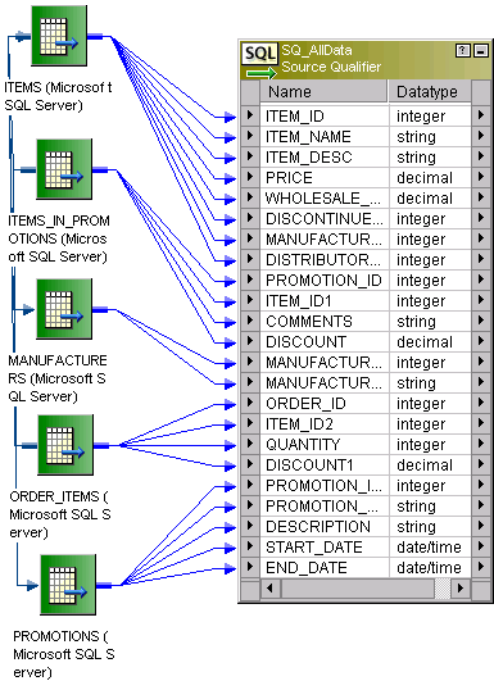
1. In the Designer, switch to the Mapping Designer and create a new mapping.
2. Name the mapping m_PromoItems.
3. From the list of targets, select the tables you just created and drag them into the mapping.



4. From the list of sources, add the following sources to the mapping:
   - PROMOTIONS
   - ITEMS_IN_PROMOTIONS
   - ITEMS
   - MANUFACTURERS
   - ORDER_ITEMS
5. Delete all Source Qualifiers that the Designer creates when you add these source definitions.

   The Designer creates a Source Qualifier for each source by default.

6. Add a Source Qualifier transformation, SQ_AllData, to the mapping, connecting all the sources to it.



When you create a single Source Qualifier, the Informatica Server increases performance with a single read on the source database instead of multiple reads.

7. Choose View-Navigator to close the Navigator window to allow extra space in the workbook.

8. Choose Repository-Save.

Creating a Filter Transformation

The Filter transformation filters records from a source. If you connect a Filter to a Source Qualifier transformation, you can filter records passed through the Source Qualifier using any condition you want to apply. In this exercise, you remove all discontinued items from the mapping.

The mapping contains a Filter transformation that limits records queried from the ITEMS table to only those items that have not been discontinued.

To add the Filter transformation:

1. Click the Filter button on the Transformations toolbar.

2. Click and drag between the Source Qualifier and the target tables to create the Filter transformation.

3. Click and drag the following columns from the Source Qualifier transformation into the Filter transformation:

   - ITEM_ID
   - ITEM_NAME
   - PRICE
   - DISCONTINUED_FLAG

4. Double-click the Filter transformation title bar.

5. Rename the transformation FIL_CurrentItems.

6. Click the Properties tab to specify the filter condition.

7. Select the Filter Condition field, and click the arrow button that appears in the right corner of the field.

   The Expression Editor dialog box appears.

8. Select the word TRUE in the Formula field and press Delete.

9. Click the Ports tab, then double-click the DISCONTINUED_FLAG port.

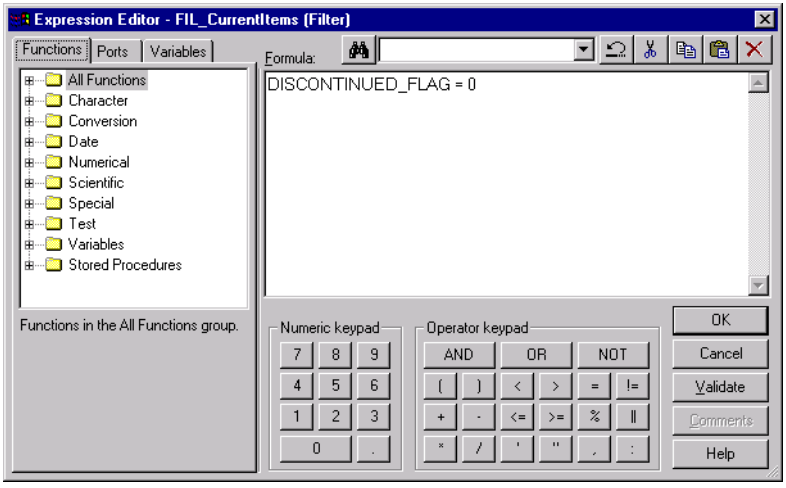
10. Click = and 0 (zero).

    The complete condition should now read:

    ```
    DISCONTINUED_FLAG=0
    ```

11. Click Validate, then click OK.

    The new condition now appears in the Value field.

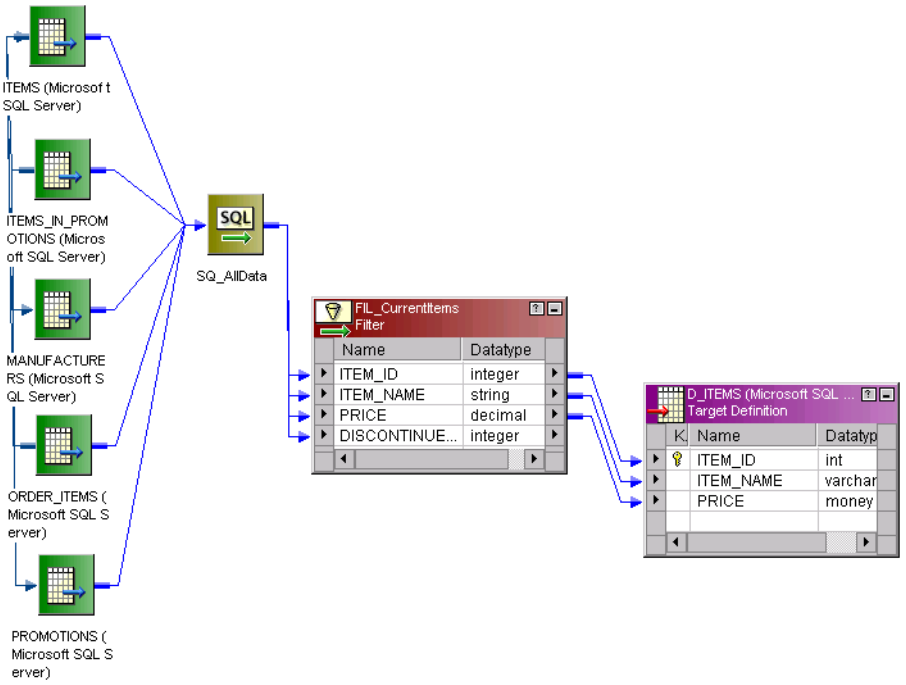


12. Click OK to return to the workbook.

The last step is to connect the Filter transformation to the D_ITEMS target table. Only currently sold items are written to this target.

To connect the Filter transformation:

1. Connect the columns ITEM_ID, ITEM_NAME, and PRICE to their counterpart columns in D_ITEMS.



2. Choose Repository-Save.

Creating a Sequence Generator Transformation

Many relational databases include sequences, a special database object designed exclusively to generate values. The Sequence Generator transformation functions exactly like a sequence, but instead of writing the SQL code needed to create and use the sequence, you add one more transformation to your mapping to generate unique values for a target. You can use this transformation to generate unique values, like primary key values. You can also use it to cycle through a closed set of values.

If you have used sequences before, the properties of the Sequence Generator transformation should look familiar:

- The starting number (normally 1).
- The current value stored in the repository.

- The number that the Sequence Generator transformation adds to its current value for every request for a new ID.
- The maximum value in the sequence.
- A flag indicating whether the Sequence Generator transformation counter resets to the minimum value once it has reached its maximum value.

The Sequence Generator transformation has two output ports, NEXTVAL and CURRVAL, which correspond to the two pseudo-columns in a sequence. When you query a value from the NEXTVAL port, the transformation generates a new value.

In our new mapping, you add a Sequence Generator transformation to generate IDs for the fact table, F_PROMO_ITEMS. Every time the database inserts a new record into this table, the transformation provides a unique ID for PROMO_ITEM_ID.

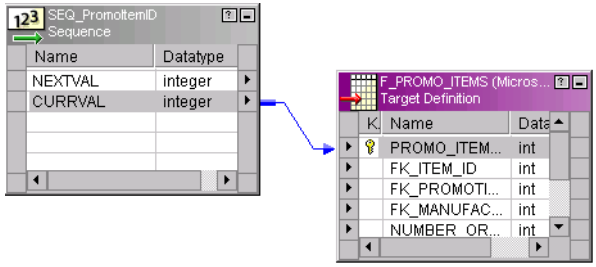
To add the Sequence Generator transformation:

1. On the Transformations toolbar, click the Sequence Generator button.
2. Click and drag to create a new Sequence Generator transformation.
3. Double-click the Sequence Generator transformation title bar.
4. Rename the transformation SEQ_PromoItemID.
5. Click the Ports tab.

   The two output ports, NEXTVAL and CURRVAL, appear in the list.

   Note: You cannot add any new ports to this transformation, or reconfigure NEXTVAL and CURRVAL.
6. Click the Properties tab.

   The properties for the Sequence Generator transformation appear. You do not have to change any of these settings.
7. Click OK.
8. Connect the NEXTVAL column from the Sequence Generator transformation to the PROMO_ITEM_ID column in the target table F_PROMO_ITEMS.



9. Choose Repository-Save.

Creating a Stored Procedure Transformation

When you install the sample database objects, you also create a stored procedure, SP_GET_ITEM_COUNT. This procedure takes one argument, an ITEM_ID value, and returns the number of times that item has been ordered.

Table 3-1 describes the syntax for the stored procedure:

Table 3-1. Stored Procedure Syntax

Database	Syntax

Oracle	`CREATE FUNCTION SP_GET_ITEM_COUNT`
	`(ARG_ITEM_ID IN NUMBER) RETURN NUMBER`
	`IS SP_RESULT NUMBER;`
	`BEGIN`
	`SELECT COUNT(*)`
	`INTO SP_RESULT`
	`FROM ORDER_ITEMS`
	`WHERE ITEM_ID = ARG_ITEM_ID;`
	`RETURN (SP_RESULT);`
	`END;`

Microsoft SQL Server	`CREATE PROCEDURE SP_GET_ITEM_COUNT (@ITEM_ID INT)`
	`AS SELECT COUNT(*) FROM ORDER_ITEMS`
	`WHERE ITEM_ID = @ITEM_ID`

Sybase	`CREATE PROCEDURE SP_GET_ITEM_COUNT (@ITEM_ID INT)` `AS SELECT COUNT(*) FROM ORDER_ITEMS WHERE ITEM_ID = @ITEM_ID`

Informix	`CREATE PROCEDURE SP_GET_ITEM_COUNT (ITEM_ID_INPUT INT)`
	`RETURNING INT;`
	`DEFINE CNT INT;`
	`SELECT COUNT(*) INTO CNT FROM ORDER_ITEMS WHERE ITEM_ID =`
	`ITEM_ID_INPUT;`
	`RETURN CNT;`

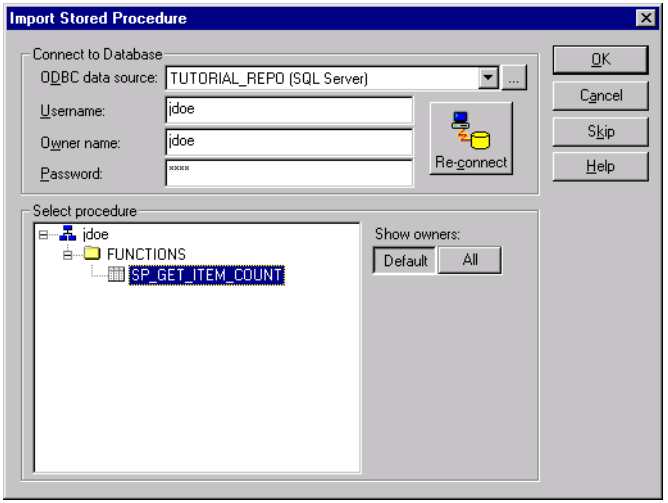
In the mapping, add a Stored Procedure transformation to call this procedure. The Stored Procedure transformation returns the number of orders containing an item to an output port.

To create the Stored Procedure transformation:

1. Click the Stored Procedure button on the Transformations toolbar.
2. Click in the workbook.

   The Import Stored Procedure dialog box appears.

3. Select the ODBC connection for your source database. Enter your username, owner name, and password. Click Connect.



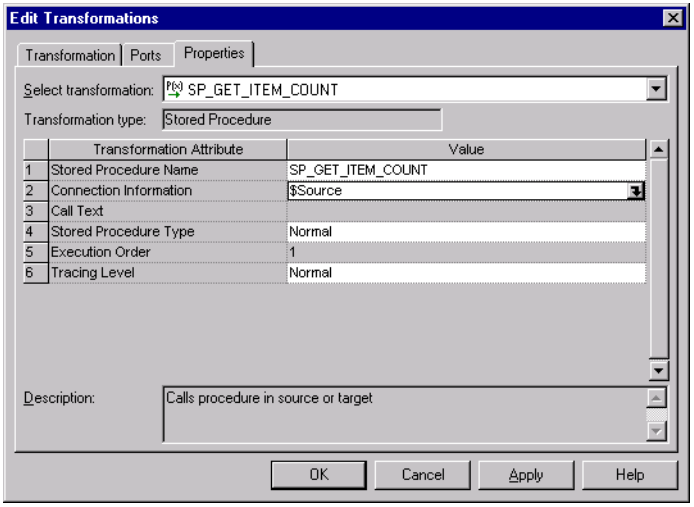
4. Select the stored procedure named SP_GET_ITEM_COUNT from the tree and click OK.

   The Stored Procedure transformation appears in the mapping.

5. Double-click the title bar of the transformation, and select the Properties tab.

6. In the Connection information section, select $Source.

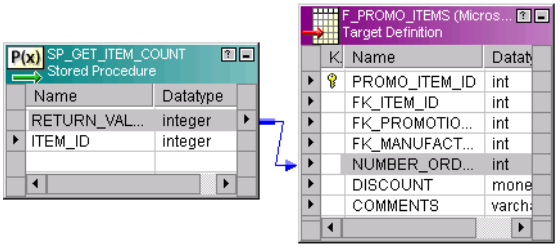
   You can call stored procedures in both source and target databases.



7. Click OK.

8. Connect the ITEM_ID column from the Source Qualifier to the ITEM_ID column in the Stored Procedure transformation.

9. Connect the RETURN_VALUE column from the Stored Procedure transformation to the NUMBER_ORDERED column in the target table F_PROMO_ITEMS.



10. Choose Repository-Save.

Completing the Mapping

The final step is to map data to the remaining columns in targets.

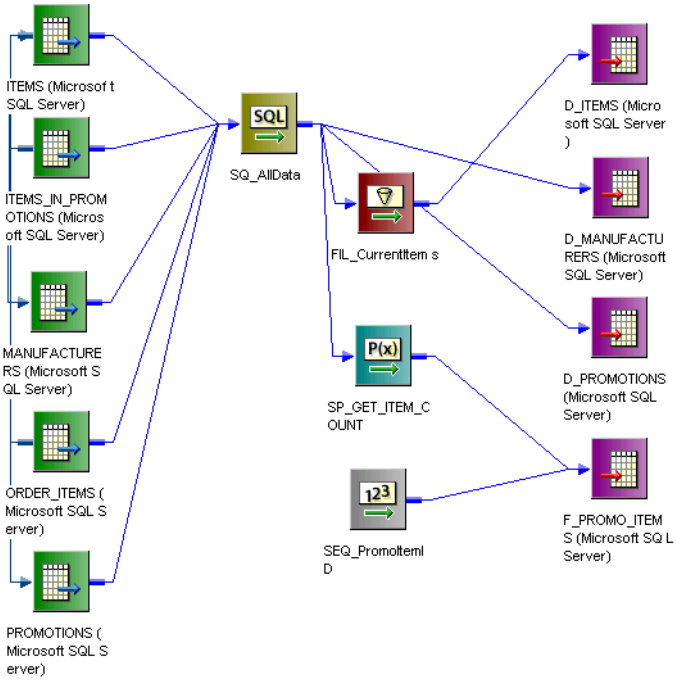
To complete the mapping:

1. Connect the following columns from the Source Qualifier transformation to the targets:

Source Qualifier	Target Table	Column
PROMOTION_ID	D_PROMOTIONS	PROMOTION_ID
PROMOTION_NAME	D_PROMOTIONS	PROMOTION_NAME
DESCRIPTION	D_PROMOTIONS	DESCRIPTION
START_DATE	D_PROMOTIONS	START_DATE
END_DATE	D_PROMOTIONS	END_DATE
MANUFACTURER_ID	D_MANUFACTURERS	MANUFACTURER_ID
MANUFACTURER_NAME	D_MANUFACTURERS	MANUFACTURER_NAME

2. Choose Repository-Save.

The mapping is now complete. You can create and run a new session with this mapping.



Chapter 4

Tutorial Lesson 7

This lesson teaches you how to create mappings with XML sources and targets and the Router transformation.

Lesson 7: Creating a Mapping with XML Sources and Targets

In this lesson, you learn how to create XML sources and targets and how to use the Router transformation.

XML is a common means of exchanging data on the web. You can use XML files as a source of data and as a target for transformed data.

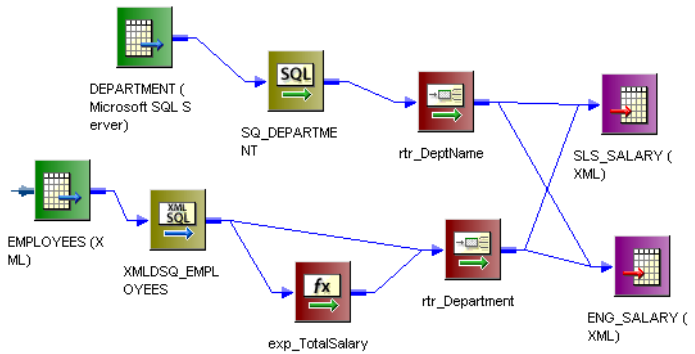
You use the Router transformation to test data for several different conditions at once and to send the results of each test to a separate data flow. In this case, you test for the department ID and send data for each department to a separate target.

In this lesson, you use an XML file that contains employee and salary data to create an XML source definition. Employees can have three types of wages, which appear on the XML file as three occurrences of salary. You pivot the occurrences of employee salaries into three columns (base salary, commission, and bonus) and calculate the total salary in an Expression transformation.

You use a Router transformation to separate the data by department. You send the data for the Engineering and Sales department into two instances of an XML target. Additionally, you obtain the name of the department from a relational source definition. You use another Router transformation to select the correct department names to pass to the targets.

Figure 4-1 shows the mapping that you create in this lesson:

Figure 4-1. Mapping with XML Sources and Targets



Creating the XML Source

To create the XML source definition, use the Employees.xml file and its associated DTD, Employees.dtd.

To create the XML source definition:

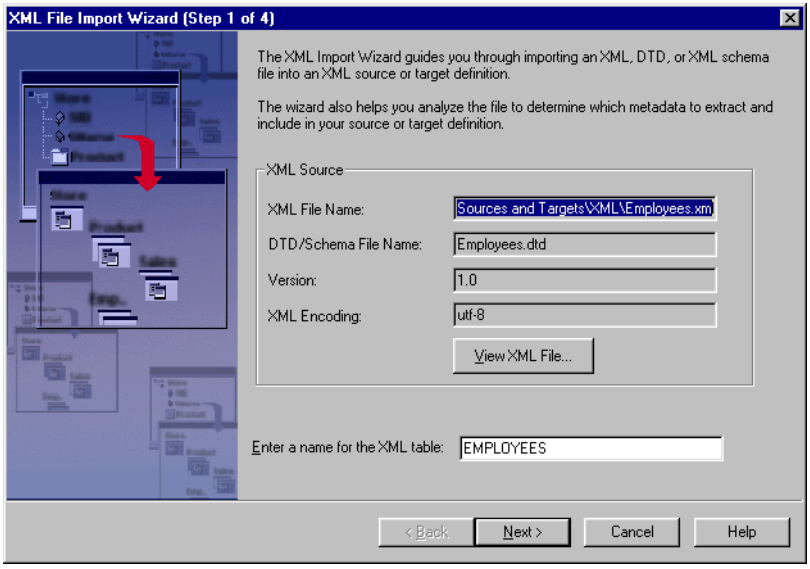
1. In the Designer, switch to the Source Analyzer.

   Note that you are still working in the Orders folder, and your workspace may contain sources from your other lessons. To clear your workspace, right-click in the workspace and choose Clear All.

2. Choose Sources-Import from XML.

3. In the Open XML File dialog, navigate to the Tutorial directory under the PowerMart/ PowerCenter installation directory and select the Employees.xml file. Click Open.

   The XML File Import Wizard appears.

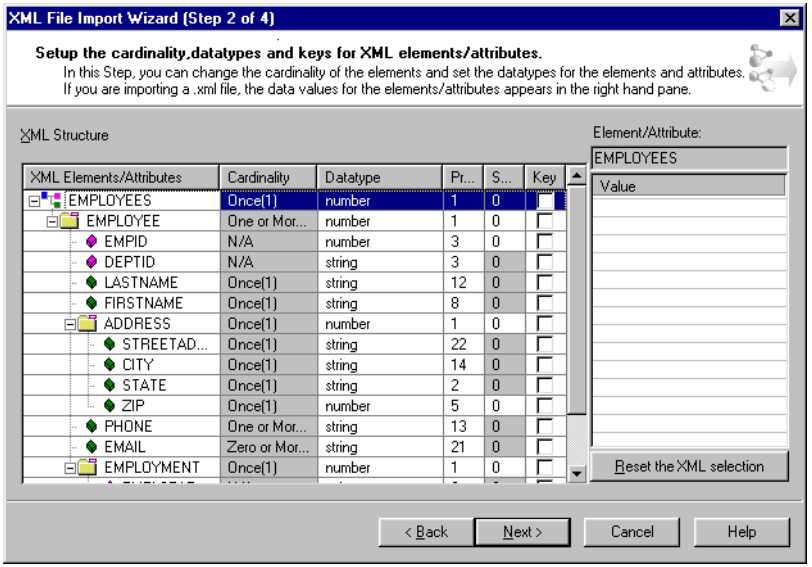


   The XML FileName shows the file path and Employee.xml. The DTD/Schema Filename displays Employees.dtd.

4. Use the default value for all options and click Next.

   Note: The Informatica Server uses the repository code page for XML sources. It ignores the code page defined in the XML source.

In the next window, the XML File Import wizard displays the hierarchy of the Employees.xml file and the data that is available for each element.

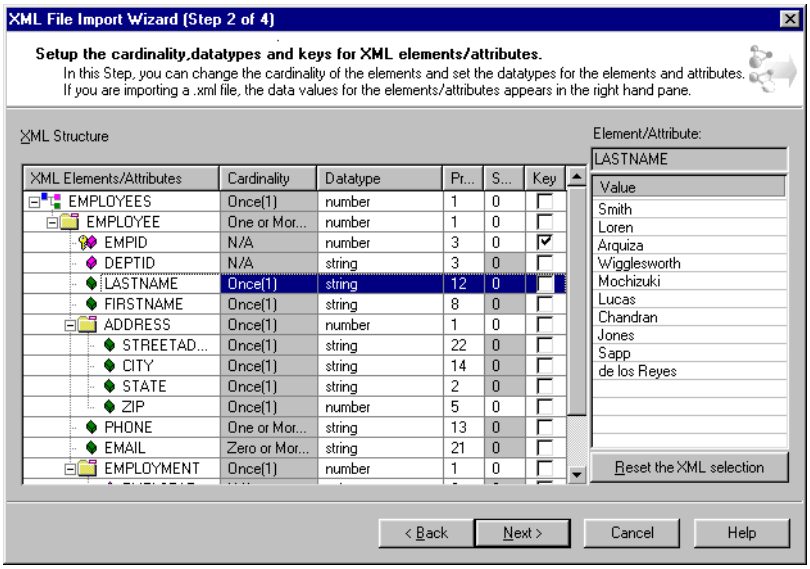


5. Set the EMPID element as a key.
6. Change the datatype, precision, and scale for the element.

   Use the following table as a guide:

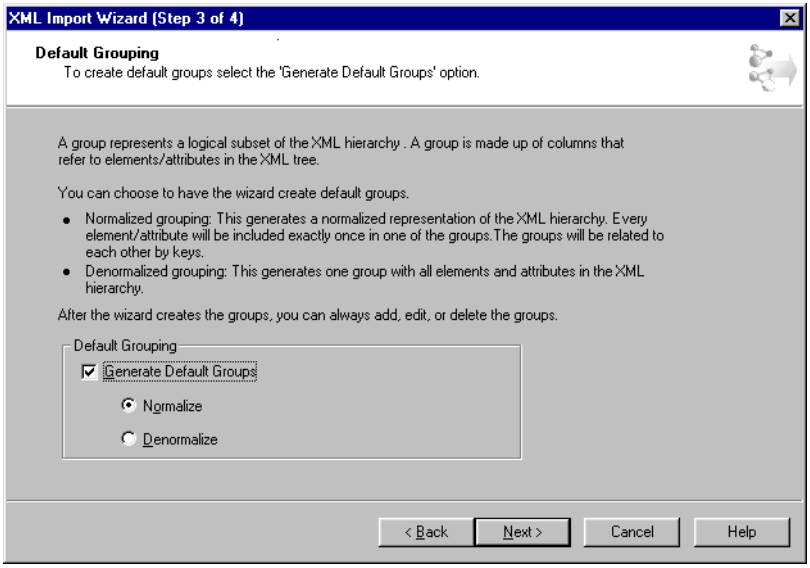
Column	Datatype	Precision	Scale
EMPID	integer		
DEPTID	string	5	
LASTNAME	string	30	
FIRSTNAME	string	30	
STREETADDRESS	string	30	
CITY	string	30	
STATE	string	2	
ZIP	string	10	
PHONE	string	20	
EMAIL	string	60	
EMPLSTAT	string	3	
DATEOFHIRE	string	30	
SALARY	decimal	10	2

You do not need to modify the Employees, Employee, Address, or Employment elements.



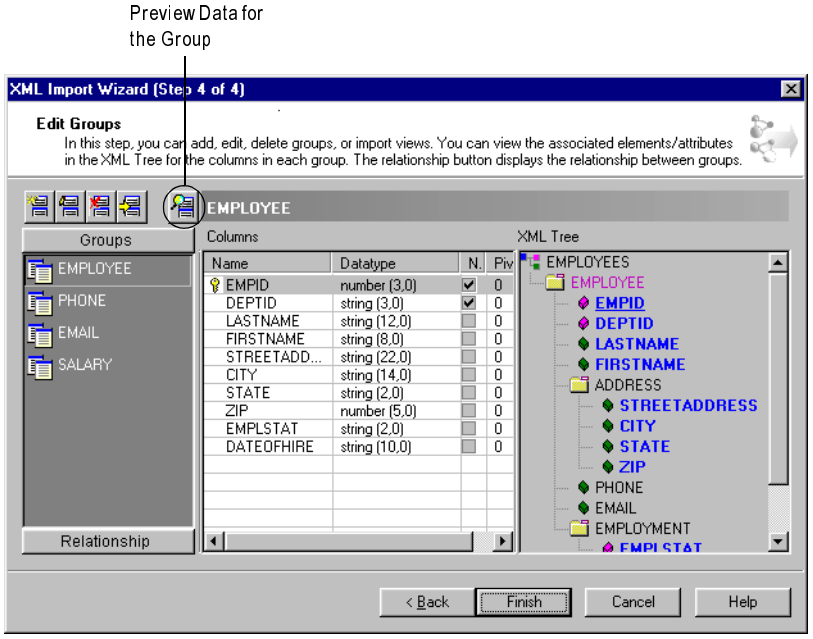
7. Click Next.

8. Select Normalize and Generate Default Groups to have the Wizard create normalized default groups. Click Next.

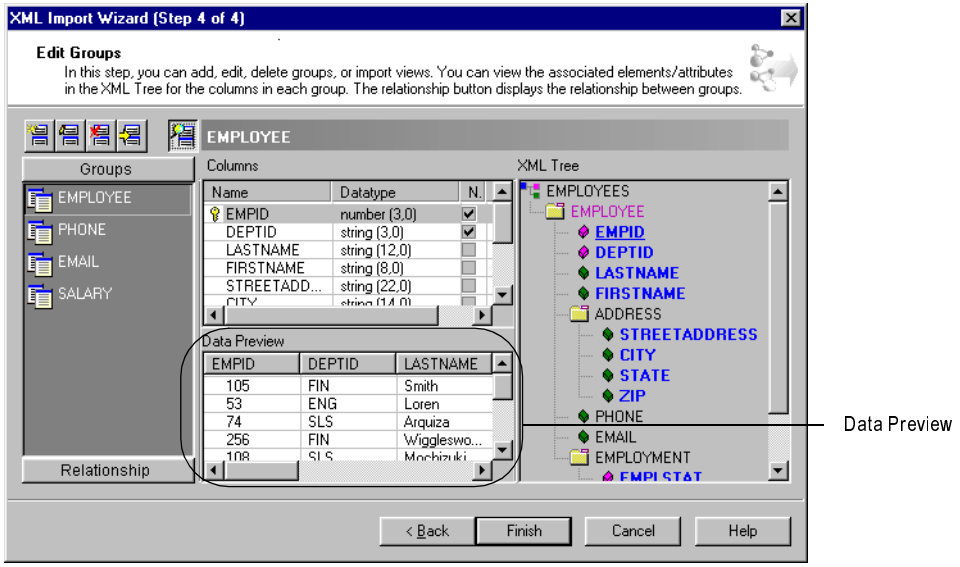


The next window displays the groups that the wizard creates for the source definition and information about those groups. In this window you can preview the XML data and pivot the three occurrences of salary.

9. Click Preview Data for the Group.



The data preview area appears and the data available in the XML file for each column displays. Use the horizontal scroll bar to see all the columns.



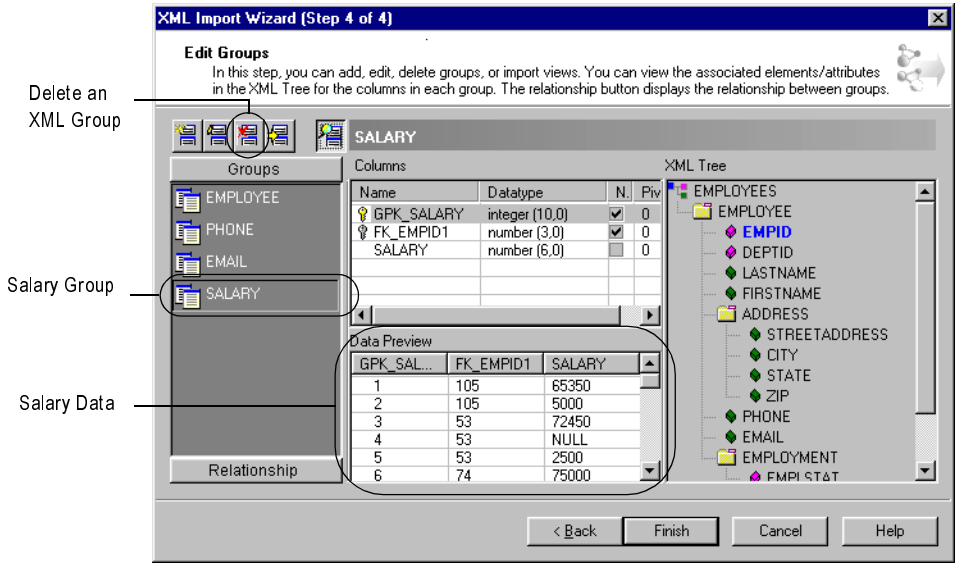
Confirm that the data looks like the following sample:

EMPID	DEPTID	LASTNAME	FIRSTNAME	STREETADDRESS	CITY	STATE	ZIP	EMPLSTAT	DATEOFHIRE
105	FIN	Smith	Martha	335 Westshore Road	Fausta City	CA	97584	PF	12-02-1997
53	ENG	Loren	Pamela	19 B. Candido Street	Zamboanga City	CA	94404	TF	07-05-1998
74	SLS	Arquiza	Noriel	89 Catamaran Drive	Alonzo	CA	94229	PF	03-30-2000
256	FIN	Wigglesworth	Wesley	756 N. First Drive	Jamison	CA	97754	PF	11-19-1996
108	SLS	Mochizuki	Cecilia	45 Edgewater Drive	Palo Colorado	CA	94039	PF	02-16-2000
45	ENG	Lucas	Rommel	22 North Cajon Drive	Logreene	CA	92340	PF	03-30-1997
99	SLS	Chandran	Savitha	2349 English Oak Drive	Ayala	CA	93255	PF	01-21-1999
72	ENG	Jones	Kathleen	42 Tetuan Court	Fausta City	CA	97584	PF	03-30-2000
138	ENG	Sapp	Carmina	251 Topsail Court	Alonzo	CA	94229	PF	11-20-1998
102	HR	de los Reyes	Luis	112 Socorro Drive	Loyola Heights	CA	95333	PF	04-05-1998

Next, you need to pivot the three occurrences of salary into one occurrence in the Employee group. Delete the Salary group and create three columns in the Employee group called BaseSalary, Commission, and Bonus to hold the salary information.

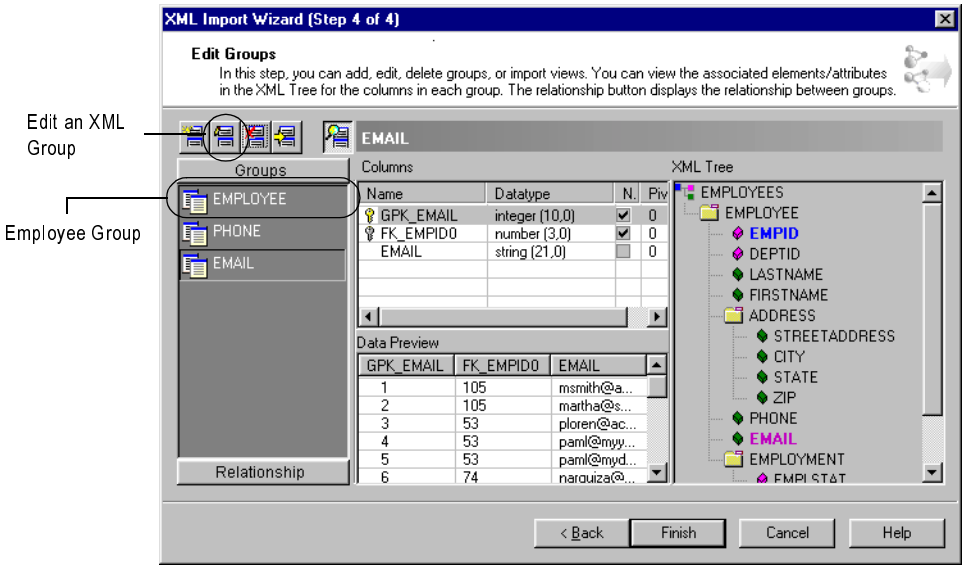
10. Select the Salary group.

The wizard displays the columns for the Salary group and displays the salary data in the data preview.



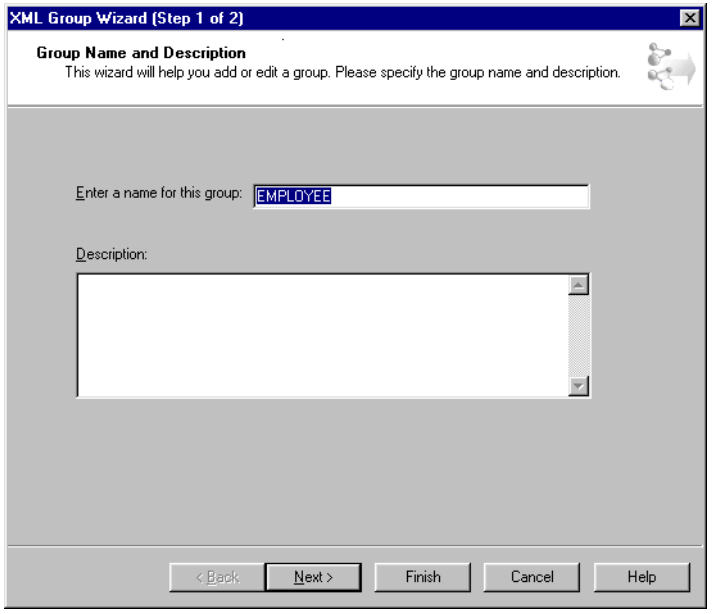
11. Click Delete an XML Group and confirm that you want to delete the Salary group.

The Salary group no longer shows in the list of groups.



12. Next, you need to add the new salary columns to the Employee group. Select the Employee group and click Edit an XML group.

The XML Group Wizard appears.

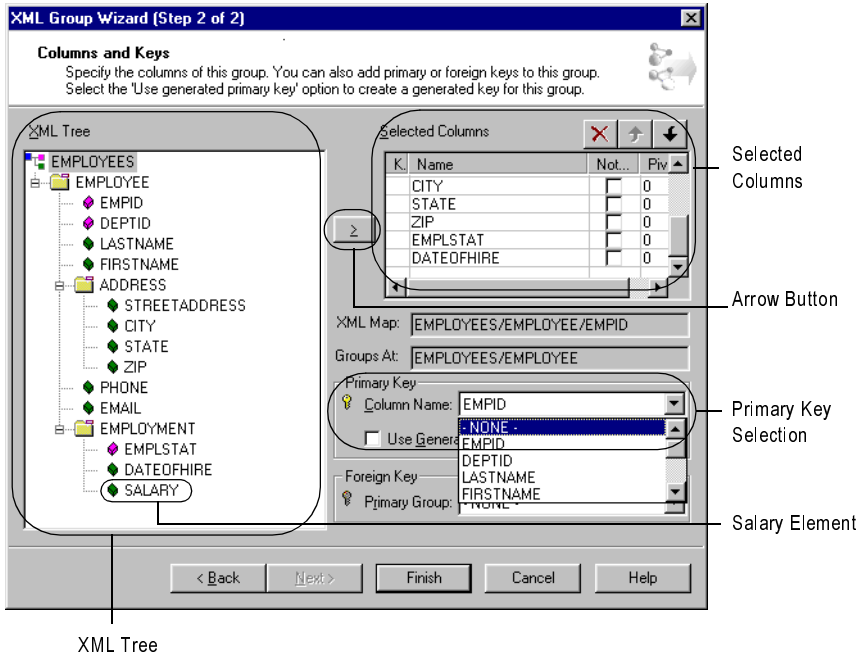


13. Click Next.

In the next window you perform several tasks. You temporarily remove the primary key from the Employee group. You then create three columns in the Employee group and set

the pivot occurrence for each column. After pivoting all three columns, you reset the primary key.

When the wizard displays warnings that the tasks might cause errors, confirm that you want to proceed. The wizard validates each change you make, and since it takes several steps to pivot all the columns, you get several warnings.



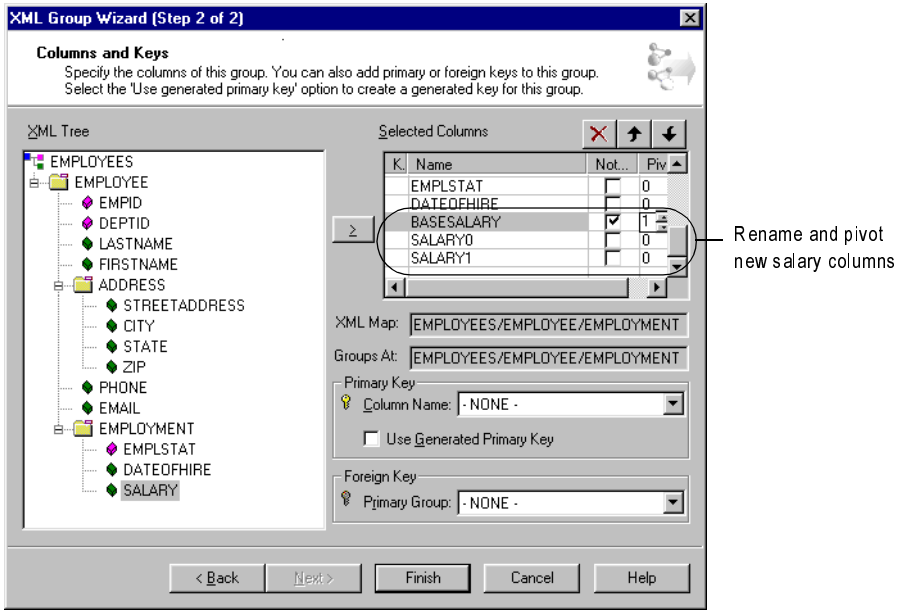
14. In Primary Key, select None for the Column Name.

    When the wizard displays a warning, confirm that you want to proceed.

15. In Selected Columns, scroll down and select DateOfHire.

16. In the XML Tree, select the Salary element and click the arrow button three times.

The wizard adds three new columns below the DateOfHire column, naming them Salary, Salary0, and Salary1.

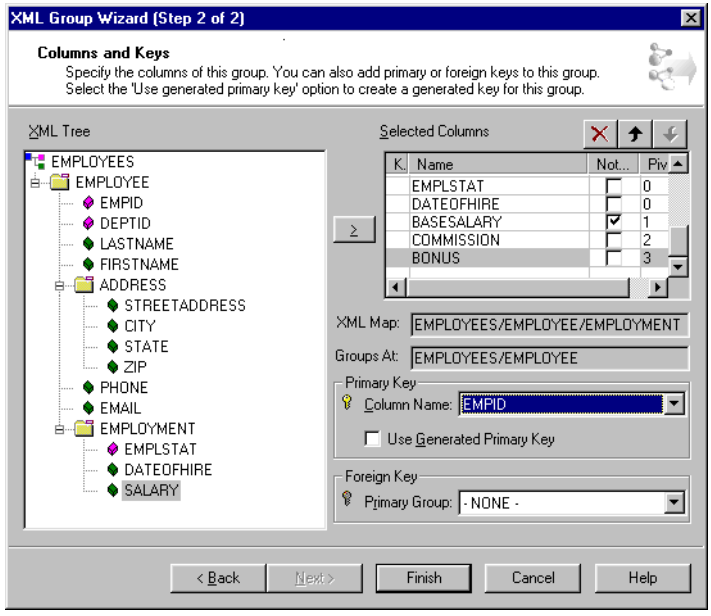


17. Rename the new columns and set the occurrence you want to pivot.

Use the following guide to modify the name and pivot properties:

Column Name	New Column Name	Not Null	Pivot Occurrence
SALARY	BASESALARY	Yes	1
SALARY0	COMMISSION		2
SALARY1	BONUS		3

The wizard displays a warning every time you modify the pivot property. Confirm that you want to continue.

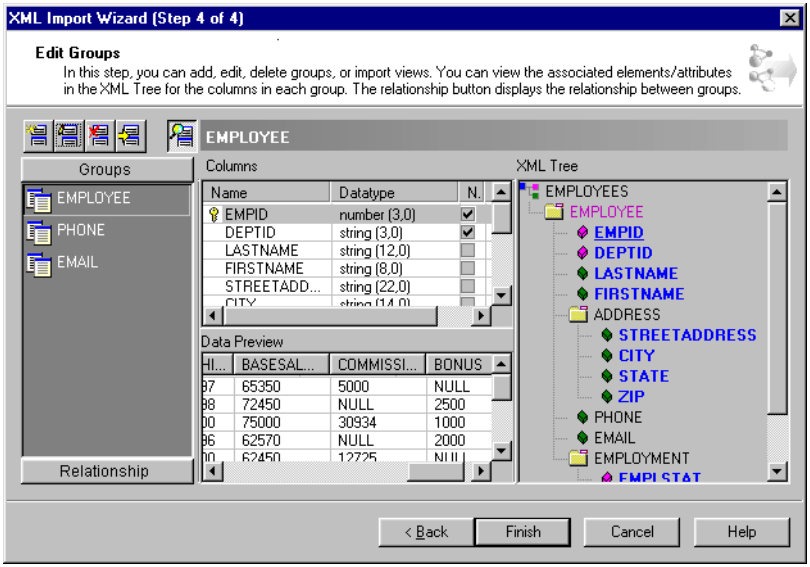


18. Add the primary key back to the group by selecting the column name EmpID in the Primary Key section.

19. Click Finish.

    The XML Group wizard closes.

In the XML Import Wizard, you can see the new columns in the Employee group. The data for the new columns also display in the Data Preview for the Employee group.



Confirm that the data looks like the following sample:

EMPID	DEPTID	LASTNAME	FIRSTNAME	STREETADDRESS	STATE	ZIP	EMPLSTAT	DATEOFHIRE	BASESALARY	COMMISSION	BONUS
102	HR	de los Reyes	Luis	112 Socorro Drive	CA	95333	PF	04-05-1998	39450	NULL	2000
105	FIN	Smith	Martha	335 Westshore Road	CA	97584	PF	12-02-1997	65350	5000	NULL
108	SLS	Mochizuki	Cecilia	45 Edgewater Drive	CA	94039	PF	02-16-2000	62450	12725	NULL
138	ENG	Sapp	Carmina	251 Topsail Court	CA	94229	PF	11-20-1998	95675	NULL	7500
256	FIN	Wigglesworth	Wesley	756 N. First Drive	CA	97754	PF	11-19-1996	62570	NULL	2000
45	ENG	Lucas	Rommel	22 North Cajon Drive	CA	92340	PF	03-30-1997	62500	NULL	4500
53	ENG	Loren	Pamela	19 B. Candido Street	CA	94404	TF	07-05-1998	72450	NULL	2500
72	ENG	Jones	Kathleen	42 Tetuan Court	CA	97584	PF	03-30-2000	132000	NULL	4000
74	SLS	Arquiza	Noriel	89 Catamaran Drive	CA	94229	PF	03-30-2000	75000	30934	1000
99	SLS	Chandran	Savitha	2349 English Oak Drive	CA	93255	PF	01-21-1999	74975	23200	NULL

Each of the three rows of salary, becomes a column for each row of employee in the Employee group.

20. Click Finish then choose Repository-Save.

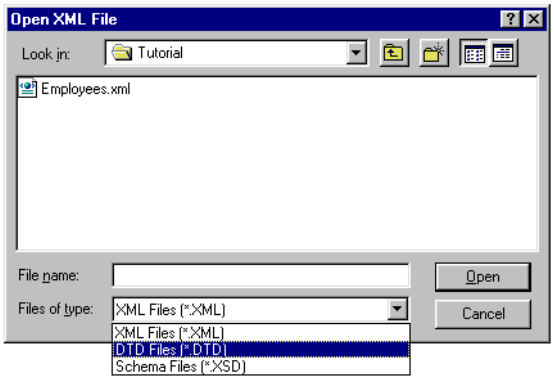
Creating Targets

The mapping that you create in this lesson passes data through router transformations to two separate targets. Since the structure for the target data is the same, you can use two instances of one target definition. Create the target definition from a DTD file.

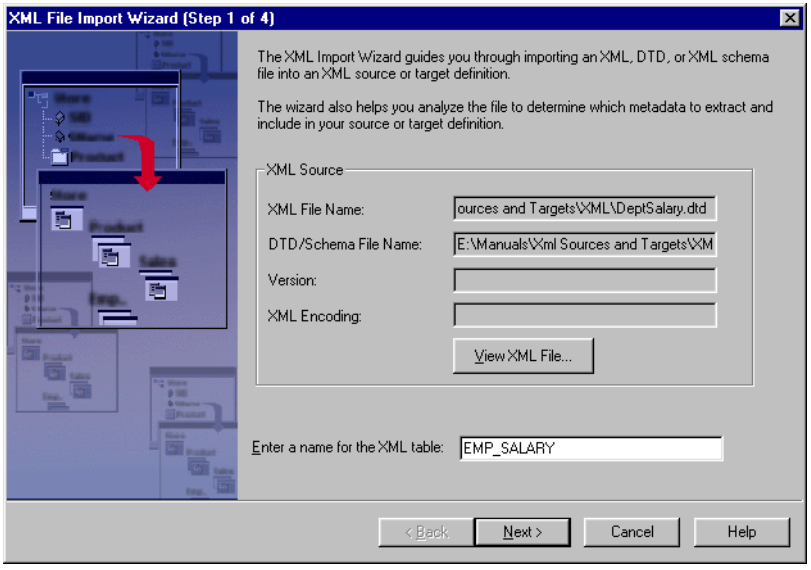
To import the XML target:

1. In the Designer, switch to the Warehouse Designer. If your workspace contains targets from other lessons, right-click in the workspace and choose Clear All.
2. Choose Targets-Import XML-From File.

3. In the Open XML File dialog, select DTD Files from the Files of Type list.



4. Navigate to the Tutorial directory and select the DeptSalary.dtd file. Click Open.
5. The XML File Import Wizard appears.



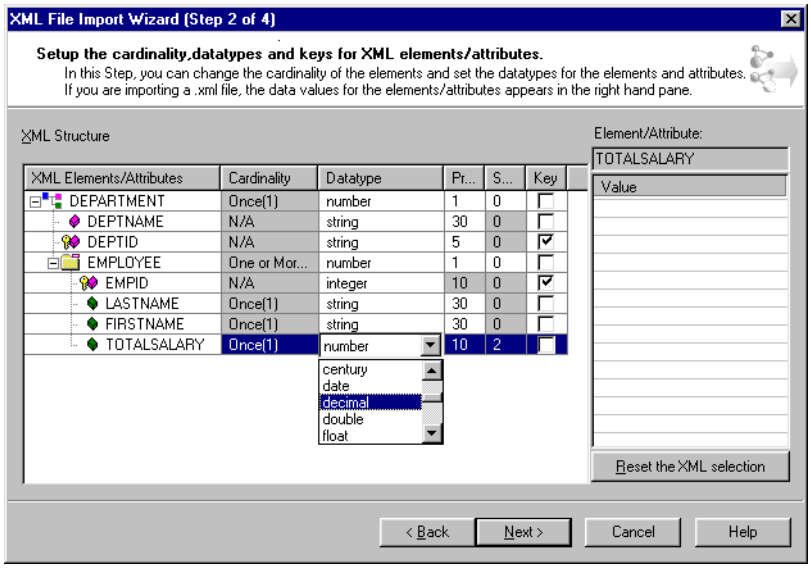
6. Change the XML table name to EMPL_SALARY and click Next.

   The wizard displays the properties of the elements in the hierarchy.
7. Set the DeptID and EmpID elements as keys.

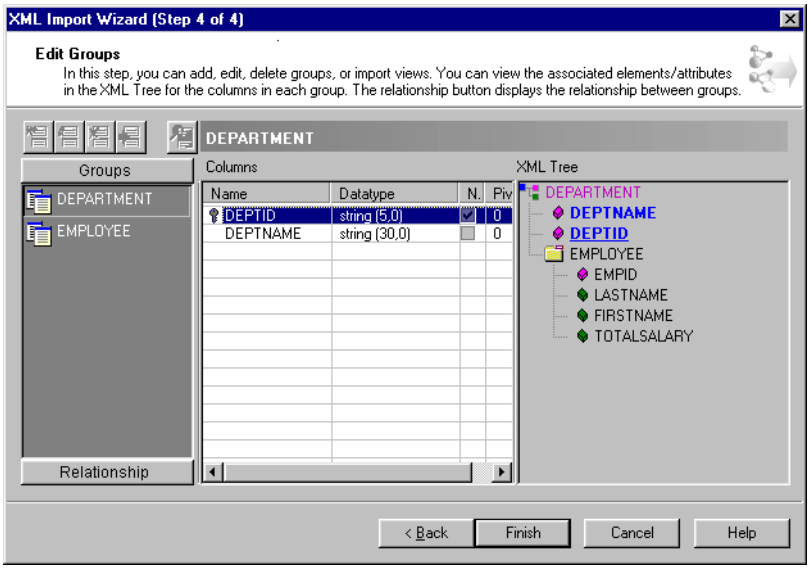
8. Change the datatype, precision, and scale for each element, using the following table as a guide:

Column	Datatype	Precision	Scale
DEPTNAME	string	30	
DEPTID	string	5	
EMPID	integer		
LASTNAME	string	30	
FIRSTNAME	string	30	
TOTALSALARY	decimal	10	2

You do not need to modify the Department and Employee elements.



9. Click Next.



The wizard displays the details of the Department and Employee groups.

10. Click Finish.

11. Choose Repository-Save.

Creating the New Mapping

Create a new mapping for transforming the employee data. In this mapping, you add the following objects:

- The Employees XML source definition you created
- The relational source definition Department you created in a previous lesson
- Two instances of the Empl_Salary target definition you created
- An Expression transformation to calculate the total salary for each employee
- Two Router transformations to select the data for the Sales and Engineering departments

You pass the data from the Employees source through the Expression and Router transformations before sending it to two separate targets. You also pass data from the relational table through another Router transformation to add the departmental names to the targets. You need the data only for two departments: sales and engineering.

To create the new mapping:

1. In the Designer, switch to the Mapping Designer and create a new mapping.

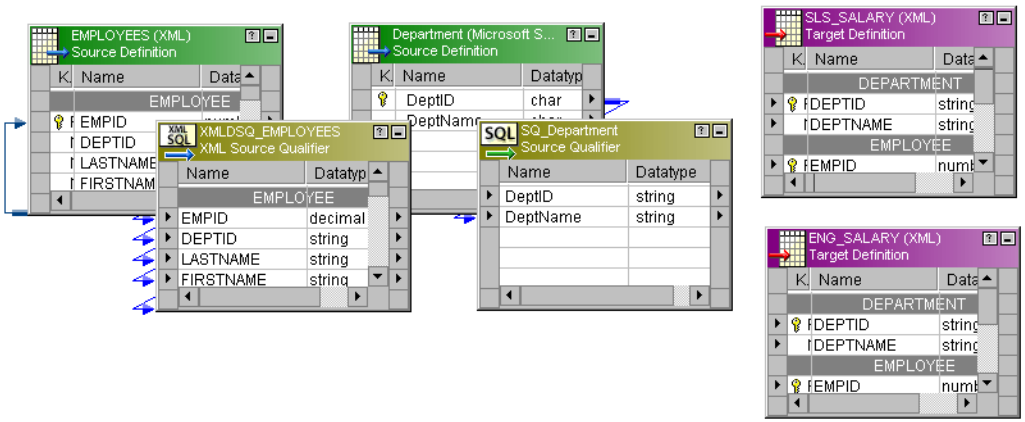
2. Name the mapping m_EmployeeSalary.

3. From the list of sources, select the XML source Employees you just created and drag it into the mapping. Then select the relational source Department and drag it into the mapping.

   By default, the Designer creates a source qualifier for each source.

4. From the list of targets, drag the EMPL_SALARY to the workspace. Rename this first instance of the target definition SLS_SALARY.

5. Drag the EMPL_SALARY target definition to the workspace again to create a second instance of the same target definition. Rename this instance ENG_SALARY.



6. Choose Repository-Save.

   Because you have not yet completed the mapping, the Designer displays a warning that the mapping m_EmployeeSalary is invalid.

   To complete the mapping, add an Expression transformation and two Router transformations. Then create a data flow from the source definitions to the Expression and Router transformations and then to the two target definition.

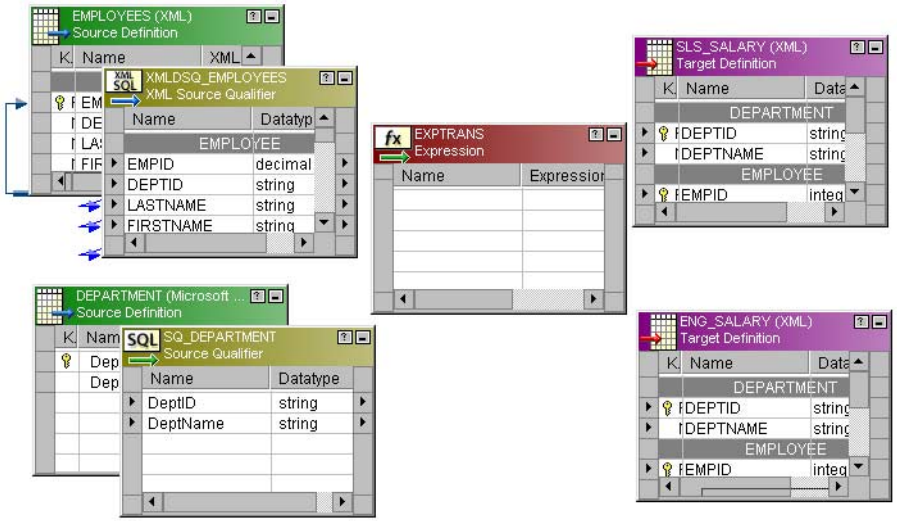
Creating an Expression Transformation

In this exercise, you use an Expression transformation to calculate the total salary for each employee. You use the BaseSalary, Commission, and Bonus as input columns to the Expression transformation and create a TotalSalary column as output.

To add the Expression transformation to the mapping:

1. Click the Expression button on the Transformations toolbar.
2. Click in the area to the right of the XML Source Qualifier.

The new transformation appears.



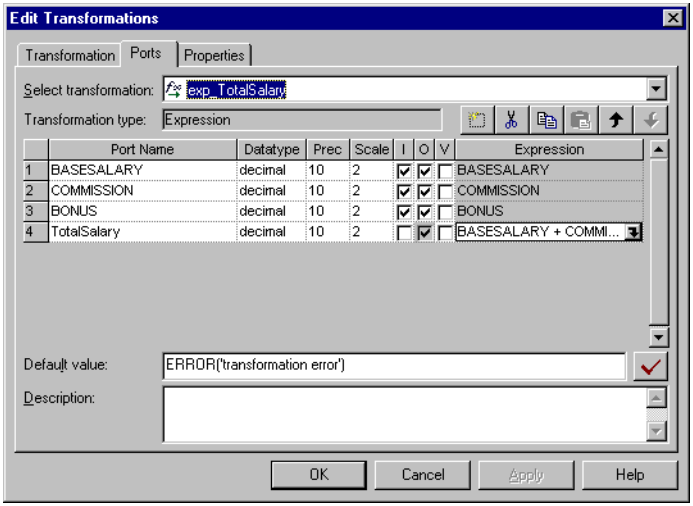
3. In the XML Source Qualifier, select the columns BaseSalary, Commission, and Bonus and drag them to the Expression transformation.

   Three input ports appear in the Expression transformation. The output ports of the columns in the source qualifier are linked to the input ports of the columns in the Expression transformation.

4. Double-click the title bar of the Expression transformation.

5. In the Transformation tab of the Edit Transformations dialog box, change the name of this transformation to exp_TotalSalary.

6. In the Ports tab, add a new output port, TotalSalary. Use decimal datatype with precision of 10 and scale of 2.



7. Enter the following expression for TotalSalary:

```
BASESALARY + COMMISSION + BONUS
```

8. Validate the expression.
9. Close the Expression Editor, then close the Edit Transformations dialog box.
10. Choose Repository-Save.

Creating the Router Transformations

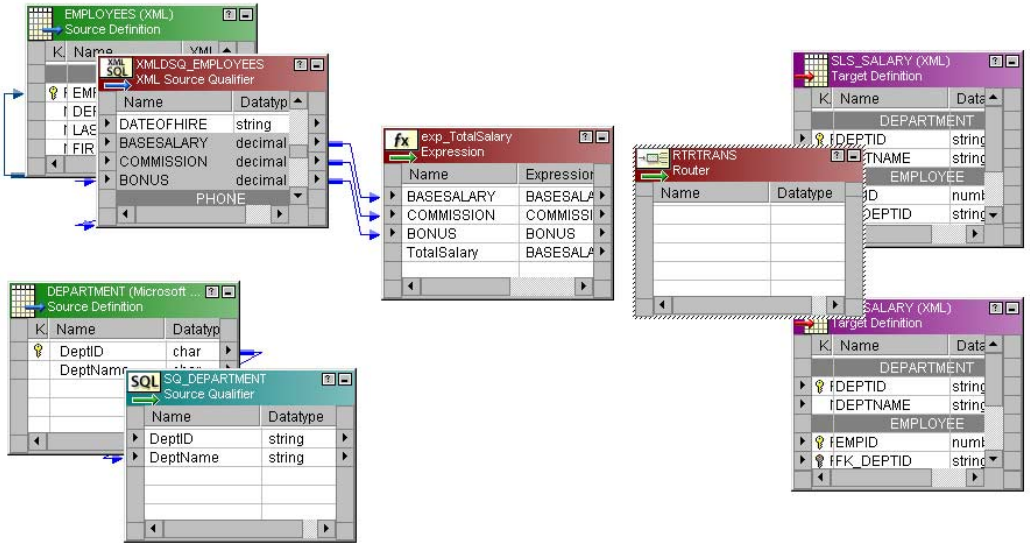
You add two Router transformations to the mapping. In each of the Router transformations, create two groups. One group returns true for rows where the DeptID column contains 'SLS'. The other group returns true if the DeptID contains 'ENG'. All other rows that do not meet either condition go into the default group.

Create the Router transformation for the employee salary data first. Then create the Router transformation for the department data.

To create the first Router transformation:

1. Click the Router button on the Transformations toolbar.
2. Click in the area between the Expression transformation and the target definitions.

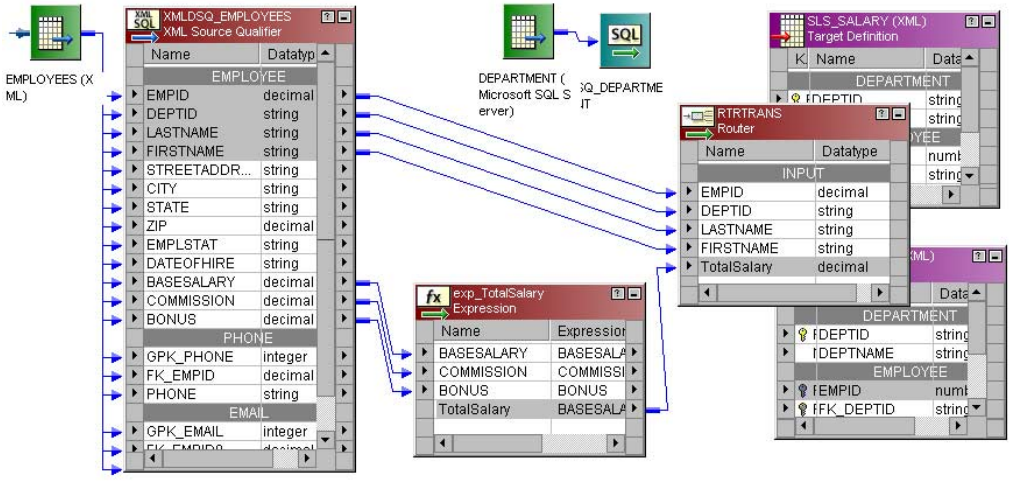
   The new transformation appears when you release the mouse button.



3. In the XML Source Qualifier, select the following columns and drag them to the Router transformation:
   - EmpID
   - DeptID

- LastName
- FirstName

4. In the exp_TotalSalary Expression transformation, select the TotalSalary column and drag it to the Router transformation.



The Designer automatically creates an input group and adds the columns you drag from the XML Source Qualifier and the Expression Transformation.

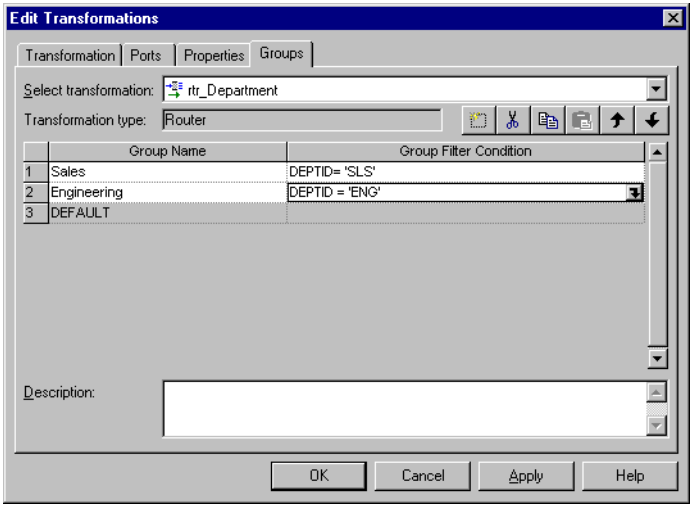
5. Double-click the title bar of the Router transformation.

6. In the Transformation tab of the Edit Transformations dialog box, change the name of the transformation to rtr_Department.

7. In the Groups tab, add two new groups. Change the group names and set the filter conditions. Use the following table as a guide:

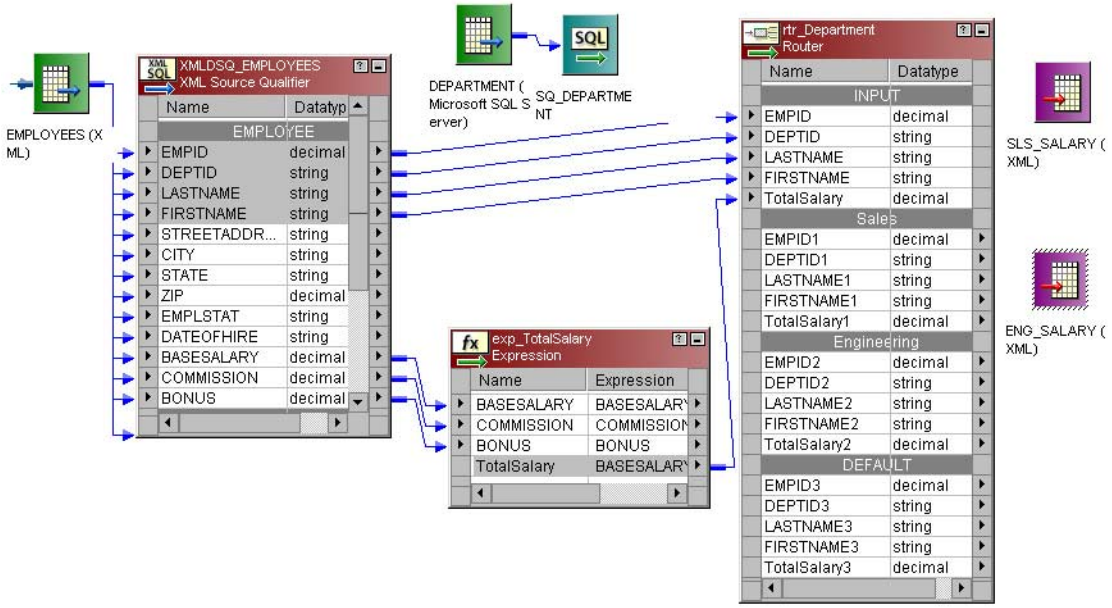
Group Name	Filter Condition
Sales	`DEPTID = 'SLS'`
Engineering	`DEPTID = 'ENG'`

The Designer automatically adds a default group to the list of groups. You cannot add a filter condition to the default group.



8. Close the Edit Transformations dialog box.

9. In the workspace, expand the rtr_Department transformation to see all the groups and columns.



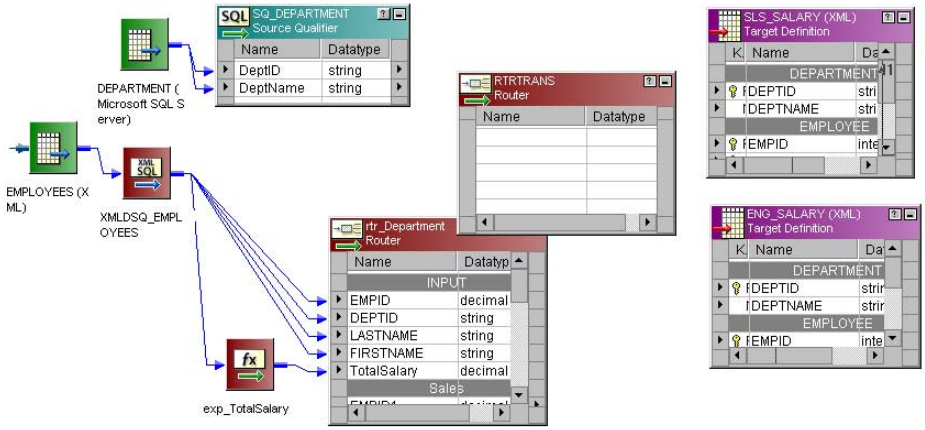
10. Choose Repository-Save.

Now you create the Router transformation to filter the Sales and Engineering department data from the Department relational source.

To create the second Router transformation:

1. Click the Router button on the Transformations toolbar.
2. Click in an area between the Expression transformation and the target definitions.

   The new transformation appears:

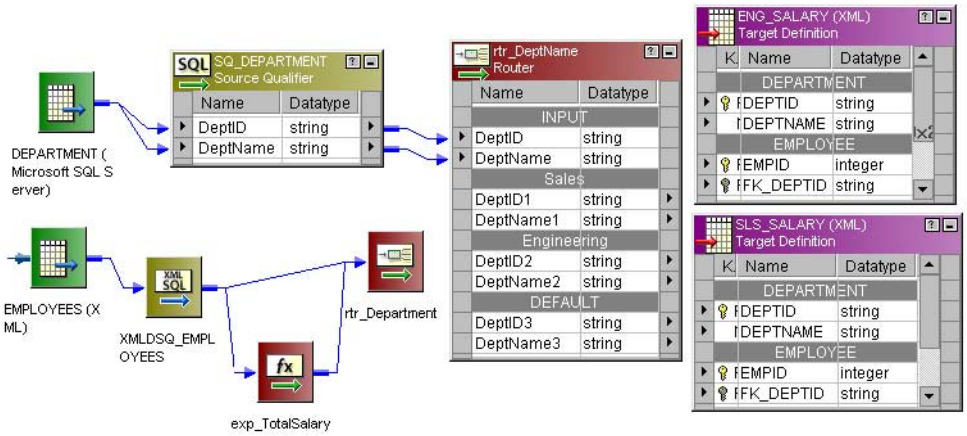


3. Select the DeptID and DeptName columns in the SQ_Department relational source qualifier, and drag them to the new Router transformation.
4. Double-click the title bar of the new Router transformation.
5. In the Transformation tab of the Edit Transformations dialog box, change the name of the transformation to rtr_DeptName.
6. In the Groups tab, add two new groups. Change the group names and set the filter conditions using the following table as a guide. These are the same filter conditions as those you added to the rtr_Department Router transformation:

Group Name	Filter Condition
Sales	`DEPTID = 'SLS'`
Engineering	`DEPTID = 'ENG'`

7. Close the Edit Transformations dialog box.

8. In the workspace, expand the rtr_DeptName transformation to see all the groups and columns.



9. Choose Repository-Save.

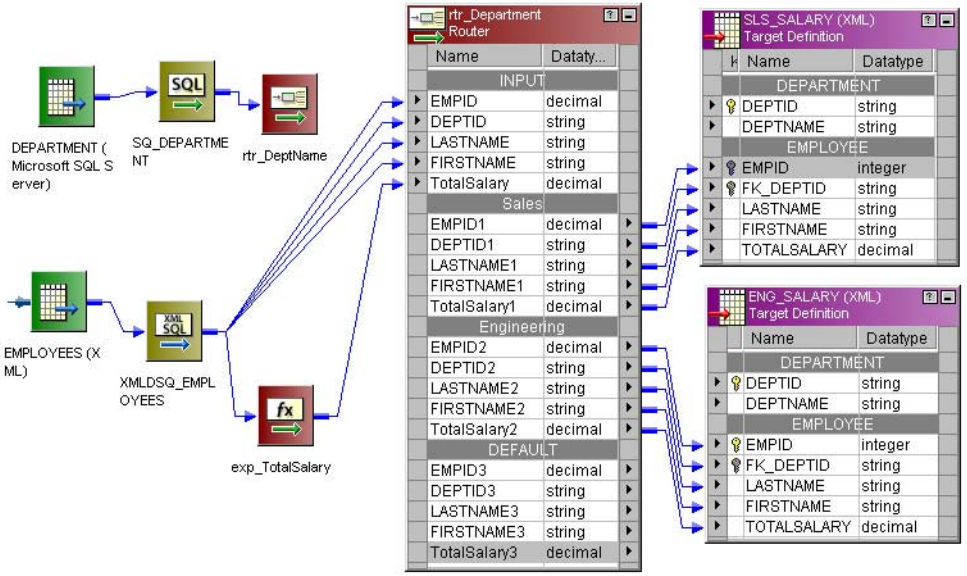
Completing the Mapping

The final task is to connect data from the Router transformations to the columns in the targets.

To complete the mapping:

1. Connect the following ports from rtr_Department groups to the ports in the XML target definitions:

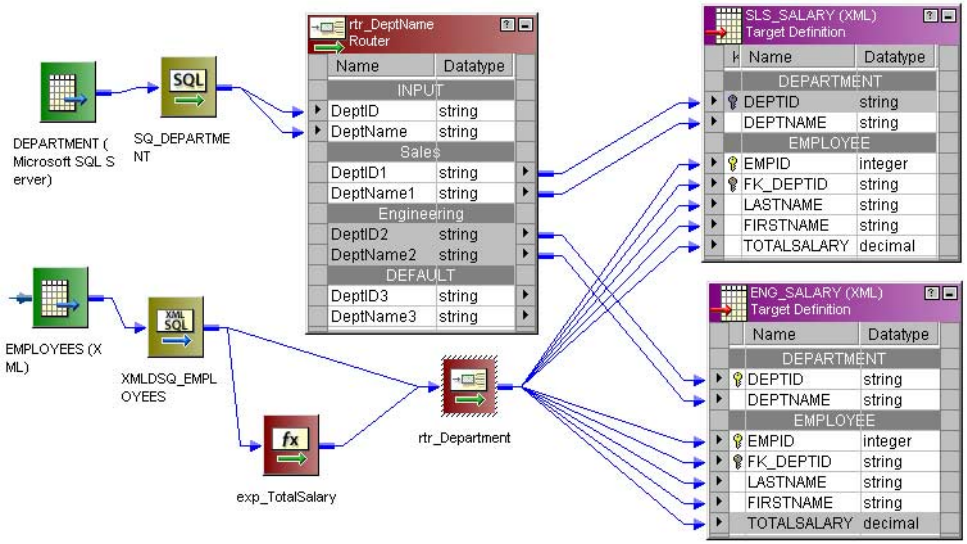
Router Group	Router Port	Target	Target Group	Target Port
Sales	EMPID1	SLS_SALARY	EMPLOYEE	EMPID
	DEPTID1			FK_DEPTID
	LASTNAME1			LASTNAME
	FIRSTNAME1			FIRSTNAME
	TotalSalary1			TOTALSALARY
Engineering	EMPID2	ENG_SALARY	EMPLOYEE	EMPID
	DEPTID2			FK_DEPTID
	LASTNAME2			LASTNAME
	FIRSTNAME2			FIRSTNAME
	TotalSalary2			TOTALSALARY



2. Connect the following ports from the rtr_DeptName transformation to the target ports:

Router Group	Router Port	Target	Target Group	Target Port
Sales	DeptID1	SLS_SALARY	DEPARTMENT	DEPTID
	DeptName1			DEPTNAME

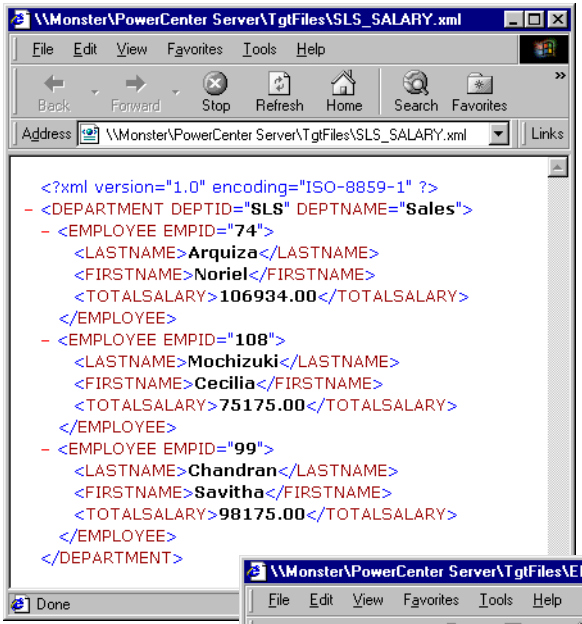
Router Group	Router Port	Target	Target Group	Target Port
Engineering	DeptID2	ENG_SALARY	DEPARTMENT	DEPTID
	DeptName2			DEPTNAME



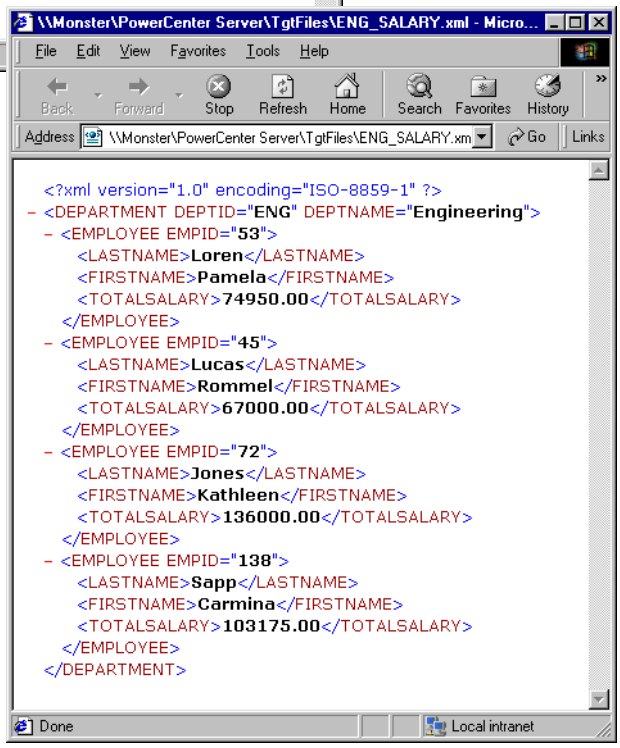
3. Choose Repository-Save.

The mapping is now complete. When you save the mapping, the Designer displays a message that the mapping m_EmployeeSalary is valid.

You can create a new session and run this mapping. When you run a session with this mapping, the Informatica Server returns the following target XML files:



Resulting data in the Sls_SALARY.xml



Resulting data in the Eng_SALARY.xml

Chapter 5

Sample Repository

This chapter contains information about the following:

Overview

The sample repository enables you to integrate web-based data from multiple channels into your data warehouse for e-business analysis of customer, sales, and supply-chain activity. It works in conjunction with PowerMart/PowerCenter and your existing e-business data management tool, to perform lookups, data transformations, and data analysis of common web-based file formats.

The sample repository delivers a set of reusable transformations, mapplets, and mappings that you can use as a template for sourcing e-business data in standard and extended web log formats. The sample repository includes reusable transformations that perform the following tasks:

- Sort file data.
- Call Perl functions from mappings.

The predefined transformations in the sample repository reduce the complexity in accessing web data. You can also modify the transformations and mappings to access and manipulate data relevant to your organization's e-business applications.

Note: Please note that the sample repository components only provide a small subset of the functionalities required to process web logs for clickstream analysis. If you are interested in a complete clickstream analysis application, Informatica's eSite application provides an end-to-end clickstream analysis solution.

eSite is built on the PowerCenter platform and provides the following:

- A data model for clickstream analysis
- Mappings to extract and load web log data
- Preconfigured metrics and reports
- A reporting tool for ad hoc reporting

E-Business Data Sources

To facilitate real-time collection and analysis of web log data, the sample repository supports these common file formats for e-business data:

- **Web log files in CLF and IIS formats.** Web log files are a rich source of data about customer traffic on web sites. The sample repository includes sources, transformations, mapplets and mappings that are ready to use for reading the common log format used by many browsers, including Netscape and Apache. The mappings also support the Microsoft IIS log file format.
- **Web data handling.** To help you manage all your web data more efficiently, the sample repository provides reusable transformations that allow you to add sort logic to your mappings and to call any number of Perl routines from a PowerCenter session.

Sample Repository Components

The sample repository includes the following components:

- **Sample repository.** A sample repository you can restore to a database.
- **Repository objects.** Sample sources, reusable transformations, mapplets, business components, and mappings.
- **Shared libraries.** Compiled procedures. Many of the sample repository components are delivered as Advanced External Procedures. The shared libraries for those procedures are included as part of the installation.
- **Sample files.** Web log files. These samples demonstrate how to use the sample repository for typical web data.

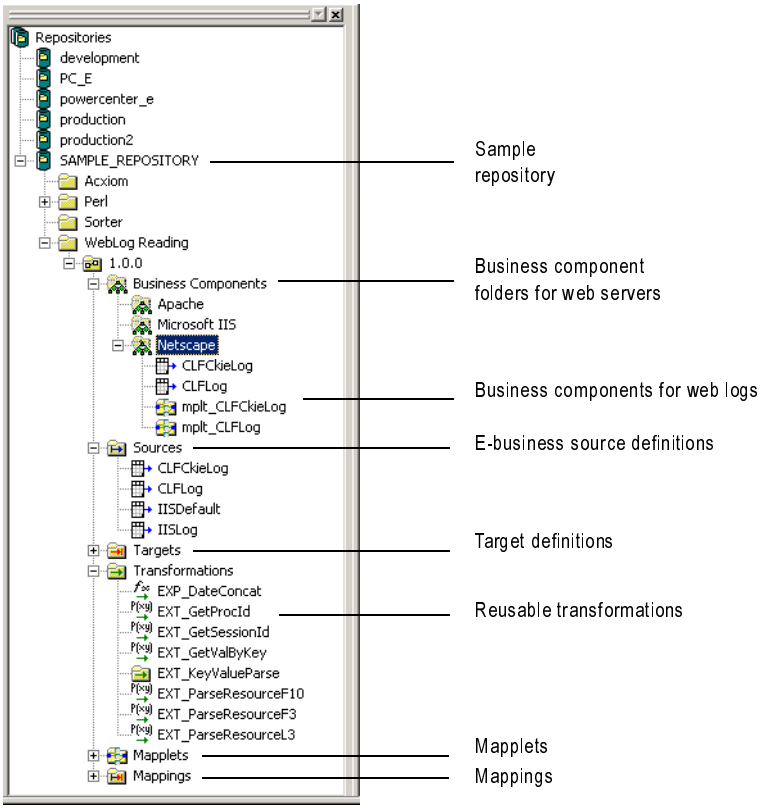
The sample repository objects are stored in read-only folders. You can create shortcuts to the objects, or you can copy the objects into your working folder.

Sample Repository

The sample repository has different folders that contain repository objects that you can use as templates for developing your web data solution.

Figure 5-1 displays the sample repository folders in the Navigator window:

Figure 5-1. Sample Repository Folder Structure



The sample repository folders all contain the same hierarchy as other repository folders.

Repository Objects

The sample repository includes the following repository objects:

- Reusable transformation
- Mapplets
- Mappings

Reusable Transformations

The sample repository contains reusable transformations that perform data extraction on common web data. Reusable transformations allow you to build logic in a single transformation to use in multiple mappings or mapplets. The reusable transformations in the sample repository are used in the sample mapplets and mappings.

For example, web log data is stored in a log format. By adding one of the reusable web log transformations to a mapping, you can strip relevant values from the log file and write this web data to your data warehouse.

For more information about reusable transformations, see "Transformations" in the *Designer Guide*.

Mappings

The sample repository includes a number of sample mappings that demonstrate how to use the sample repository components.

The mappings use a combination of sample source definitions, mapplets, and transformations to process typical e-business data.

For more information about mappings, see "Mappings" in the *Designer Guide*.

Mapplets

The sample repository includes two mapplets in the WebLog Reading folder. Mapplets allow you to build logic into source definitions and transformations that you can use in multiple mappings.

The sample repository mapplets filter relevant data from web log files. For example, the mapplet, mplt_CLFCkieLog, filters records from a Netscape web log with cookie support. The supplied mapplets are used in the sample repository mappings, but you can also use them to build your own mapplets or mappings.

For more information about using mapplets, see "Mapplets" in the *Designer Guide*.

Shared Libraries

The shared libraries shipped with the sample repository must reside on the same machine as the Informatica Server in order to run sessions using the pre-built mappings, or to run sessions with one of the reusable transformations in your own mappings.

During installation, the shared libraries are installed in the PowerCenter Server directory.

Sample Files

In addition to sample sources in the prebuilt mappings, the sample repository supplies sample web log files. The sample files are installed in the PowerCenter Server directory.

Accessing Sample Repository Folders

The sample repository delivers several read-only folders that you access through shortcuts or by making copies of objects. Before you start using the sample repository, copy the sample repository objects into your working folder or create shortcuts from your working folder to the sample repository folder.

You can create shortcuts under the following repository structures:

- Shortcut to a global repository from a local repository within the same domain.
- Shortcut to a folder from a folder within the same repository.

Guidelines for Shortcuts, Copies, and Changes

To maintain object validity in your working folders when you upgrade the sample repository, follow the guidelines below:

- Do not make any changes to the sample repository folder.

  Any change that you make to the folder will be lost in future upgrades.
- Use shortcuts to the sample repository folder only when you have no need to edit the objects you access.

  For the sample repository, most objects are intended to be used as templates, in which case you can make copies and build new objects. Copied objects do not reflect changes made to the original objects during any future upgrade.

Using the Sample Mappings

Since the sample mappings provided with the sample repository are intended only as templates, you can use the prebuilt mappings to experiment with the transformations and review how the sample code works. When you are ready to develop your own mappings, you can customize them.

Guidelines

As with other mappings, when you create physical target tables and sessions, remember these guidelines:

- Create the target tables in your target database using the Generate/Execute SQL option in the Warehouse Designer. When you create targets, clear the option to generate foreign keys. This maximizes session performance and eliminates synchronization errors that result from loading dependent tables.
- When you create a session, select the Truncate Table option in the session properties. This ensures that you do not load duplicate data into the data warehouse.

Prebuilt Mappings

If you want to use the a sample mapping without making changes to any of the objects in the mapping, create a shortcut to the mapping in a sample repository folder.

Customized Mappings

If you want to customize a sample repository mapping, copy it from a sample repository folder. When you copy a mapping, you copy all the objects used in the mapping. You can edit the mapping in the same way you edit any mapping.

Chapter 6

Installing and Configuring the Sample Repository

This chapter contains information about the following:

Overview

The sample repository contains repository objects and business components to help you understand and extract data from web file formats. After you install the sample repository, you also need to configure the PowerMart/PowerCenter client machines to create a shortcut to the sample repository framework.

Installation Prerequisites

Before you can install and configure the sample repository, you must first complete the following steps:

- **Install and configure PowerMart/PowerCenter.** For more information, see the *Installation and Configuration Guide*.
- **Create a database.** Create a database to restore the sample repository. You also need to configure PowerMart/PowerCenter to connect to this database.
- **Install and configure the third-party software you are using.** For example, to use the sample repository with Perl, you must install this product before you add the sample repository.

Installing and Configuring the Sample Repository

Complete the following steps to configure the sample repository:

1. Restore the sample repository.
2. Set up the repository objects.
3. Configure other components.

Step 1. Restoring the Sample Repository

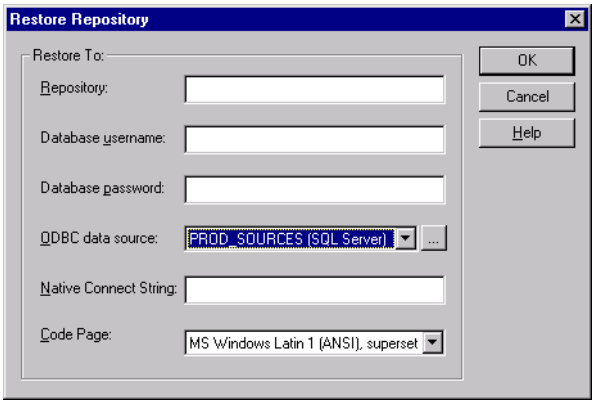
Before you can restore your sample repository, you need to establish connectivity between the machine hosting the PowerMart/PowerCenter Client and the repository database. For information on connectivity issues, such as using native drivers, see "Installing and Configuring the Informatica Client" in the *Installation and Configuration Guide*.

You restore the sample repository from the sample.rep file located in the \pc.e directory of your PowerMart/PowerCenter Client directory.

To restore the sample repository:

1. In the Repository Manager, choose Repository-Restore Repository.
2. Scroll to locate the sample.rep file. Select the file, then click OK.

   The Repository Manager states that a repository must be created before restoring the existing repository.
3. Click OK.



   When you restore the repository, the Repository Manager creates a repository before restoring the backup.
4. Enter the following information:

Restore Repository Field	Required/ Optional	Description
Repository	Required	Name for the restored repository. Each repository name must be unique within the domain and easily distinguished from all other repositories.
Database Username	Required	Username required to connect to the database. This login must have the appropriate database permissions to create the repository.
Database Password	Required	Password associated with the database username.

Restore Repository Field	Required/ Optional	Description
ODBC Data Source	Required	Data source used to connect to the database.
Native Connect String	Required	Connect string identifying the location of the database. See Table 2-1 on page 22.
Code Page	Required	Character set associated with the repository. Must be a superset of the code page of the repository you want to restore.

5. Click OK.
6. When the Repository Manager completes the restore process, click OK.

Step 2. Setting up the Repository Objects

The sample repository contains folders that you can use in the sample repository or copy to your working repository. To make the contents in the sample repository available to users, copy any folders you want to use into your working repository. Informatica recommends that you copy the sample repository folder to a global repository. During the installation process, you must temporarily allocate database space to store the restored repository. After you copy the folder to your working repository, you can delete the sample repository.

Copying the Folder to Global Repository

After you restore the sample repository, you need to make its contents available to other users. Copy the sample repository folders you want to use to your global repository. To copy a folder you must be a repository user in both source and target repositories.

You must have one of the following sets of privileges and permissions in the source repository:

- Browse Repository privilege with read permission on the original folder
- Administer Repository privilege with read permission on the original folder
- Super User privilege

You must have one of the following privileges in the target repository:

- Administer Repository
- Super User

For details on permissions and privileges, see "Repository Security" in the *Repository Guide*.

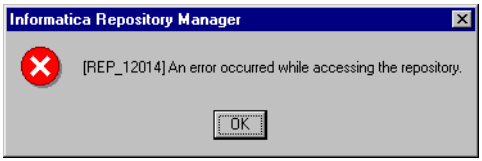
If you do not have a global repository, copy the folder to a repository where you plan to develop mappings.

Note: Before you copy any sample repository folder, verify that you have enough free space in your target repository database.

To copy a sample repository folder:

1. In the Repository Manager, connect to the global repository.
2. Connect to the sample repository, and select a sample repository folder.
3. Choose Edit-Copy.
4. In the Navigator, select the target repository, and choose Edit-Paste.
5. Follow the steps of the Copy Folder Wizard to complete the copy.

If you do not have enough space in the target repository, the Repository Manager displays the following error:



The Output Window in the Repository Manager issues the following error message:

```
Copy failed, rolling back changes...
```

You need to increase the database space and copy the folder again.

Step 3. Configuring Other Components

Depending on the sample repository components you want to use, you might need to configure or install third-party software products to support your e-business solution.

For more information, see the following:

- "Reading Web Logs" on page 127
- "Calling Perl Functions" on page 149
- "Sorting Data with Web Sorter Transformation" on page 159

Chapter 7

Reading Web Logs

This chapter contains the following information:

Overview

With the emergence of new paradigms for e-business, web server logs have become a valuable source of information about web site traffic. The sample repository includes a set of pre-built mapplets and mappings that allow you to source web log data. Using these sample repository objects, you can deploy data models that extract that data and make it available for real-time analysis in your data warehouse. Since the sample repository is built with the core Informatica platform, you can combine web data with any other data source in your environment.

Web logs are complex flat files that contain fixed fields to track requests and web traffic. These log files are created on both Intranet and Internet web servers. The sample repository procedures for web logs parse the most common formats of web log files and enable you to use the web logs as data sources in mappings.

The sample repository supports fixed format web log files from the following web servers:

- Netscape
- Apache
- Microsoft IIS, common and extended log formats

The advanced external procedures and external procedures discussed in this chapter read these typical web server logs and reduce the data to a general format based on the common log format. Other web server logs that contain a variable number of fields are not currently supported.

In addition to reading these common formats, the sample repository procedures allow you to parse the log file to obtain the resource and referral fields. This allows you to represent these elements as the appropriate domain name and key value pairs.

Business Component Hierarchy

Inside the Web Logs folder of the sample repository, the sample repository stores instances of sources and mapplets in a business component framework. The business component hierarchy displays instances of mapplets as business components that you can use to build mappings.

A business component is a pointer to a source or mapplet object that you store in the business component framework for easy access. The business components in the sample repository point to mapplets that contain web server log source definitions and transformation logic. The business component hierarchy organizes Web Log business components by the following web server types:

- Apache
- Microsoft IIS
- Netscape

Web Log Repository Objects

The sample repository provides the following repository objects in the WebLog folder:

- Sources
- Targets
- Transformations
- Mapplets
- Mappings

Sources

The following sample sources represent the supported web log formats:

- **IISDefault.** A sample source that represents the default format for the Microsoft IIS web server log.
- **IISLog.** A sample source that represents the extended format for the Microsoft IIS web server log.
- **CLFCkieLog.** A sample source that represents the Netscape and Apache web server logs with cookie support.
- **CLFLog.** A sample source that represents the Netscape and Apache web server logs without cookie support.

Targets

The following sample targets represent the supported web log formats:

- **T_IIS_DEFAULT.** A sample target that represents the default format for the Microsoft IIS web server log.
- **T_IIS_LOG.** A sample target that represents the extended format for the Microsoft IIS web server log.
- **T_CLF_CKIE_LOG.** A sample target that represents the Netscape and Apache web server logs with cookie support.
- **T_CLF_LOG.** A sample target that represents the Netscape and Apache web server logs without cookie support.
- **T_REFERER_DETAILS.** A target used in the sample mappings with the ParseResource external procedures.

Transformations

The following transformations are used in the sample mappings:

- **AEP_KeyValueParse.** An advanced external procedure transformation that parses input URL strings and passes all the “key, value” pairs for each input row to the next transformation.

- **EXP_DateConcat.** An expression transformation that concatenates the two-part date format to a single field.
- **EXT_GetProcID.** An external procedure transformation that returns the ProcID key specified as an initialization parameter.
- **EXT_GetSessionID.** An external procedure transformation that returns the SessionID key specified as an initialization parameter.
- **EXT_GetValByKey.** An external procedure transformation that returns the key value with the URL, key, and terminator specified as input ports. You use this procedure with an expression transformation to evaluate the ports.
- **EXT_ParseResourceF3.** An external procedure transformation that returns the first three levels from the Resource name.
- **EXT_ParseResourceF10.** An external procedure transformation that returns the first ten levels from the Resource name.
- **EXT_ParseResourceL3.** An external procedure transformation that returns the first level plus the last two levels from the Resource name.

Mappings

The sample repository contains the following mappings:

- **m_IIS_CLF and m_IIS_Default.** A sample mapping that reads IIS log files.
- **m_CLFckieStraight.** A sample mapping that demonstrates the Netscape log reading mapplet with cookie support.
- **m_CLFStraight.** A sample mapping that demonstrates the use of the Netscape log reading mapplet without cookie support.
- **m_TestGetVal.** A sample mapping that uses an external procedure with an Expression transformation to evaluate input ports and get values based on more than one key string.
- **m_TestGetVal2.** A sample mapping that demonstrates the use of an external procedure to return values based on certain key strings.
- **m_TestKeyValueParse.** A sample mapping that demonstrates the use of an advanced external procedure to parse input URL strings and write all "key, value" pairs for each input row to a target. This mapping can generate multiple output rows if a single URL has multiple key, value pairs.
- **m_TestParseResourceF3, m_TestParseResourceF10, m_TestParseResourceFL3, and m_TestParseResourceL3.** Sample mappings that demonstrate the use of an external procedure to obtain different values from a single log record.

You can use these prebuilt repository objects or copy them to make changes and build your own. A number of sample source and target definitions are also provided.

Mapplets

The following mapplets implement the EXP_DateConcat transformation to concatenate the date fields from Netscape and Apache web logs, with and without cookie support:

- mplt_CLFCKieLog
- mplt_CLFLog

Other Web Log Components

The sample repository components are based on sample web logs. The following data files are supplied as references:

- iislog.dat
- iisdefault.log
- nscplog.dat
- nscplog2.dat
- nscplogcookie.dat
- testgetval.dat
- testparsef3.dat

Analysis of Web Logs

The sample repository provides the web log reading components to help you get started on developing your own clickstream analysis solution to implement in your data warehouse. The target schemas included with the web log readers are delivered as part of the prebuilt mappings.

Before You Begin

The external procedures supplied for reading web logs assume you are using a supported PowerMart/PowerCenter version and have a network connection to your web server.

For More Information

Additional details about web server logs can be found at the following web sites:

http://www.apache.org

http://home.netscape.com/enterprise/v3.6/index.html

http://technet.microsoft.com/cdonline/Content/Complete/Internet/Server/Sitesrv/Manuals/SSExpGde/SSExpUa/uan4.htm

Reviewing the Sample Web Log Mappings

The transformations in the Web Logs folder provide a starting point for you to capture vital web log data and add these sources to your solution for web log analysis. The sample mappings illustrate how to:

- Concatenate dates from a two-part format to the single field in the standard CLF, CS_SYSDATE.
- Create mappings of the IIS and Netscape/Apache log formats.
- Parse a dynamic URL to obtain the domain name and the different levels.
- Get specific data from a web log file, by specifying a key value and terminator. The transformation uses this input to return relevant indicators to the target table.

This section illustrates the following mappings:

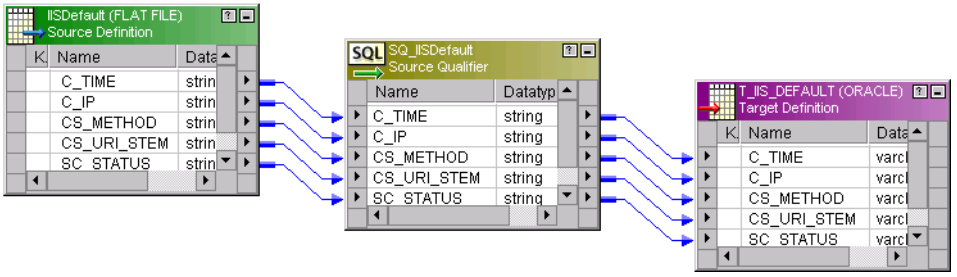
- m_IISDefault
- m_CLFckieStraight
- m_TestGetVal
- m_TestGetVal2
- m_TestParseResourceF3

Other sample mappings are available for review in the sample repository.

m_IIS_Default

This mapping reads the default web log format from Microsoft IIS web server. It maps the web log data to a target schema with no additional data transformation:

Figure 7-1. m_IIS_Default Web Log Mapping

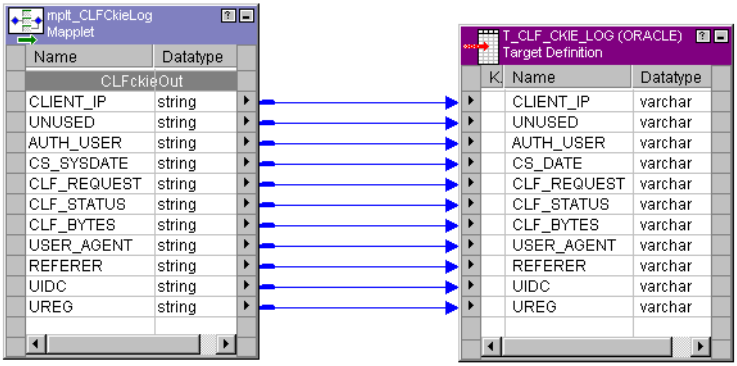


The additional mapping, m_IIS_CLF, performs a similar read for the extended IIS log format.

m_CLFCkieStraight

This mapping reads the default web log format, with cookies, from the Netscape and Apache web servers. It maps the web log data to a target schema and uses a mapplet to concatenate the two-part date format to a single field, in CLF format:

Figure 7-2. m_CLFCkieStraight Web Log Mapping

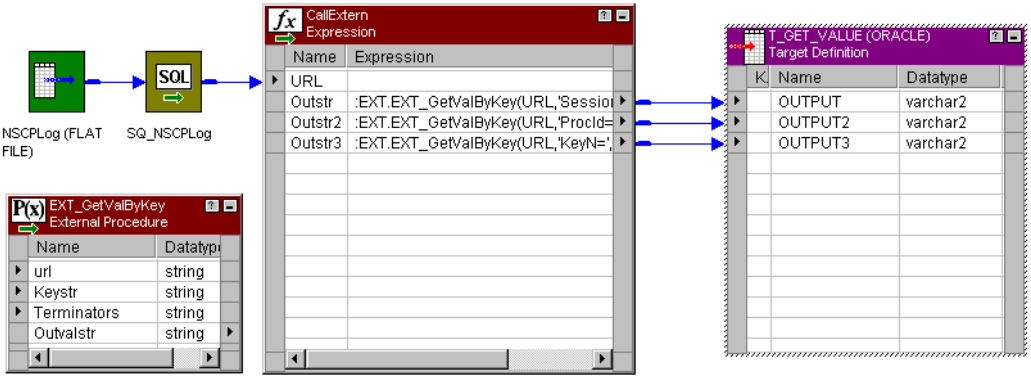


The additional mapping, m_CLFStraight performs a similar read without cookie support.

m_TestGetVal

This mapping uses an unconnected instance of the GetValByKey transformation and a single Expression transformation to get multiple “key,value” pairs from a URL string. The key and delimiter are specified as input ports to get values from the Netscape/Apache web log format and output these values to the target:

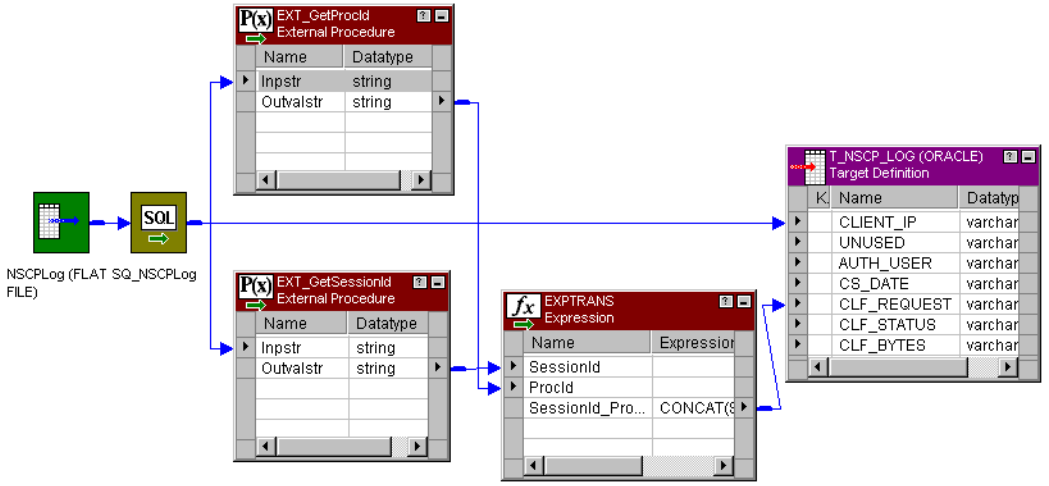
Figure 7-3. m_TestGetVal Web Log Mapping



m_TestGetVal2

This mapping gets values from the Netscape/Apache web log format, based on key strings specified as initialization parameters. It also concatenates SessionID and ProcID to one string for the CLF_REQUEST target field:

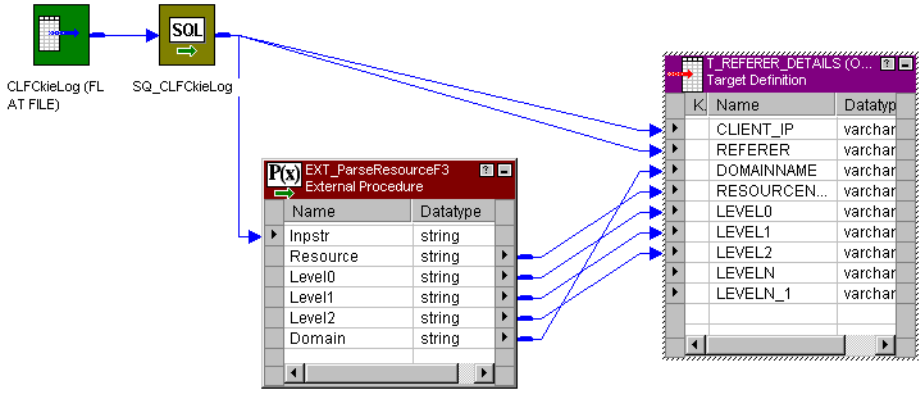
Figure 7-4. m_TestGetVal2 Web Log Mapping



m_TestParseResourceF3

This mapping uses an external procedure to parse the first three levels from the RESOURCE_NAME of a single log record and return both these levels and the domain name:

Figure 7-5. m_TestParseResourceF3 Web Log Mapping



The additional mappings, m_TestParseResourceFL3 and m_TestParseReourceL3, parse other levels.

Understanding Web Log Files

The sample repository provides a predefined, configurable web log schema against which you can analyze supported web server logs. The web log components reduce input from Apache, Netscape, and Microsoft IIS web server logs to a generic format, the Common Log Format (CLF). Both Netscape and Apache maintain their log file in CLF.

PowerMart/PowerCenter reads web log data in the following ways:

- Extracts source data from web servers.
- Parses text strings into the appropriate fields.
- Transforms the log data for needed analysis.
- Maps the log data to a target schema.

The prebuilt sample repository components ease the process of retrieving data from web logs and parse that web log data for you. The current release does not include prebuilt analytical components.

Web Log Conventions

For web logs, the sample repository documentation adopts these terms common to e-business:

- Requester
- Referral
- Dynamic URL
- CLF

Log Formats

The sample repository standardizes on the CLF for all web server logs, transforming data from the IIS log as needed. The Apache, Netscape, and IIS logs all use a fixed format that the web log transformations can read. However, the syntax for a specific access log is defined in the log file header, and your implementation could deviate from the standard. Read the following sections to understand the assumptions made by the sample repository and determine if your web logs follow a log format compatible with the sample web log components.

Common Log Format

The Common Log Format (CLF) is a standard format for web server logging. The Common Log Format (CLF) file contains a separate line for each request. A line is composed of several tokens separated by spaces, for example:

```
host ident authuser date request status bytes
```

If a token does not have a value, then it is represented by a hyphen (-).

Table 7-1 lists the tokens in a CLF log file:

Table 7-1. Common Log Format Tokens

Token	Description
host	Fully-qualified domain name of the client, or its IP address if the name is not available.
ident	If IdentityCheck is enabled and the client machine runs identid, this is the identity information reported by the client.
authuser	If the request was for a password-protected document, then this is the userid used in the request.
date	Date and time of the request, in the following format: - date = [day/month/year:hour:minute:second zone] - day = 2*digit - month = 3*letter - year = 4*digit - hour = 2*digit - minute = 2*digit - second = 2*digit - zone = (`+' | `-') 4*digit
request	Request line from the client, enclosed in double quotes (“).
status	Three-digit status code returned to the client.
bytes	Number of bytes in the object returned to the client, not including any headers.

For example, these lines represent two entries in an access log using the CLF:

```
wiley.a.com - - [16/Feb/1996:21:18:26 -0800] "GET / HTTP/1.0" 200 751

arrow.a.com - john [29/Mar/1996:4:36:53 -0800] "GET /help HTTP/1.0"
401 571
```

The CLF is defined by the string "%h %l %u %t \"%r\" %s %b", which you can use to track other log data by adding extra fields to the end of the CLF. For example, NCSA’s extended log format would be:

```
"%h %l %u %t \"%r\" %s %b \"%{Referer}i\" \"%{User-agent}i\""
 [%...{Foobar}i: The contents of Foobar: header line(s) in the
request sent to the server].
```

Netscape Log Format

Netscape defines the CLF for its web server logs with the following syntax:

```
host - usr [time] "req" s1 c1 s2 c2 b1 b2 h1 h2 h3 h4 xt
```

The fields through *c1* represent the common log format, and the remaining fields represent the extended format.

Table 7-2 lists the Netscape extended format fields:

Table 7-2. Netscape Extended Format Fields

Token	Description
s2	Remote server HTTP response status code to the proxy when the proxy makes a request in part of the client.
c2	Content-length received from the remote server by the proxy.
b1	Size of the client HTTP request message body. (In other words, it is POST-data to be forwarded to the remote server. This data is also passed to the remote server if no error occurs.)
b2	Size of the proxy HTTP request message body. It is the amount of data in the body that was sent to the remote server. (This data is the same as b1 if no error occurs.)
h1	Size of the client HTTP request header to the proxy.
h2	Size of the proxy server response header to the client.
h3	Size of the proxy server request header to the remote server.
h4	Size of the remote server HTTP response header to the proxy.
xt	Total transfer time, in seconds.

For example, the following entry is taken from a Netscape log file:

```
209.27.103.2 - - [20/Jul/1999:12:48:56 -0700]
"GET /informatique99/images/accom.gif HTTP/1.1" 200 232
```

It corresponds to the following header, which emulates the CLF:

```
format=%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%]
"%Req->reqpb.clf-request%" %Req->srvhdrs.clf-status%
%Req->srvhdrs.content-length%
```

SYSDATE Handling

The EXP_DateConcat transformation supplied with the sample repository reads the [%SYSDATE%] token from the log file as two fields, and uses the Informatica transformation language to convert and merge the two fields into one. This is handled using a mapplet that localizes the conversion process to generate a usable flat file for source analysis. As a result:

- The SYSDATE field in the Netscape and Apache web logs cannot contain the hyphen (-) used by other fields to represent a blank.
- The mappings supplied to read web logs assume that the [%SYSDATE%] field is represented as two fields instead of one.

IIS Log Format

Microsoft defines the IIS log format for its web server logs using the syntax illustrated in the following sample entry:

```
1999-08-08 02:07:12 206.24.101.52 - W3SVC1 WWW2 192.168.250.101
GET /informaticapress_register/images/Bullet-blue.gif - 200 0 403
428 47 80 HTTP/1.0 Mozilla/4.04+[en]+(WinNT;+I+;Nav) - http://
www2.informatica.com/informaticapress_register/register_press.html
```

This entry corresponds to the following header:

```
#Fields: date time c-ip cs-username s-sitename s-computername s-ip
cs-method cs-uri-stem cs-uri-query sc-status sc-win32-status sc-bytes
cs-bytes time-taken s-port cs-version cs(User-Agent) cs(Cookie)
cs(Referer)
```

Managing Web Log Data

Using the sample repository prebuilt external procedures, you can accomplish the following web log reading tasks:

- Input the referral field from an HTTP request and return one of the following:
  - The entire resource name.
  - Only the domain name portion of the request.
  - The first three *or* first and last two levels of the resource name.
- Clean up the syntax of a web log to a readable flat file format by converting brackets in the date fields to quotes so the date and time values can be read as a single field.
- Get specified key values from the supported web log files.

For additional capabilities such as parsing other web log strings, you might want to enhance your solution using the Perl and Sorter transformations supplied with the sample repository. For more information, see "Using the Sorter Advanced External Procedure" on page 162 or "Working with the Perl Advanced External Procedure Code" on page 154.

The reusable web log transformations are:

- AEP_KeyValueParse
- EXP_DateConcat
- EXT_GetValByKey
- EXT_GetProcID
- EXT_GetSessionID
- EXT_ParseResourceF3
- EXT_ParseResourceF10
- EXT_ParseResourceL3

Getting Resource and Domain Names

When you design mappings to include web log data, you can use the sample repository to get the resource and domain names from your log file entries, and load this information to the reporting schema you plan to use. This information reveals, for example, the browsing patterns of customers who visit your web site and their reaction to promotional banners. It can also retrieve data about the average session duration and other usage details.

The following transformations are available to parse the different levels from a referral or resource field in a web log from Apache, Netscape, or IIS:

- **EXT_ParseResourceF3.** Returns the resource name and the first three levels from the HTTP request.
- **EXT_ParseResourceL3.** Returns the entire resource plus the first and last two levels.

- **EXT_ParseResourceF10.** Returns the resource name and the first 10 levels from the HTTP request.

You can extend these to return more levels, as needed. These transformations include the domain in its return (.com or .org), and the domain is also one of the output fields.

When these transformations return the resource name, they return the name of any physical files referenced in the input URL.

For example, if the web log contains the following URL in the referral field:

```
http://shop.barnesandnoble.com/booksearch/titles/isbninquiry.asp
?user id=4KJH66HG&isbn=0065735748
```

The resource name is:

```
isbninquiry.asp
```

If you select the option to return the levels of the resource, level 0 is:

```
shop.barnesandnoble.com
```

Level 1 is:

```
booksearch
```

Level 2 is:

```
titles
```

Level N is:

```
titles
```

Level N-1 is:

```
booksearch
```

If the input string does not contain all the levels specified, the transformations return NULL for the empty fields.

Request Field

The module to parse the resource name can handle both request and referral fields from the web log files. For a request field, log files list only the elements after the root directory. For example, if the log entry above were taken from the request field rather than the referral field, it would be:

```
“GET/booksearch/isbninquiry.asp?....”
```

Logic in the module can handle data from both the request and referral fields.

Ports

The following ports are defined for the ParseResource transformations:

Port	Description
Inpstr	Input only. The input string from the web log. Default precision is 1024, but you can change the length.
Resource	Output only. The entire parsed resource name.
Level0	Output only. The first level to return.
Level1	Output only. The second level to return.
Level2	Output only. The third level to return.
Domain	The domain name portion of the resource.

Properties

The following properties are required for the ParseResource transformations:

Property	Value
Type	Specify Informatica as the procedure type.
Module/Program Identifier	Enter the module name of the transformation. Both EXT_ParseResourceF3 and EXT_ParseResourceL3 use the module, *ParseResourceNoRe*. EXT_ParseResourceF10 uses the module, *ParseResource*.
Procedure Name	Identifies the procedure name in the shared library. Procedure names for ParseResourceNoRe are: - ParseResourceF3. Strips the resource name from a dynamic URL and gets the first three levels. - ParseResourceL3. Strips the resource name from a dynamic URL and gets the first level plus the last two levels. Procedure names for ParseResource are: - ParseResourceF10. Strips the resource name from a dynamic URL and gets the first ten levels.
Runtime Location	Identifies the location of the module library. You can use the environment variable, as illustrated, or enter a path.
Tracing Level	Normal is the default value. For more information on setting the Tracing Level property, see “Transformations” in the *Designer Guide*.

Initialization Properties

In the initialization property sheet for your transformation, use the following guidelines:

Property	Value
InitParam1	1=Return the first three levels. 2=Return the first level plus the last two levels.

Getting Key Values

The sample repository provides two transformations for getting key values from a URL string. You can set up mappings to get key values using either one of these transformations:

- **EXT_GetValByKey.** Parses the URL string and return a set of "key,value" pairs for each input row, using key and delimiter input ports. Using this transformation, you can read multiple key/value pairs from the same URL.
- **EXT_GetValByKey2.** An instance of *EXT_GetValByKey* used in a mapping. You can specify a key at initialization time and return the key value.

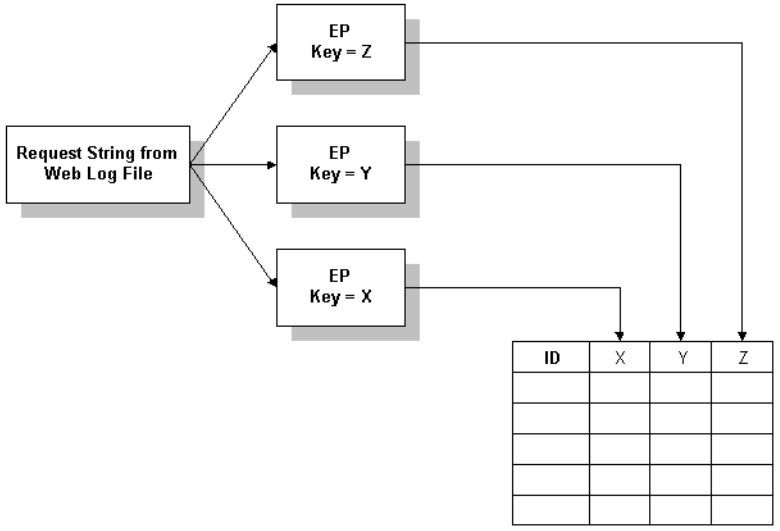
You can copy and edit the transformations to get any key you want and output the key to the desired columns. For example, the following log entry comes from a Netscape web server log:

```
"GET /cgi-bin/hpdirect/hpdirect/shopping/scripts /general/
shopping_basket.jsp?BV_SessionID=362513807.929520572
&BV_EngineID=ealelhedhmibemfcfjfcffcjh.0 HTTP/1.0"
```

Using the EXT_GetValByKey2 transformation, you can, for example, extract the keys "SessionID" and "EngineID" and the associated values, "362513807.929520572" and "ealelhedhmibemfcfjfcffcjh.0" respectively.

Figure 7-6 illustrates the process of extracting key values from a URL string:

Figure 7-6. Getting Web Log Values Using EXT_GetValByKey2 Transformation



Getting Key Value Pairs

With the EXT_GetValByKey transformation, you can parse the query string to separate the dynamic key value pairs and output all of them to a single table. For example, you could parse the following single-line entry to have multiple output columns:

```
http://www.mysite.com/page.asp?OMAction=AddToBasket&RestartFlow=t&
OMLineMerchantID=1&OMProductID=23
```

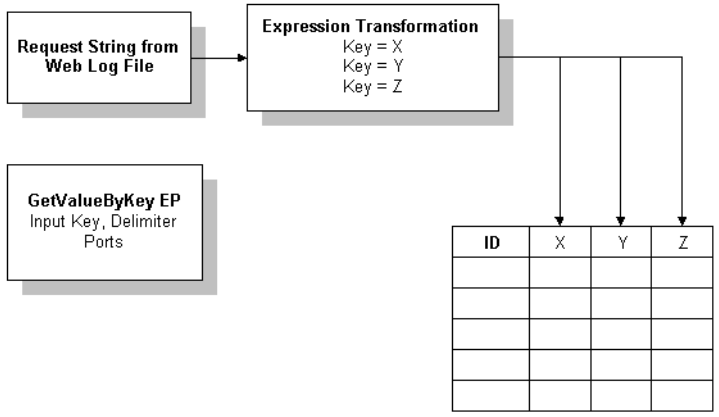
In this case, you can obtain these output columns:

Key	Value
OMAction	AddToBasket
RestartFlow	t
OMLineMerchantID	1
OMProductID	23

You need to use an unconnected instance of EXT_GetValByKey.

Figure 7-7 illustrates the process of separating key value pairs into multiple output columns:

Figure 7-7. Getting Web Log Values Using EXT_GetValByKey Transformation



Properties

Table 7-3 lists the properties for both EXT_GetValByKey and EXT_GetValByKey2 transformations:

Table 7-3. External Procedure EXT_GetValByKey Properties

Property	Value
Type	Specify Informatica as the procedure type.
Module/Program Identifier	Enter the module name of the transformation, GetValEP.
Procedure Name	Identifies the procedure name in the shared library. Procedure names for GetValEP are: - GetValByKey. Returns the "key,value" pairs specified as input ports. - GetValByKey2. Returns the key value specified as an initialization parameter.
Runtime Location	Identifies the location of the module library. You can use the environment variable, as illustrated, or enter a path.
Tracing Level	Normal is the default value. For more information on setting the Tracing Level property, see "Transformations" in the *Designer Guide*.

Ports

The EXT_GetValByKey transformation contains the following ports:

Port	Description
Url	Input only. The input string from the web log. Default precision is 255, but you can change the length.
Keystr	Contains the text string of the key to read and return the value associated with it.
Terminators	Contains the terminator that identifies the end of the entry for the specified key.
Outvalstr	Output only. Contains the specified value for each output row.

The EXT_GetValByKey2 transformation contains the following ports:

Port	Description
Inpstr	Input only. The input string from the web log. Default precision is 255, but you can change the length.
Outvalstr	Output only. Contains the specified value for each output row.

Initialization Properties

EXT_GetValByKey has no initialization properties.

For the EXT_GetValByKey2 transformation, enter the following initialization properties:

Property	Value
Key	Contains the text string of the key to read and return the value associated with it.
ValueTerminators	Contains the terminator that identifies the end of the entry for the specified key.

Performing Log File Source Analysis

For each of the supported log files, the sample repository supplies a sample flat file and prebuilt mappings. These sources are:

- IISDefault
- IISLog
- CLFLog

- CLFckieLog

You can use these source definitions immediately, or copy and add the sample mappings to your own web log mappings.

Creating New Web Log Mappings

The basic steps for adding a web log transformation to a mapping are:

1. Create a mapping where you need to implement one of the prebuilt web log transformations.
2. Make a non-reusable copy of the transformation you want to use (already installed in the sample repository).

   Note: Since the web log transformations are generic transformation objects, be sure to make a copy rather than create an instance of the external procedure. You can then define the ports that you want to use on a mapping-to-mapping basis. This allows you to maintain the same base procedure that all uses of the transformation derive from, adding ports as needed.
3. Drag the ports you need into the transformation object.
4. Specify other attributes of the ports and the transformation properties. See the following section for more information.
5. In the Server Manager, create a session for the mapping.

Running Sessions

When the Informatica Server processes your web log data, it handles the following conditions as indicated:

- **Value not found.** Returns an empty string.
- **Terminator not specified.** Returns the rest of the string as the key value.
- **Empty strings.** Returns an error message in the session log file.
- **No value specified.** Returns an error message in the session log file.
- **Levels longer than 128 characters.** Returns a partial string, as each level has a maximum length of 128 characters.
- **Unexpected syntax in the log file.** Returns unexpected results.

For a list of error messages related to web logs, see “Reviewing the Sample Web Log Mappings” on page 132.

Chapter 8

Calling Perl Functions

This chapter contains the following information:

Overview

The sample repository provides a simple Perl procedure and sample input to illustrate the Perl functionality you can incorporate into your mappings.

The Perl component of the sample repository provides an advanced external procedure that you can use to call any Perl procedure from within a mapping. Using this advanced external procedure, you can easily incorporate the string manipulation and program integration that might be needed to process various types of web log and web server data.

From a Perl procedure called within a sample repository mapping, you can invoke any number of Perl subroutines. For example, you might call a C routine from Perl to invoke the Informatica callback function directly from Perl. You could do this rather than returning values and invoking the callback in the C wrapper. Perl is generally an optimal solution for string manipulation.

You can make copies of the Perl transformation object to use as needed and speed up development of your custom Perl solutions. A sample mapping and sample Perl routine are also supplied with the sample repository.

Perl Repository Objects

In the sample repository, the Perl folder includes the following repository objects:

- **S_logs.** A source definition used in the sample mapping.
- **T_LOGS.** A target definition that holds data from the sample mappings.
- **EXT_INF_Perl.** An external procedure transformation where you specify how to call the Perl subroutine.
- **m_Hex2Char.** A sample mapping that demonstrates string manipulation on web log data.

Perl Sample Files

The following input files are installed to your samples directory:

- pack.pl
- logs

Date Handling with Perl

Although the Perl language does not have native date support, you can add date handling to mappings that use the Perl advanced procedure transformation by using the transformation language functions, IS_DATE and TO_DATE. For more information on these functions, refer to the *Transformation Language Reference.*

Before You Begin

Before you can use the samples described in this chapter or implement a Perl advanced external procedure in your own mapping, you must have Perl installed and configured on the machine hosting the Informatica Server. You can download the Perl program from one of several web sites, depending on your platform.

Perl is widely available as freeware for all supported PowerMart/PowerCenter platforms.

To verify your Perl installation for use with the sample repository, complete these steps:

1. Install a compatible version, or verify that you have installed Perl with the required options.
2. Verify that all environment variables are set in your operating system and in your Perl parameters.

Installing Perl

The following web sites provide downloads of Perl versions that are compatible with the sample repository:

- **www.activestate.com.** Provides a version of Perl recommended for Windows NT/2000.
- **www.cpan.org.** Provides versions of Perl for all UNIX platforms. A Windows NT/2000 version is also available from cpan.org, but the current release of the sample repository supports only the activestate.com version of Perl on Windows NT/2000.

HP and Solaris Configuration

Because HP and Solaris use the dynamically linked library, you must install Perl with the shared library option to run Perl with the sample repository on these platforms. If you did not install Perl with this option, you must reinstall Perl before you can use the sample repository Perl advanced external procedure.

The Perl components of the sample repository do not support Perl with the thread option. If you installed Perl with the thread option, you need to reinstall Perl. If you attempt to run a session with threaded Perl, you might get the following error message in the session log:

```
EP_13008 Cannot load external module.
```

Under UNIX, verify the following environment variables:

1. The libperl library must reside in your path. (On HP, the library is libperl.sl. On Solaris, the library is libperl.so.)
2. The @INC Perl variable must point to the \lib directory of your Perl installation and the \site\lib directory.

To check if the libraries were built shared:

Enter the following command from a command line prompt:

```
perl -MExtUtils::Embed -e ccopts -e ldopts
```

To determine where the shared libraries are located:

Enter the following from a command line prompt:

```
perl -MConfig -e 'print $Config{archlib}'
```

To determine how the @INC variable is set on your machine:

Enter the following from a command line prompt:

```
perl -e 'print@INC'
```

Windows NT Configuration

For Windows NT and AIX, the library is statically linked. On Windows NT, you must also verify the following environment variables:

1. The PERL5LIB environment variable must be set to the \lib directory where Perl is installed. For example, c:\perl\lib. Set this environment variable using the System properties in Control Panel.
2. The @INC Perl variable must also point to the \lib directory of your Perl installation and the \site\lib directory. For example, c:\perl\lib and c:\perl\site\lib.

To determine how the @INC variable is set on your machine:

Issue the following command from a command line prompt:

```
perl -e "print@INC"
```

Note: The double quotes will be single quotes under UNIX.

For More Information

Additional information about the Perl language are available from many different organizations. As good starting points, Informatica recommends the web sites mentioned in "Installing Perl" on page 151.

Reviewing the Perl Sample Mapping

The sample repository provides the m_Hex2Char sample mapping to illustrate how to call Perl. This mapping converts a hexadecimal web log string to ASCII. For example, the following string taken from a web log file contains hexadecimal notation:

```
"GET/cgi-bin/display.html?query=%28Scope%2CAC%27s%29&SortOrder=%5Bi+1%5D"
```

The sample mapping converts this string to:

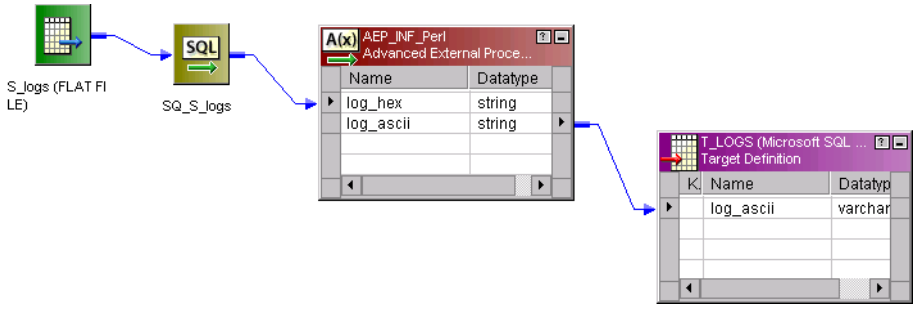
```
"GET/cgi-bin/display.html?query=(Scope,AC's)&SortOrder=[i+1]"
```

m_Hex2Char Sample Mapping

The sample mapping, m_Hex2Char, reads string from the web log, converts the string to ASCII format, and writes to a target. The mapping contains the following:

- **Source definition.** Flat file source definition that reads strings from a web log.
- **Source Qualifier.** Maps the varchar datatypes to strings.
- **Perl advanced external procedure transformation.** Accepts the web log strings as input, and passes them to a named Perl procedure that converts the string to ASCII format.
- **Target definition.** Receives the ASCII value that corresponds to the hexadecimal value in the source.

Figure 8-1. m_Hex2Char Mapping



Working with the Perl Advanced External Procedure Code

The sample repository Perl advanced external procedure accepts an arbitrary number of inputs and outputs and passes them to a named Perl procedure.

The procedure validates input and output types. The Perl routine used in the example below reads row-by-row. Although the advanced external procedure can have a greater number of outputs than inputs, the sample library does not utilize this capability. Using Perl, you can have a variable number of inputs and outputs, but you will always have one row in, one row out.

This section assumes you are familiar with using advanced external procedures with PowerMart/PowerCenter. For complete information on working with advanced external procedures, see "Advanced External Procedure Transformation" in the *Designer Guide*.

Verifying Perl Routines

When a Perl routine is embedded in another program, the path is processed differently and this can sometimes affect the way you call functions. For best results, be sure to verify your Perl routines from the command line before you call them from a mapping.

Note: If the number of return values in the Perl subroutine is greater than the number of output ports defined in the advanced external procedure transformation, the Informatica Server discards the extra return values. For example, if a Perl routine returns three values (A, B, C), and the advanced external procedure transformation has only two output ports defined, the Informatica Server reads the first two values (A, B) and discards the third value (C).

Parsing Perl Output Strings

Output from Perl can be an array or a scaler context. The Perl advanced external procedure supplied with the sample repository reads output from the Perl stack, which could be string or numeric values. If you want to parse Perl output strings, you must use the transformation language. Perl cannot determine if the output is concatenated with a delimiter.

For example, a Perl subroutine might pass back a single string such as:

```
"element 1, element 2, element 3"
```

In this case, the Perl advanced external procedure would pass back one variable corresponding to a single output port. To get three different elements, you must create transformation expressions to divide the string.

Perl Arrays

The sample repository advanced external procedures can accept an array returned from Perl. However, the advanced external procedure interprets an array as a set of elements. Therefore,

if the array has two elements and the advanced external procedure has three output ports, the Informatica Server returns an error.

On the other hand, if the return array contains three elements and the transformation contains two output ports, the Informatica Server passes the first two elements through the output ports and discards the third element.

Handling Dates

To implement date handling with Perl, you need to use the IS_DATE and TO_DATE functions in the transformation language. The Perl language does not support dates and longs.

For more information on handling dates, see the *Transformation Language Reference*.

Calling Perl Subroutines

A single mapping can call more than one Perl subroutine as long as all subroutines are contained in a single Perl file.

Calling Perl Subroutines Contained in Another Perl File

If you need to call a Perl subroutine that is contained in another Perl file, follow these guidelines when writing Perl code:

- Put the called subroutine in a package.
- Have the calling file 'use' that package.
- Locate the package on the PERL5LIB path.
- Invoke the subroutine using standard package::subroutine invocation syntax.

For example, file p2.pm contains a package:

```
package p2;

print;

sub appendstuff
{
   return ("stuff");
}
```

And, the sample Perl script, pack.pl, uses this package:

```
use p2;

while (<>)
{
   $a = hex2char($_);
   print "$a \n";
}

sub hex2char
{
```

```
 my $in = shirt(@_);
 $in =~ s/%(..)/pack “H2”, substr $&, 1, 2/eg;
 $in = $in . p2::appendstuff();
 return $in;
}
```

When executed from the command line or from a session, this appends ‘stuff’ to every output string.

Invoking Perl from the Advanced External Procedure

The basic steps for adding the Perl advanced external procedure to a mapping are:

1. Create a mapping where you need to implement the Perl advanced external procedure.
2. Make a non-reusable copy of the Perl advanced external procedure (already installed in the sample repository).

   Note: Since the Perl advanced external procedure is a generic transformation object, be sure to make a copy rather than create an instance of the advanced external procedure. You can then define the initialization parameters you want to use for each mapping. This allows you to maintain the same base procedure.
3. Specify other attributes of the ports and the Perl properties. See the following section for more information.
4. In the Server Manager, create a session for the mapping.

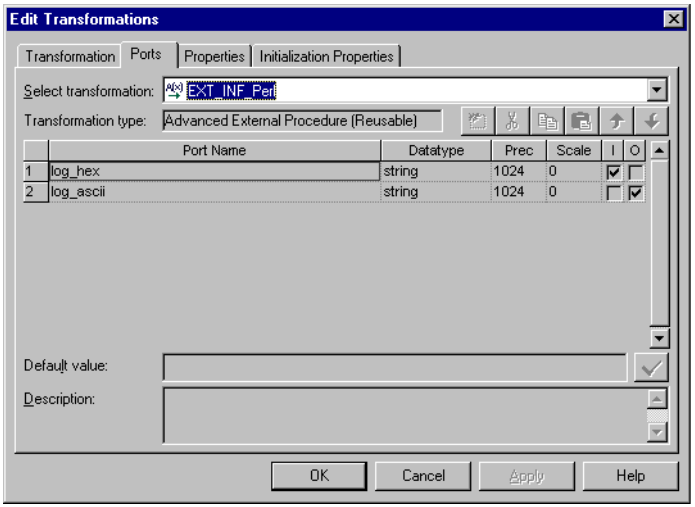
Perl Ports

The sample Perl advanced external procedure transformation contains the following ports:

- log_hex
- log_ascii

Figure 8-2 shows the transformation ports of the Perl advanced external procedure:

Figure 8-2. Perl Transformation Ports



When you define ports for Perl transformations used in your mappings, remember these guidelines:

- In the current release, you can only define one row in, one row out.
- Only numeric and character datatypes are supported. Raw datatypes are not supported. For advanced external procedures in general, the decimal datatype is not supported. You can use a double datatype instead.

Perl Properties

When you use the Perl advanced external procedure transformation in a mapping, you need to configure the properties sheet of the Perl advanced external procedure transformation.

Figure 8-1 displays the values you need to enter in the properties sheet of the Perl advanced external procedure transformation:

Table 8-1. Perl Advanced External Procedure Transformation Properties

Property	Value
Type	Specify Informatica as the procedure type.
Module/Program Identifier	Identifies the module name of the advanced external procedure, INF_Perl. This property is constant for every invocation of Perl.
Procedure Name	Identifies the procedure name, perlExecute, in the shared library. This property is constant for every invocation of Perl.
Runtime Location	Identifies the location of the advanced external procedure library. You can use the environment variable, as illustrated, or enter a path.

Table 8-1. Perl Advanced External Procedure Transformation Properties

Property	Value
Tracing Level	Normal is the default value. For more information on setting the Tracing Level property, see "Transformations" in the *Designer Guide*.
Is Partitionable	Do not select this check box. Specifies whether or not to allow partitioning. The Perl Advanced External Procedure transformation does not support partitioning. By default, this property is not selected. For more information on partitioning data, see "Partitioning Data" in the *Session and Server Guide*.
Is Active	Specifies the transformation to remain an active transformation. By default, the Advanced External Procedure transformation is an active transformation. For more information about active and passive transformations, see "Transformations" in the *Designer Guide*.

Perl Initialization Properties

Configure the initialization properties of the Perl advanced external procedure transformation.

Table 8-2 lists the Perl advanced external procedure initialization properties:

Table 8-2. Perl Advanced External Procedure Transformation Initialization Properties

Property	Value
filename	File containing the Perl subroutine. The procedure entered here is loaded and runs at execute time. It can call other procedures within it.
directory	Directory where the Perl module resides.
subroutine	Name of the subroutine inside the Perl file. This is invoked at runtime.

Chapter 9

Sorting Data with Web Sorter Transformation

This chapter contains the following information:

Overview

The Sorter advanced external procedure transformation provided with the sample repository allows sorting of data sources that are not capable of returning sorted data. Web log files, for example, do not return sorted data. The Sorter advanced external procedure transformation is an enhanced sorting utility you can use in your mappings to perform this task.

This sorter is introduced as an advanced external procedure so you can use it in mappings with other transformations that can read incoming web data. It is available on all platforms that support the sample repository.

You can use this advanced external procedure to perform the following tasks:

- Sort a single data set based on either a numeric, date, or character string key.
- Handle an arbitrarily large data set.
- Reuse the same sorting logic for any set of input rows.
- Sort only in memory, or employ disk caching for large data sets.

You can have multiple sorters active at one time in a given data pipeline.

Sorter Repository Objects

In the sample repository, the Sorter folder includes the following repository objects:

- **URLs.** A sample source containing a list of IP addresses.
- **T_SORTEDIPS.** A sample target table that holds one row for each address.
- **AEP_cachedIPSorter.** An advanced external procedure transformation that calls the sample Perl procedure.
- **m_CachedSorter.** A sample mapping that implements the Sorter advanced external procedure against the data source.

Working with the Sorter Sample Mapping

The sample repository provides a sample mapping that uses the advanced external procedure transformation, AEP_cachedIPSorter. The advanced external procedure transformation is copied to your shared library directory during installation. This section discusses the mappings and the advanced external procedure, and assumes you are already familiar with how to develop advanced external procedures and mappings. For more information on using mappings, see "Advanced External Procedure Transformation" in the *Designer Guide*.

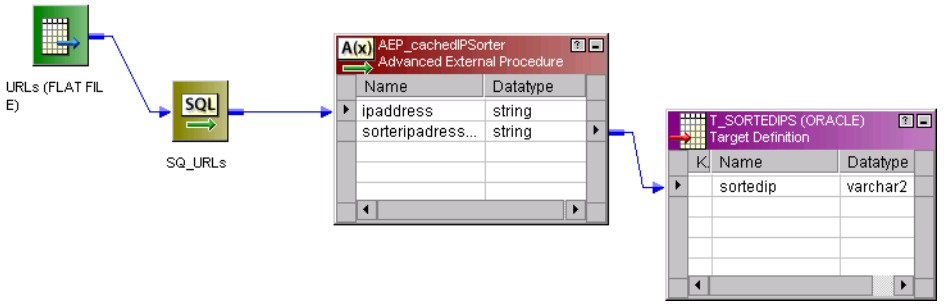
Sample Mapping

The sample mapping, m_CachedSorter, uses the following transformations:

- **Source.** References a flat file source containing a list of IP addresses.
- **Source Qualifier.** Passes the source data through to the advanced external procedure.
- **AEP_cachedIPSorter Advanced External Procedure.** Contains the logic to sort source data.
- **Target.** Identifies target columns in a sample table.

Figure 9-1 displays the m_CachedSorter mapping with the AEP_cachedIPSorter transformation:

Figure 9-1. m_CachedSorter Mapping



Using the Sorter Advanced External Procedure

The Sorter advanced external procedure accepts a set of input ports ordered through the sort keys that contain all of the additional rows to be used as output. To allocate memory, the procedure uses metadata that describes the rows and the initialization parameters set when you install the advanced external procedure.

Null Handling

Nulls are sorted as the highest possible value for the given column.

Adding the Sorter to a Mapping

The basic steps for adding the Sorter advanced external procedure to a mapping are:

1. Create a mapping where you need to implement the Sorter.
2. Make a non-reusable copy of the Sorter advanced external procedure (already installed in the sample repository).

   Note: Since the Sorter is a generic transformation object, be sure to make a copy rather than create an instance of the advanced external procedure. You can then define the ports that you want to sort on a mapping-to-mapping basis. This allows you to maintain the same base procedure that all uses of the sorter derive from, adding ports as needed.
3. Drag the ports on which you want to perform sorting into the transformation object.

   By default, the Sorter is active. Therefore, all ports you want to use after the sorting need to pass through the Sorter, whether or not you actually sort on these ports.
4. Specify other attributes of the ports and the sort properties. See the following section for more information.
5. In the Server Manager, create a session for the mapping.

   Note: In the current release, you can only use the Sorter advanced external procedure to sort in binary order. When you configure the session in the Server Manager, select binary as the sort order in the session properties Transformation tab.

Defining Ports

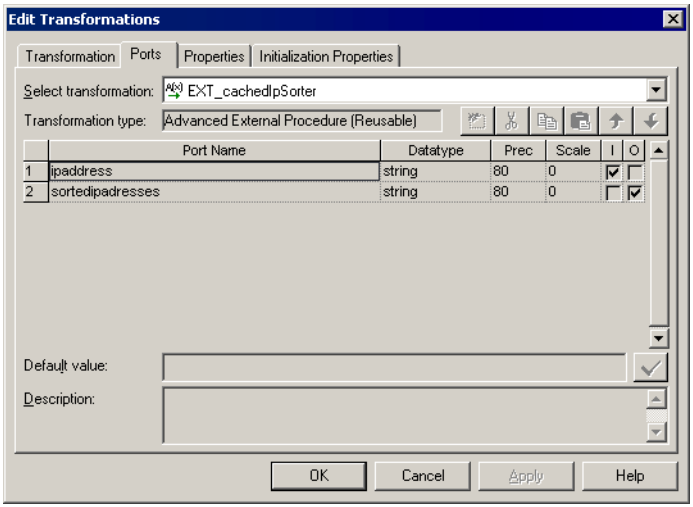
The sample Sorter advanced external procedure transformation contains the following ports:

- ipaddress
- sortedipaddresses

The ports you implement depend on the source data you want to sort.

Figure 9-2 displays the sample Sorter advanced external procedure transformation ports:

Figure 9-2. Sample Sorter Transformation Ports



Guidelines for Sorter Ports

When you define ports for the Sorter transformation used in your mappings, use these guidelines:

- The order of ports is important if you are sorting ports by a value of *n*, where *n* is the number of significant ports in the sortby property (or 1 if the property is not set).
- Ports you want to use after sorting must pass through the Sorter transformation, whether or not you sort with these ports.
- For advanced external procedures in general, the decimal datatype is not supported. You can use a double datatype instead.

Setting Sorter Properties

When you use the Sorter advanced external procedure transformation in a mapping, you need to enter certain values in the properties tab of the Sorter advanced external procedure transformation.

Table 9-1 lists the required properties for the Sorter advanced external procedure transformation:

Table 9-1. Sorter Advanced External Procedure Transformation Properties

Property	Value
Type	Specify Informatica as the procedure type.
Module/Program Identifier	Enter the module name of the Sorter, INF_Sort.

Table 9-1. Sorter Advanced External Procedure Transformation Properties

Property	Value
Procedure Name	Identifies the procedure name, Sort, in the shared library.
Runtime Location	Identifies the location of the advanced external procedure library. You can use the environment variable, as illustrated, or enter a path.
Tracing Level	Normal is the default value. For more information on setting the Tracing Level property, see "Transformations" in the *Designer Guide*.
Is Partitionable	Do not select this check box. Specifies whether or not to allow partitioning. The Sorter Advanced External Procedure transformation does not support partitioning. By default, this property is not selected. For more information on partitioning data, see "Partitioning Data" in the *Session and Server Guide*.
Is Active	Specifies the transformation to remain an active transformation. By default, the Advanced External Procedure transformation is an active transformation. For more information about active and passive transformations, see "Transformations" in the *Designer Guide*.

Specifying Initialization Properties

You also need to enter values in the initialization properties sheet of the Sorter advanced external procedure transformation.

Enter the following values in the initialization properties sheet of the Sorter advanced external procedure transformation:

Table 9-2. Sorter Advanced External Procedure Transformation Initialization Properties

Property	Value
sortby	1 to the maximum number of ports on the transformation that identifies the number of columns to be sorted.
sortorder	Ascending or Descending. Accepts 'a' or 'd'. If you specify an invalid value, the Informatica Server uses ascending order.
buffersize	The number of rows per buffer. 100 is the minimum value. The default buffersize is 1,000. Note: The Informatica Server allocates memory based on the buffersize specified in the advanced external procedure initialization properties. To ensure high performance for the Informatica Server, set the buffersize to the value you need to complete the session. Avoid setting to buffersize to a large number (such as 100,000). Large buffersize values slow session run times. The recommended buffersize range is 500 to 1,000.
cachetodisk	Enter Yes or No. If you do not cache to disk, the Sorter checks for memory allocation before running a session.

Appendix A

Naming Conventions

This appendix provides suggested naming conventions for PowerMart and PowerCenter repository objects.

Suggested Naming Conventions

The following naming conventions appear throughout the Informatica documentation and client tools. Informatica recommends using the following naming convention when you design mappings and create sessions.

Transformations

Table A-1 lists the naming convention you should use for all transformations:

Table A-1. Naming Conventions for Transformations

Transformation	Naming Convention
Advanced External Procedure	AEP_*TransformationName*
Aggregator	AGG_*TransformationName*
ERP Source Qualifier	ESQ_*TransformationName*
Expression	EXP_*TransformationName*
External Procedure	EXT_*TransformationName*
Filter	FIL_*TransformationName*
Joiner	JNR_*TransformationName*
Lookup	LKP_*TransformationName*
MQ Source Qualifier	SQ_MQ_*TransformationName*
Normalizer	NRM_*TransformationName*
Rank	RNK_*TransformationName*
Router	RTR_*TransformationName*
Sequence Generator	SEQ_*TransformationName*
Stored Procedure	SP_*TransformationName*
Source Qualifier	SQ_*TransformationName*
Update Strategy	UPD_*TransformationName*
XML Source Qualifier	XSQ_*TransformationName*

Targets

The naming convention for targets is: T_*TargetName*.

Mappings

The naming convention for mappings is: m_*MappingName*.

Mapplets

The naming convention for mapplets is: mplt_*MappletName*.

Sessions

The naming convention for sessions is: s_*MappingName*.

Sequential Batches

The naming convention for sequential batches is: bs_*SequentialBatchName*.

Concurrent Batches

The naming convention for concurrent batches is: bc_*ConcurrentBatchName*.

Index

F

G

I

L

M

N

O

P

R

S

T

U

W