



**Cognizant**  
Passion for making a difference



## Handout: Informatica

Version: INFORMATICA8X/Handout/2009/1.0

Date: 03-04-09

Cognizant

500 Glen Pointe Center West

Teaneck, NJ 07666

Ph: 201-801-0233

[www.cognizant.com](http://www.cognizant.com)

## TABLE OF CONTENTS

Introduction.....	6
About this Module.....	6
Target Audience .....	6
Module Objectives .....	6
Pre-requisite .....	6
<b>Session 1: ETL &amp; Informatica .....</b>	<b>7</b>
Learning Objectives.....	7
ETL Overview .....	7
ETL Terms.....	7
Extraction .....	8
Transformation .....	9
Loading.....	9
ETL Process Flow .....	10
Detailed ETL Process Flow .....	11
Popular ETL Tools.....	11
Informatica: An Overview.....	12
Informatica Architecture and Components.....	12
Informatica Services Connectivity.....	14
New Features .....	14
Summary .....	16
Test Your Understanding.....	16
<b>Session 2: Integration Service .....</b>	<b>17</b>
Learning Objectives.....	17
Integration Service Overview.....	17
Integration Service Process.....	18
Load Balancer .....	18
Data Transformation Manager (DTM) Process .....	19
Integration Service and the Administration Console .....	19
Summary .....	24
Test your Understanding .....	24
Learning Objectives.....	25
PowerCenter Repository.....	25
Administration Console.....	25
Creating Repository Service .....	26
Creating Repository Content .....	28
Deleting Repository Content.....	29

Backing up a Repository .....	29
Restoring a Repository .....	30
Licensing .....	32
Summary .....	33
Test your Understanding .....	33
Learning Objectives.....	34
Repository Manager Overview .....	34
Understanding the Repository Manager Windows.....	35
Connecting to a Repository .....	38
Managing Groups .....	39
Managing Users .....	40
Managing Privileges .....	43
Managing Folders.....	43
Viewing User Connections.....	48
Viewing Locks .....	49
Viewing Dependencies .....	49
Try It Out .....	51
Summary .....	51
Test your Understanding .....	51
<b>Session 5: Mapping Designer.....</b>	<b>52</b>
Learning Objectives.....	52
Designer Overview .....	52
Source Analyzer .....	53
Warehouse Designer.....	56
Transformation Developer .....	57
Source Qualifier Transformation.....	58
Expression Transformation.....	59
Filter Transformation .....	60
Joiner Transformation.....	62
Rank .....	65
Router Transformation.....	66
Normalizer Transformation .....	68
Look Up Transformation .....	69
Sorter Transformation.....	71
Aggregator Transformation.....	72
Sequence Generator Transformation .....	75
Update Strategy Transformation.....	76
Stored Procedure .....	79
Custom Transformation .....	81

XML Transformation .....	82
Mapplet Designer .....	82
Mapping Designer .....	83
Mapping Parameters and Variables.....	84
Parameter Files .....	87
Try It Out .....	90
Summary .....	90
Test your Understanding .....	90
 <b>Session 6: Workflow Manager .....</b>	 <b>91</b>
Learning Objectives.....	91
Manager Overview .....	91
Source and Target Connection.....	91
Tasks.....	94
Sessions.....	102
Workflows.....	106
Try It Out .....	108
Summary .....	108
Test your Understanding .....	108
 <b>Session 7: Workflow Monitor .....</b>	 <b>109</b>
Learning Objectives.....	109
Monitor Overview.....	109
Workflow Monitor Views .....	110
Running and Monitoring Workflows .....	111
Scheduling Workflows .....	112
Workflow and Session Logs .....	112
Monitor Configuration .....	113
Try It Out .....	114
Summary .....	114
Test your Understanding .....	114
 <b>Session 8: Reusable Objects.....</b>	 <b>115</b>
Learning Objectives.....	115
Reusable Transformations.....	115
Mapplets.....	118
Worklets .....	123
Reusable Workflow Tasks .....	126
Try It Out .....	128
Summary .....	128

Test your Understanding .....	128
<b>Session 9: Advanced Topics .....</b>	<b>129</b>
Learning Objectives .....	129
Versioning .....	129
Deployment Groups.....	132
Exporting and Importing Objects .....	133
Copying Objects .....	140
Debugger.....	142
Data Recovery.....	143
Partitioning .....	144
Performance Data Collection.....	145
Bottlenecks.....	145
Summary .....	147
Test your Understanding .....	147
<b>References .....</b>	<b>148</b>
Websites.....	148
Books .....	148
<b>STUDENT NOTES: .....</b>	<b>149</b>

## Introduction

### About this Module

---

This handout document

- ❑ Introduces the PowerCenter Tools
- ❑ Explains the basic concepts in ETL and Informatica
- ❑ Touches the Advanced topics in Informatica

### Target Audience

---

Entry Level Trainees

### Module Objectives

---

After completing this module, you will be able to:

- ❑ Explain the ETL process in detail
- ❑ Describe the basic concepts in Integration and Repository Service
- ❑ Work with Informatica Repository Manager
- ❑ Explain all the Designer Tools
- ❑ Describe the transformations in Informatica
- ❑ Work with Workflow Manager
- ❑ Track all the Workflows in the Workflow Monitor
- ❑ Work with Reusable Objects

### Pre-requisite

---

The trainee needs to have basic knowledge in RDBMS and Data Warehousing Concepts

## Session 1: ETL & Informatica

### Learning Objectives

---

- ☐ ETL Overview
- ☐ ETL Terms
- ☐ Extraction, Transformation, Loading
- ☐ ETL Process Flow
- ☐ Detailed ETL Process Flow
- ☐ Popular ETL Tools
- ☐ Informatica Overview
- ☐ Informatica Architecture and Components
- ☐ Informatica Services Connectivity
- ☐ New Features

### ETL Overview

---

ETL stands for Extraction, Transformation and Loading. ETL is a process that involves the following tasks:

- **Extracting** data from source operational or archive systems which are the primary source of data for the data warehouse
- **Transforming** the data - which may involve cleaning, filtering, validating and applying business rules
- **Loading** the data into a data warehouse or any other database or application that houses data

### ETL Terms

---

#### Source System

Source System is a database, application, file, or other storage facility from which the data in a data warehouse is derived. Some of them are Flat files, Oracle Tables, Microsoft SQL server tables, COBOL Sources, XML files.

#### Mapping

Mapping is the definition of the relationship and data flow between source and target objects. It is a pictorial representation about the flow of data from source to target.

#### Metadata

Metadata describes data and other structures, such as objects, business rules, and processes. For example, the schema design of a data warehouse is typically stored in a repository as metadata, which is used to generate scripts used to build and populate the data warehouse. Metadata contains all the information about the source tables, target tables, the transformations, so that it will be useful and easy to perform transformations during the ETL process. A repository contains metadata.

#### Staging Area

Staging area is place where you hold temporary tables on data warehouse server. Staging tables are connected to work area or fact tables. We basically need staging area to hold the data, and perform data cleansing and merging, before loading the data into warehouse.

## Data Cleansing

It is the process of resolving inconsistencies and fixing the anomalies in source data, typically as part of the ETL process. The data cleansing technology improves data quality by validating, correctly naming and standardizing data. For example, a person's address may not be same in all source systems because of typos and postal code, city name may not match with address. These errors can be corrected by using data cleansing process and standardized data can be loaded in target systems (data warehouse).

## Transformation

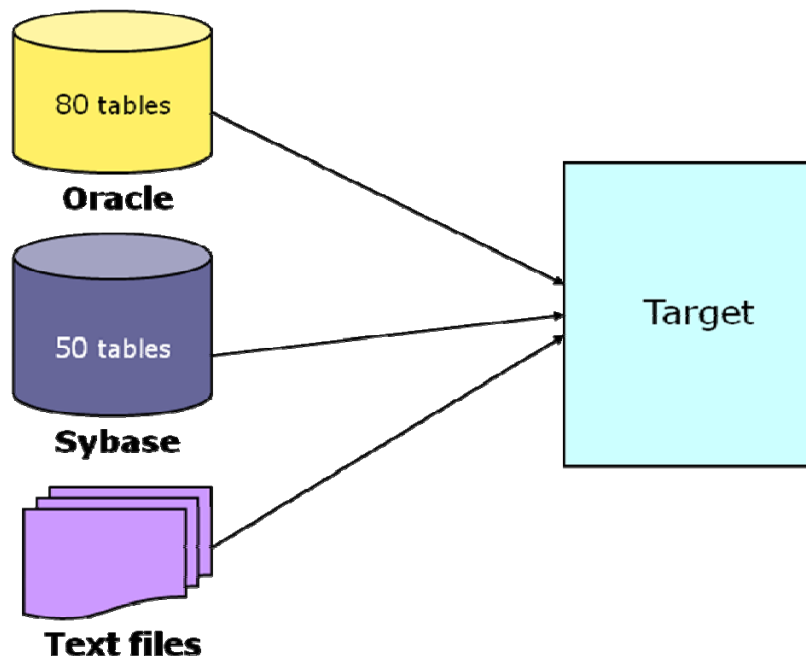
Transformation is the process of manipulating data. Any manipulation beyond copying is a transformation. Examples include cleansing, aggregating, and integrating data from multiple sources. A transformation is a repository object that generates, modifies, or passes data.

## Target System

Target System is a database, application, file, or other storage facility to which the "transformed source data" is loaded in a data warehouse.

## Extraction

The first part of an ETL process involves extracting the data from the source systems. Most data warehousing projects consolidate data from different source systems. Each separate system may also use a different data organization / format. Common data source formats are relational databases and flat files, but may include non-relational database structures such as Information Management System (IMS) or other data structures such as Virtual Storage Access Method (VSAM) or Indexed Sequential Access Method (ISAM), or even fetching from outside sources such as web spidering or screen-scraping. Extraction converts the data into a format for transformation processing.



An intrinsic part of the extraction involves the parsing of extracted data, resulting in a check if the data meets an expected pattern or structure. If not, the data may be rejected entirely.

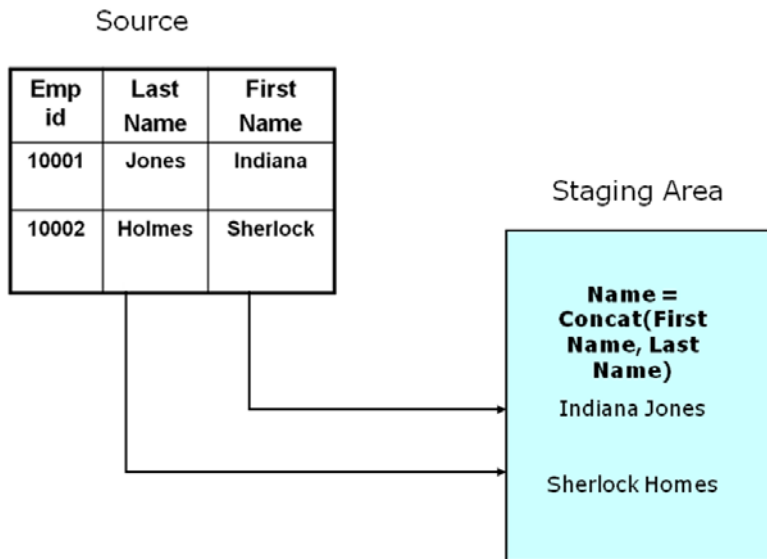


## Transformation

The transform stage applies to a series of rules or functions to the extracted data from the source to derive the data for loading into the end target. Some data sources will require very little or even no manipulation of data. In other cases, one or more of the following transformations types to meet the business and technical needs of the end target may be required:

- Selecting only certain columns to load (or selecting null columns not to load)
- Translating coded values and automated data cleansing
- Encoding free-form values
- Deriving a new calculated value
- Filtering
- Sorting
- Joining data from multiple sources
- Aggregation
- Generating surrogate-key values
- Transposing or pivoting columns
- Splitting a column into multiple columns

Applying any form of simple or complex data validation. If validation fails, it may result in a full, partial or no rejection of the data, and thus none, some or all the data is handed over to the next step, depending on the rule design and exception handling. Many of the above transformations may result in exceptions, for example, when a code translation parses an unknown code in the extracted data.

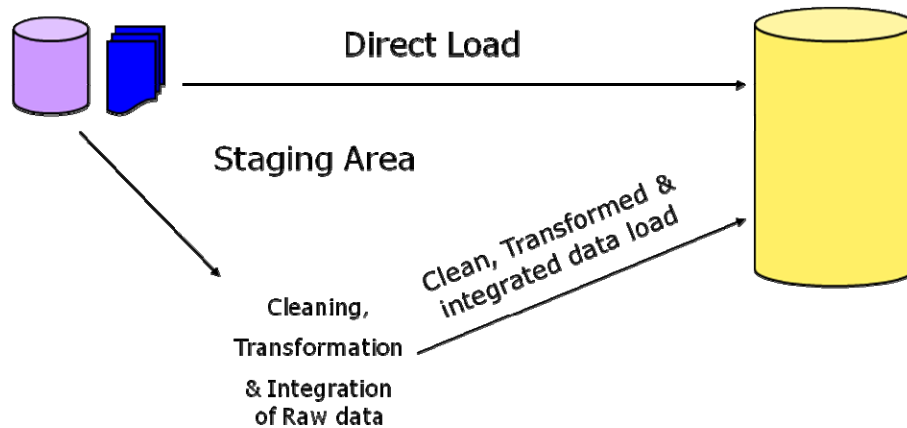


In Source, the data available is First name and Last name. To get the full name, the transformation logic is applied by concatenating both the first and last name. The place where these transformations take place is called the Staging Area.

## Loading

In loading stage, some data are loaded to the target directly without applying any transformation logic and some data are loaded to the target after applying the logic or business rules. The load phase loads the data into the end target, usually the Data Warehouse (DW). Depending on the requirements of the organization, this process varies widely. Some data warehouses may overwrite existing information with cumulative, updated data every week, while other DW (or even other parts of the same DW) may add new data in a historized form, for example, hourly. The timing and scope to replace or append are strategic design choices dependent on the time available and the business needs. More complex systems maintain a history and audit trail of all changes to the data loaded in the DW.

As the load phase interacts with a database, the constraints defined in the database schema, as well as in triggers activated upon data load apply (for example, uniqueness, referential integrity, mandatory fields), which also contribute to the overall data quality performance of the ETL process.

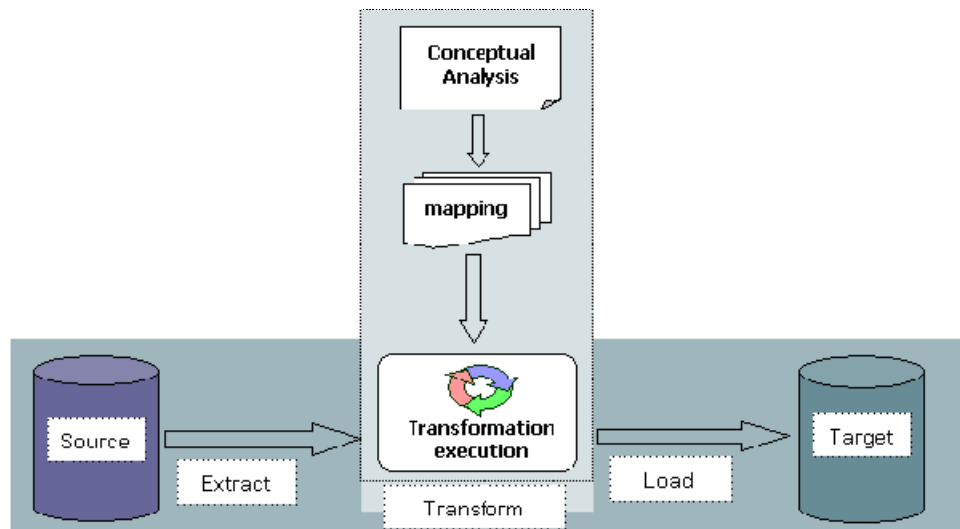


## ETL Process Flow

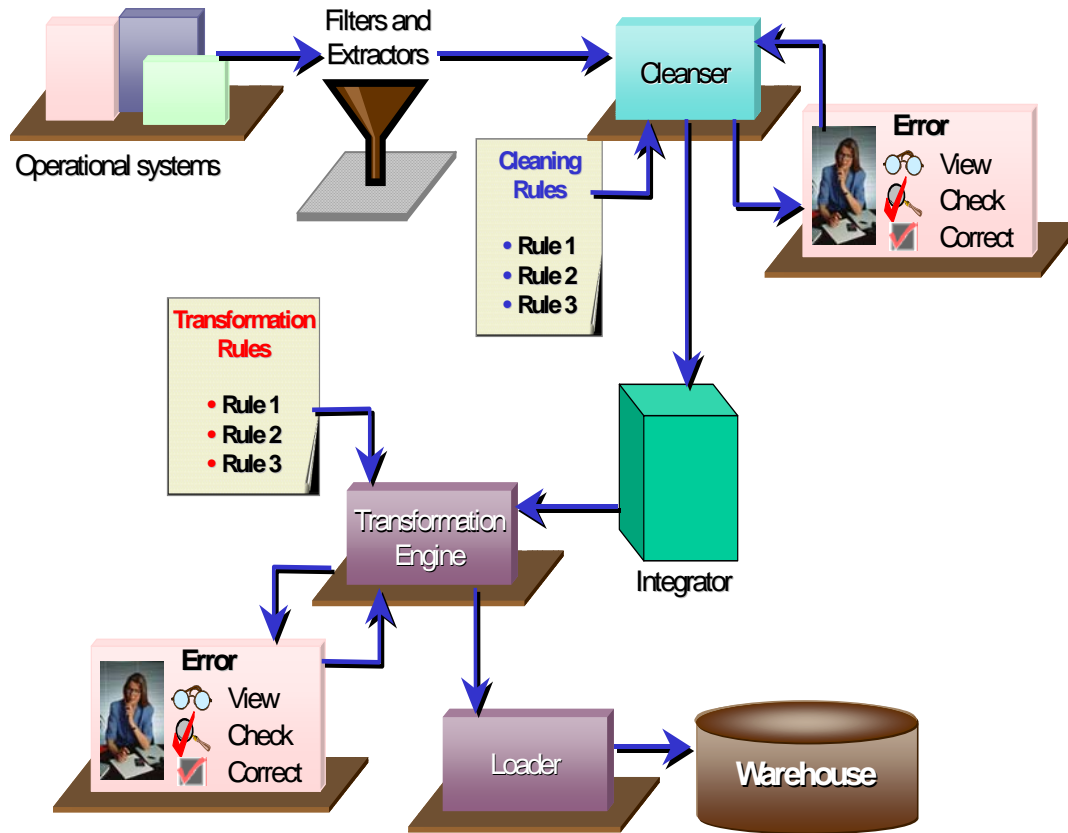
During Extraction, the desired data is identified and extracted from many different sources, including database systems and applications. Very often, it is not possible to identify the specific subset of interest; therefore more data than necessary has to be extracted, so the identification of the relevant data will be done at a later point in time. After extracting data, it has to be physically transported to an intermediate system for further processing.

Depending on the chosen way of transportation, some transformations can be done during this process, too. For example, a SQL statement which directly accesses a remote target through a gateway can concatenate two columns as part of the SELECT statement. Based on the requirements, some transformations may take place during the Transformation and Execution Phase. Through Informatica mappings, the necessary changes and updates of the data are made using transformations.

Then in the Load phase the data is loaded in the target. After all the transformations, it has to be physically transported to the target system for loading the data into the Target.



## Detailed ETL Process Flow



## Popular ETL Tools

Tool Name	Company Name
Informatica	Informatica Corporation
DT/Studio	Embarcadero Technologies
Data Stage	IBM
Ab Initio	Ab Initio Software Corporation
Data Junction	Pervasive Software
Oracle Warehouse Builder	Oracle Corporation
Microsoft SQL Server Integration	Microsoft
TransformOnDemand	Solonde
Transformation Manager	ETL Solutions

## Informatica: An Overview

Informatica provides a single enterprise data integration platform to help organizations access, transform, and integrate data from a large variety of systems and deliver that information to other transactional systems, real-time business processes, and users.

Informatica supports the activities of Business Integration Competency Center (ICC) and other integration experts by serving as the foundation for data warehousing, data migration, consolidation, "single-view," metadata management and synchronization.

### Informatica: Key Benefits

With Informatica you can

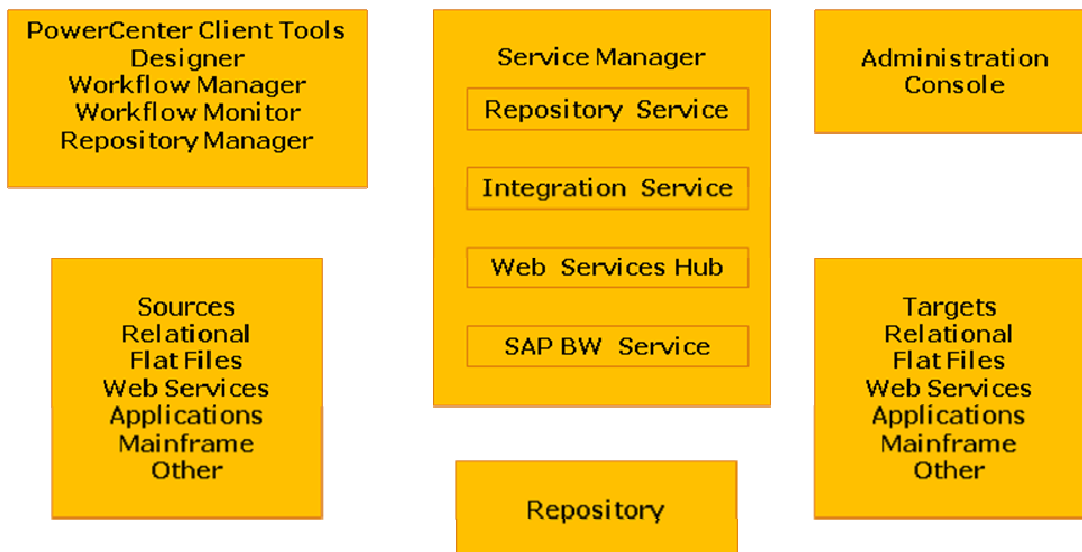
- Integrate data to provide business users holistic access to enterprise data - data is comprehensive, accurate, and timely.
- Scale and respond to business needs for information - deliver data in a secure, scalable environment that provides immediate data access to all disparate sources.
- Simplify design, collaboration, and re-use to reduce developers' time to results - unique metadata management helps boost efficiency to meet changing market demands.

### Informatica: Key Features

- **Enterprise-level data integration.** Informatica ensures accuracy of data through a single environment for transforming, profiling, integrating, cleansing, and reconciling data and managing metadata.
- **Security.** Informatica ensures security through complete user authentication, granular privacy management, and secure transport of your data.
- **Visual Interface.** Informatica is an easy to use tool with a simple visual interface for designing the integration.
- **Developer productivity.** Informatica simplifies design processes by making it easy to search and profile data, reuse objects across teams and projects, and leverage metadata.
- **Compatibility.** Informatica can communicate with a wide range of data sources and move huge volume of data between them effectively.

## Informatica Architecture and Components

When a workflow starts, the Informatica Server retrieves mapping, workflow, and session metadata from the repository to extract data from the source, transform it, and load it into the target. It also runs the tasks in the workflow. The Informatica Server uses Load Manager and Data Transformation Manager (DTM) processes to run the workflow.



## Server Components

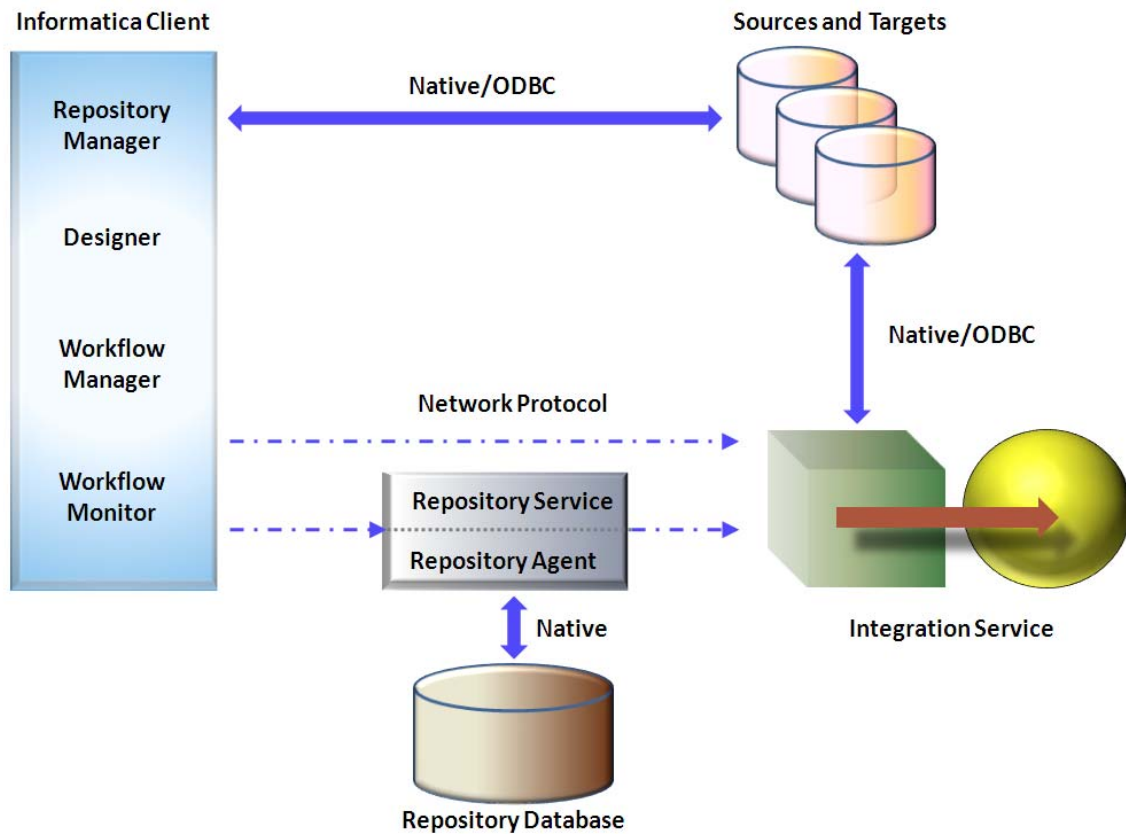
- **Integration Service.** Integration Service manages the scheduling and execution of workflows. It reads mapping and session information from the repository, extracts data from the mapping sources and stores the data in memory while it applies the transformation rules that you configure in the mapping. It loads the transformed data into the mapping targets. You can achieve high performance using symmetric multi-processing systems, start and run multiple workflows concurrently and concurrently process partitions within a single session.
- **Repository Service.** Repository Service manages the metadata in the repository database and repository connection requests from client applications. For each repository database registered with the Repository Server, it configures and manages a Repository Agent process. It monitors the status of running Repository Agents, and sends repository object notification messages to client applications. The Repository Agent is a separate, multi-threaded process that retrieves, inserts, and updates metadata in the repository database tables. The Repository Agent ensures the consistency of metadata in the repository by employing object locking.
- **Web Services Hub.** The Web Services Hub is a gateway that exposes PowerCenter functionality to external clients. It receives requests from web service clients and passes them to the Integration Service or Repository Service. The Integration Service or Repository Service processes the requests and sends a response to the web service client through the Web Services Hub.
- **SAP BW Services.** The SAP BW Service listens for RFC requests from SAP BW and initiates workflows to extract from or load to SAP BW. The SAP BW Service is not highly available. You can configure it to run on one node.

## Client Components

- **Repository Server Administration Console.** Repository Server Administration Console is used to add, edit, and remove repository configurations. It is also used to export and import repository configurations, create, promote, copy, delete, backup and restore repositories.
- **Repository Manager.** Repository Manager is used for administering the repository, perform folder functions like create, edit and delete folders. It is also used to copy a folder within a repository or to other repository, compare folders, and implement repository security. Also, it allows us to create, edit and delete repository users and user groups, assign and revoke repository privileges and folder permissions, view locks and un-lock objects. You can import and export repository connection information in the registry, analyze source/target, mapping dependencies and view the properties of repository objects.
- **Designer.** Designer provides five tools with which you can create mappings. Source Analyzer is used to import or create source definitions for flat file, ERP, and relational sources. Warehouse Designer is used to import or create target definitions. Transformation developer is used to create reusable transformations. Maplet designer is used to create maplets. Mapping designer is used to create mappings.
- **Workflow Manager.** Workflow Manager is used to create Workflow which is a set of instructions to execute mappings. Generally, a workflow contains a Session and any other task you may want to perform when you execute a session Tasks can include a Session, Email, or Scheduling information. You connect each task with Links in the workflow You can also create a worklet in the Workflow Manager. A Worklet is an object that groups a set of tasks.
- **Workflow Monitor.** Workflow Monitor is used to monitor workflows and tasks. You can view details about a workflow or task in either Gantt Chart view or Task view. You can run, stop, abort, and resume workflows. Displays workflows that have run at least once. Continuously receives information from the Integration Service and Repository Service. Fetches information from the repository to display historic information.

## Informatica Services Connectivity

The Integration Service process connects to the source or target database using ODBC or native drivers. It connects to the Repository Service to retrieve workflow and mapping metadata from the repository database. The Informatica Client tools like the Workflow Manager communicates with the Integration Service and the Repository Service process over a TCP/IP connection.



## New Features

New enhancements have been introduced in Informatica 7.1 in the following components:

- Data Profiling
- Licensing
- PowerCenter Server
- PowerCenter Metadata Reporter
- Repository Server
- Repository
- Transformations
- Usability
- Version Control
- Web services provider
- Workflow Monitor
- XML Support
- SuSE Linux Support
- Oracle OS Authentication
- Flat file Lookup

New enhancements have been introduced in Informatica 8.1 in the following components:

- Command Line Programs
- Data Analyzer (PowerAnalyzer)
- Data Profiling
- Domains
- Integration Service (PowerCenter Server)
- Mappings
- Metadata Manager (SuperGlue)
- Repository and Repository Service
- Transformations and Caches
- Usability
- Web Services Provider
- Workflow Monitor
- XML
- PowerCenter Connects
- PowerExchange Client for PowerCenter

## Summary

---

- ❖ ETL stands for extraction, transformation and loading.
- ❖ Extraction is fetching data from source operational or archive systems which are the primary source of data for the data warehouse.
- ❖ Transformation of the data involves cleaning, filtering, validating and applying business rules.
- ❖ Loading is the process of loading the data into a data warehouse or any other database or application that houses data.
- ❖ Informatica provides a single enterprise data integration platform.
- ❖ With Informatica, you can achieve
  - Enterprise level data integration
  - Scalability
  - Developer productivity
- ❖ Client tools of Informatica includes
  - Designer
  - Workflow Manager
  - Workflow Monitor
  - Repository Manager
  - Repository Administration Console

## Test Your Understanding

---

1. Expand and Explain ETL.
2. \_\_\_\_\_ is the definition of the relationship and data flow between source and target objects.
3. The process of resolving inconsistencies and fixing the anomalies in source data is called \_\_\_\_\_.
4. A database, application, file, or other storage facility to which the "transformed source data" is loaded in a data warehouse is called \_\_\_\_\_.
5. Name the Server components in Informatica.
6. Name the Client components in Informatica.
7. Which service manages the scheduling and execution of workflows?
8. Which service manages the metadata in the repository database?
9. Which tool is used to create/delete/backup/restore repositories?
10. Which tool is used in administering the repository?



## Session 2: Integration Service

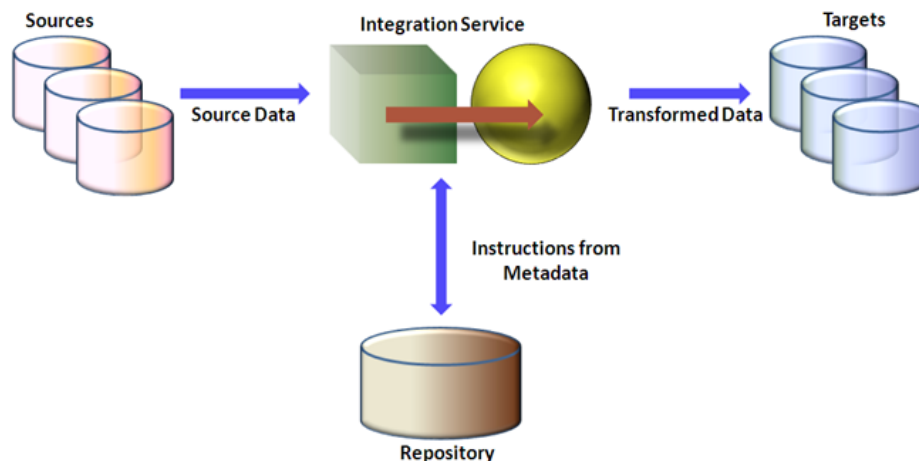
### Learning Objectives

- ❑ Integration Service Overview
- ❑ Integration Service Process
- ❑ Load Balancer
- ❑ Data Transformation Manager (DTM) Process
- ❑ Integration Service and the Administration Console

### Integration Service Overview

The Integration Service moves data from sources to targets based on workflow and mapping metadata stored in a repository. When a workflow starts, the Integration Service retrieves mapping, workflow, and session metadata from the repository. It extracts data from the mapping sources and stores the data in memory while it applies the transformation rules configured in the mapping. The Integration Service loads the transformed data into one or more targets.

Figure below shows the processing path between the Integration Service, repository, source, and target:



To move data from sources to targets, the Integration Service uses the following components:

- **Integration Service process.** The Integration Service starts one or more Integration Service processes to run and monitor workflows. When you run a workflow, the Integration Service process starts and locks the workflow, runs the workflow tasks, and starts the process to run sessions.
- **Load Balancer.** The Integration Service uses the Load Balancer to dispatch tasks. The Load Balancer dispatches tasks to achieve optimal performance. It may dispatch tasks to a single node or across the nodes in a grid.
- **Data Transformation Manager (DTM) process.** The Integration Service starts a DTM process to run each Session and Command task within a workflow. The DTM process performs session validations, creates threads to initialize the session, read, write, and transform data, and handles pre- and post- session operations.

The Integration Service can achieve high performance using symmetric multi-processing systems. It can start and run multiple tasks concurrently. It can also concurrently process partitions within a single session. When you create multiple partitions within a session, the Integration Service creates multiple database connections to a single source and extracts a separate range of data for each connection. It also transforms and loads the data in parallel. Integration Service can be created on a Windows or UNIX server machine. The Integration Service could be configured using the Administration Console or the pmcmd command line program.

## Integration Service Process

The Integration Service starts an Integration Service process to run and monitor workflows. The Integration Service process accepts requests from the PowerCenter Client and from pmcmd. It performs the following tasks:

- Manages workflow scheduling.
- Locks and reads the workflow.
- Reads the parameter file.
- Creates the workflow log.
- Runs workflow tasks and evaluates the conditional links connecting tasks.
- Starts the DTM process or processes to run the session.
- Writes historical run information to the repository.
- Sends post-session email in the event of a DTM failure.

### Managing Workflow Scheduling

The Integration Service process manages workflow scheduling in the following situations:

- **When you start the Integration Service.** When you start the Integration Service, it queries the repository for a list of workflows configured to run on it.
- **When you save a workflow.** When you save a workflow assigned to an Integration Service to the repository, the Integration Service process adds the workflow to or removes the workflow from the schedule queue.

### Locking and Reading the Workflow

When the Integration Service process starts a workflow, it requests an execute lock on the workflow from the repository. The execute lock allows it to run the workflow and prevent you from starting the workflow again until it completes. If the workflow is already locked, the Integration Service process cannot start the workflow. A workflow may be locked if it is already running.

The Integration Service process also reads the workflow from the repository at workflow run time. It reads all links and tasks in the workflow except sessions and worklet instances. The DTM retrieves the session and mapping from the repository at session run time. It reads the worklets from the repository when the worklet starts.

## Load Balancer

The Load Balancer is a component of the Integration Service that dispatches tasks to achieve optimal performance and scalability. When you run a workflow, the Load Balancer dispatches the Session, Command, and predefined Event-Wait tasks within the workflow. The Load Balancer matches task requirements with resource availability to identify the best node to run a task. It dispatches the task to an Integration Service process running on the node. It may dispatch tasks to a single node or across nodes.

The Load Balancer dispatches tasks in the order it receives them. When the Load Balancer needs to dispatch more Session and Command tasks than the Integration Service can run, it places the tasks it cannot run in a queue. When nodes become available, the Load Balancer dispatches tasks from the queue in the order determined by the workflow service level.

The following concepts describe Load Balancer functionality:

- **Dispatch process.** The Load Balancer performs several steps to dispatch tasks.
- **Resources.** The Load Balancer can use PowerCenter resources to determine if it can dispatch a task to a node.
- **Resource provision thresholds.** The Load Balancer uses resource provision thresholds to determine whether it can start additional tasks on a node.
- **Dispatch mode.** The dispatch mode determines how the Load Balancer selects nodes for dispatch.
- **Service levels.** When multiple tasks are waiting in the dispatch queue, the Load Balancer uses service levels to determine the order in which to dispatch tasks from the queue.

## Data Transformation Manager (DTM) Process

When the workflow reaches a session, the Integration Service process starts the DTM process. The DTM is the process associated with the session task. The DTM process performs the following tasks:

- Retrieves and validates session information from the repository.
- Performs pushdown optimization when the session is configured for pushdown optimization.
- Adds partitions to the session when the session is configured for dynamic partitioning.
- Forms partition groups when the session is configured to run on a grid.
- Expands the service process variables, session parameters, and mapping variables and parameters.
- Creates the session log.
- Validates source and target code pages.
- Verifies connection object permissions.
- Runs pre-session shell commands, stored procedures, and SQL.
- Sends a request to start worker DTM processes on other nodes when the session is configured to run on a grid.
- Creates and runs mapping, reader, writer, and transformation threads to extract, transform, and load data.
- Runs post-session stored procedures, SQL, and shell commands.
- Sends post-session email.

## Integration Service and the Administration Console

The Integration Service is an application service that runs sessions and workflows. You install the Integration Service when you install PowerCenter Services. After you install the PowerCenter Services, you can use the Administration Console to manage the Integration Service. To access an Integration Service, you must have permission on the service in the domain.

You can use the Administration Console to complete the following Integration Service tasks:

- **Create an Integration Service.** You can create an Integration Service to replace an existing Integration Service or to use multiple Integration Services.
- **Enable or disable the Integration Service.** You enable the Integration Service to run sessions and workflows. You might disable the Integration Service to prevent users from running sessions and workflows while performing maintenance on the machine or modifying the repository.
- **Configure normal or safe mode.** Configure the Integration Service to run in normal or safe mode.
- **Configure the Integration Service properties.** You might need to configure the Integration Service properties to change behavior of the Integration Service.
- **Configure the associated repository.** You must associate a repository with an Integration Service. The Integration Service uses the mappings in the repository to run sessions and workflows.
- **Configure the Integration Service processes.** Configure service process properties for each node, such as the code page and service process variables.
- **Configure permissions on the Integration Service.** You must have permission on an Integration Service to view it and full privileges to configure it.
- **Remove an Integration Service.** You may need to remove an Integration Service if it becomes obsolete. Use the Administration Console to remove an Integration Service.

### Creating an Integration Service

You can create an Integration Service when you install PowerCenter. However, you may need to create an additional Integration Service to replace an existing one or create multiple Integration Services. You can assign the repository when you create the Integration Service or after you create it. You must assign a repository before you can run the Integration Service. The repository

that you assign to the Integration Service is called the associated repository. The Integration Service retrieves metadata, such as workflows and mappings, from the associated repository.

After you create an Integration Service, you must assign a code page for each Integration Service process. The code page for each Integration Service process must be a subset of the code page of the associated repository. You must select the associated repository before you can select the code page for an Integration Service process. The Repository Service must be enabled to set up a code page for an Integration Service process. To create an Integration Service, you must have full privileges and permission on the domain or folder where you want to create the service. You must also have permission on the grid or node where the service runs.

**Note:** If you configure an Integration Service to run on a node that is unavailable, you must start the node and configure \$PMRootDir for the service process before you run workflows with the Integration Service.

*To create an Integration Service:*

1. In the Administration Console, click Create > Integration Service.  
The Create New Integration Service page appears.
2. Enter values for the Integration Service options. Click OK.

3. If you do not specify an associated repository, the following message appears:  
*No Repository Service is associated with this Integration Service. Select an associated Repository Service to view a list of available codepages.*  
*You must specify a Repository Service before you can enable the Integration Service.*
4. If you specify an associated repository, the Create New Integration Service page appears:

5. You can specify the code page for each Integration Service process node and select the Enable Service option to enable the service. If you do not specify the code page information now, you can specify it later. You cannot enable the Integration Service until you assign the code page for each Integration Service process node.

### Enabling and Disabling the Integration Service

You can enable and disable an Integration Service process or the entire Integration Service. If you run the Integration Service on a grid or with the high availability option, you have one Integration Service process configured for each node. For a grid, the Integration Service runs all enabled Integration Service processes. With high availability, the Integration Service runs the Integration Service process on the primary node.

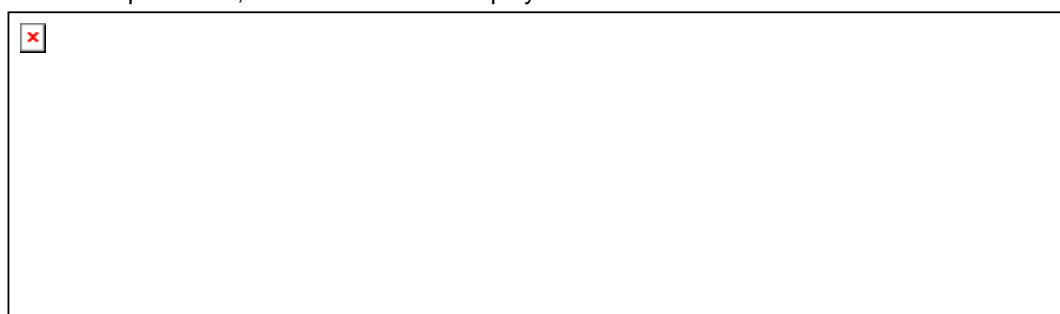
Use the Administration Console to enable and disable a service process running the Integration Service. Each service process runs on one node. You must enable the service process if you want the node to perform Integration Service tasks. You may want to disable the service process on a node to perform maintenance on that node or to enable safe mode for the Integration Service. When you disable an Integration Service process, you must choose the mode to disable it in.

You can choose one of the following options:

- **Complete.** Allows the sessions and workflows to complete before disabling the service process.
- **Stop.** Stops all sessions and workflows and then disables the service process.
- **Abort.** Attempts to stop all sessions and workflows before aborting them and disabling the service process.

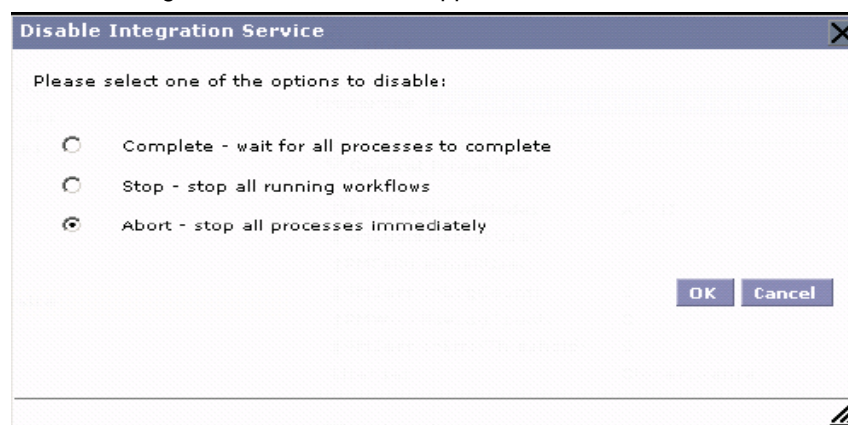
*To enable or disable an Integration Service process:*

1. In the Navigator of the Administration Console, select the Integration Service.
2. Click the Processes tab. If you configure the Integration Service to run on a grid or on multiple nodes, the Processes tab displays one area for each node.



3. To enable a service process, click Enable for the applicable node.
4. To disable a service process, click Disable for the applicable node.

The Disable Integration Service window appears.



5. Choose the disable mode and click OK.

When you enable the Integration Service, the service starts. The associated Repository Service must be started before you can enable the Integration Service. If you try to enable an Integration Service and the associated Repository Service is not running, the following error displays:

*The Service Manager could not start the service due to the following error: [DOM\_10076] Unable to enable service [<Integration Service>] because of dependent services [<Repository Service>] are not initialized.*

If the Integration Service is unable to start, the Service Manager keeps trying to start the service until it reaches the maximum restart attempts defined in the domain properties. For example, if you try to start the Integration Service without specifying the code page for each Integration Service process, the domain tries to start the service. The service does not start without specifying a valid code page for each Integration Service process. The domain keeps trying to start the service until it reaches the maximum number of attempts.

If the service is enabled but fails to start after reaching the maximum number of attempts, the following message appears:

*The Integration Service <service name> is enabled.  
The service did not start. Please check the logs for more information.*

To resolve the problem, review the logs for this Integration Service to determine the reason for failure and fix the problem. After fixing the problem, you must disable and re-enable the Integration Service to start it.

### Configuring the Integration Service Properties

Use the Administration Console to configure the following Integration Service properties:

- **Grid and node assignments.** Choose to run the Integration Service on a grid or nodes.
- **General properties.** Configure general properties, such as the data movement mode.
- **Advanced properties.** Configure advanced properties, such as the character set of the Integration Service logs and whether the Integration Service should check resources.
- **Operating mode properties.** Enable the Integration Service in normal or safe mode and configure the Integration Service to fail over in safe mode.
- **Compatibility and database properties.** Configure the source and target database properties, such the maximum number of connections, and configure properties to enable compatibility with previous versions of PowerCenter.
- **Configuration properties.** Configure the configuration properties, such as the data display format.
- **HTTP proxy properties.** Configure the connection to the HTTP proxy server.
- **Custom properties.** Custom properties include properties that are unique to your PowerCenter environment or that apply in special cases. An Integration Service has no custom properties when you initially create it. Use custom properties only if Informatica Technical Support instructs you to.

To view and update properties, select the Integration Service in the Navigator. The Properties tab for the service appears.

### Configuring the Associated Repository

When you create the Integration Service, you specify the repository associated with the Integration Service. However, you may need to change the repository connection information. For example, you need to update the connection information if the repository is moved to a new database. You may need to choose a different repository when you move from a development repository to a production repository.

When you update or choose a new repository, you must specify the repository database user name and password, the Repository Service, and the associated domain. The Administration Console lists the Repository Services defined in the selected domain.

The Repository Service and Integration Service can be in different domains. If you want to choose a domain that is not listed in the Administration Console, you can add the connection information for the other domain. To select a Repository Service in a different domain, you must select the



domain and Repository Service, and then enter the host and port number for the Repository Service.

To edit the associated repository properties, select the Integration Service in the Domain tab of the Administration Console, and then click the Properties tab > Associated Repository Properties > Edit.

The screenshot shows the 'Associated Repository Properties' dialog box. At the top, there are five tabs: 'Properties', 'Associated Repository' (which is selected), 'Processes', 'Permissions', and 'Logs'. Below the tabs, the title 'Associated Repository Properties' is displayed on the left, and 'OK' and 'Cancel' buttons are on the right. The main area contains four labeled fields: 'Domain for Associated Repository Service' with a dropdown menu showing 'Development' and a 'Manage Domain List' button; 'Associated Repository Service' with a dropdown menu showing 'HR\_S11'; 'Repository User Name' with a text box containing 'Admin\_HR80'; and 'Repository Password' with a text box filled with asterisks. The dialog is enclosed in a standard Windows-style border.

## Summary

---

- ❖ The Integration Service moves data from sources to targets based on workflow and mapping metadata stored in a repository.
- ❖ The components of Integration Service are
  - Integration Service Process
  - Load Manager
  - Data Transformation Manager (DTM) Process
- ❖ The Administration Console Tool is used to create/configure/remove and Integration Service.

## Test your Understanding

---

1. True or False: The Integration Service can start and run multiple workflows concurrently.
2. Name the 3 components of the Integration Service.
3. What is the tool which is used to create/configure/remove an Integration Service?



## Session 3: Repository Service

### Learning Objectives

- ☐ PowerCenter Repository
- ☐ Administration Console
- ☐ Creating Repository Service
- ☐ Creating Repository Content
- ☐ Deleting Repository Content
- ☐ Backing up a Repository
- ☐ Restoring a Repository
- ☐ Licensing

### PowerCenter Repository

The PowerCenter Repository is a relational database managed by the Repository Service. The Repository consists of database tables that store metadata. Metadata describes different types of objects, such as mappings and transformations, that you can create or modify using the PowerCenter Client tools. The repository also stores administrative information such as user names, passwords, permissions, and privileges.

You can connect to and manage multiple repositories. A repository domain is a group of repositories that you can connect to simultaneously in the PowerCenter Client. They share metadata through a special type of repository called a global repository. When you configure shared folders in a repository, you can share the objects in the folder with other repositories in the repository domain. You share objects to reuse metadata.

The Repository Service manages repository metadata transaction requests from client applications. Each Repository Service manages a single repository. The Repository Service uses object-locking to ensure the consistency of metadata in the repository. The Repository Service also notifies you when another user modifies or deletes repository objects you are using.

A Repository Service process is a multi-threaded process that fetches, inserts, and updates metadata in the repository database tables. A Repository Service process is an instance of the Repository Service that runs on a particular machine, or node.

### Administration Console

The PowerCenter Administration Console is the administration tool you use to administer the PowerCenter domain. If you have a user login to the domain, you can access the Administration Console. Use the Administration Console to perform administrative tasks such as managing logs, user accounts, and domain objects. Domain objects include services, nodes, grids, folders, and licenses.

The domain administrator or any user with full privileges and permission on the domain can create a login to the domain. You must have a domain user account to log in to the Administration Console.

*To log in to the Administration Console:*

1. Open Microsoft Internet Explorer or Mozilla Firefox.
2. In the Address field, enter the following URL for the Administration Console login page:  
`http://<host>:<port>/adminconsole`

In the URL, <host>:<port> represents the host name and port number of any gateway node. If you are not using the Internet Explorer Enhanced Security Configuration, you can enter the following URL, and the browser is directed to the full URL for the login page:

`http://<host>:<port>`

The Informatica PowerCenter Administration Console login page appears.



The login page features a purple header with 'Informatica PowerCenter Administration Console' and 'Help About' links. Below the header is a 'Login' section with 'Username:' and 'Password:' labels, input fields, and a 'Login' button. The background shows a modern building with glass windows.

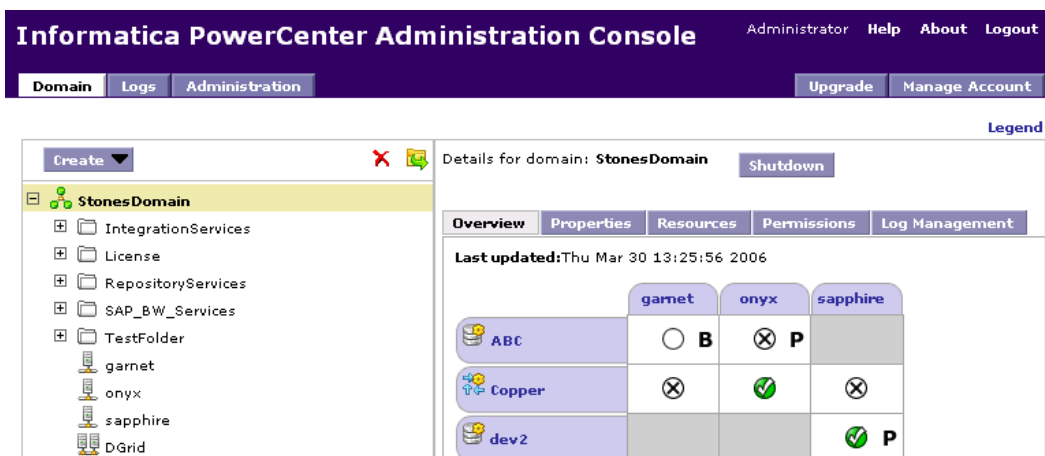
3. Enter the domain user name and password.
4. Click Login.
5. If this is the first time you log in with the user name and password provided by the domain administrator, change your password to maintain security.

### Administration Console Tabs

The Administration Console has tabs that are accessible at all times when you are logged in. Use these tabs to perform tasks such as viewing log events or configuring service properties.

The Administration Console has the following tabs:

- **Domain.** View and edit the properties of the domain and objects within the domain. The Domain tab has a Navigator and a Main window.
- **Logs.** View log events for the domain and services within the domain.
- **Administration.** Manage domain users in the Administration tab. This tab appears if you have permission on the domain.
- **Upgrade.** Upgrade repositories and Integration Services. This tab appears if you have permission on the domain.
- **Manage Account.** Manage your domain user profile. On this tab, you can change your domain user password and update your user profile.



The interface shows the 'Domain' tab selected. The top bar includes 'Administrator Help About Logout' and 'Domain Logs Administration Upgrade Manage Account' tabs. The main area is divided into a left 'Create' sidebar with a tree view of 'StonesDomain' (containing IntegrationServices, License, RepositoryServices, SAP\_BW\_Services, TestFolder, garnet, onyx, sapphire, and DGrid) and a right 'Details for domain: StonesDomain' section. The details section has a 'Shutdown' button and tabs for 'Overview Properties Resources Permissions Log Management'. The 'Overview' tab shows a table of services and their status across different environments.

	garnet	onyx	sapphire
ABC	○ B	⊗ P	
Copper	⊗	✓	⊗
dev2			✓ P

### Creating Repository Service

Create a Repository Service to manage the metadata in repository database tables. Each Repository Service manages a single repository. You need to create a unique Repository Service for each repository in a PowerCenter domain.

*To create a Repository Service:*

1. In the Navigator of the Administration Console, select the folder where you want to create the Repository Service.
2. Click Create > Repository Service. The Create New Repository Service window appears.
3. Enter values for the Repository Service options.
4. Click OK.

Specify properties for this new Repository Service.

**Service Name:**

**Location:**

**License:**

**Primary Node:**

**Backup Nodes:**

Node Name	Backup
garnet	<input type="checkbox"/>
onyx	<input type="checkbox"/>
sapphire	<input type="checkbox"/>

**DatabaseType:**

**CodePage:**

**ConnectionString:**

**DBUser:**

**DBPassword:**

**TablespaceName:**

**Creation Mode:**

☒ Create new repository content

☐ Create as global repository  
*Once set to global, the Repository Service cannot be reverted back to local.*

☐ Enable version control  
*A versioned repository cannot be unversioned.*

☐ Do not create repository content.  
*Content already exists under the specified database connection string*

☐ Enable the Repository Service

## Configuring Repository Service Properties

After you create a Repository Service, you can configure it. Use the Administration Console to configure the following types of Repository Service properties:

- **General properties.** Configure general properties, such as the operating mode.
- **Node assignments.** If you have the high availability option, configure the primary and backup nodes to run the service.
- **Database properties.** Configure repository database properties, such as the code page and connection string.
- **Advanced properties.** Configure advanced repository properties, such as the maximum connections and locks on the repository.

- **Custom properties.** Configure repository properties that are unique to your PowerCenter environment or that apply in special cases. Use custom properties only if Informatica Technical Support instructs you to do so.

To view and update properties, select the Repository Service in the Navigator. The Properties tab for the service appears.

The Repository Service **dev2** is enabled and running in normal mode.

[Disable](#)

The service is running.

☒ garnet ☐ sapphire

Actions	Properties	Processes	Connections	Locks	Plug-ins	Permissions	Logs
<div> <div>Node Assignments</div> <div> <div>Primary Node: garnet</div> <div>Backup Nodes: sapphire</div> </div> <div>Edit</div> </div>							
<div> <div>General Properties</div> <div> <div>OperatingMode: Normal</div> <div>SecurityAuditTrail: No</div> <div>GlobalRepository: false</div> <div>EnableVersionControl: true</div> <div>License: StonesLicense</div> </div> <div>Edit</div> </div>							
<div> <div>Database Properties</div> <div> <div>DatabaseType: MSSQLServer</div> <div>CodePage: MS Windows Latin 1 (ANSI), superset of Latin1</div> <div>ConnectionString: bozo@dev2</div> <div>TablesapceName:</div> <div>DBUser: dev_admin</div> <div>DBPassword: *****</div> <div>DatabaseConnectionTimeout: 180</div> <div>DatabaseArrayOperationSize: 100</div> <div>DatabasePoolSize: 500</div> <div>Table Owner Name:</div> </div> <div>Edit</div> </div>							

## Creating Repository Content

Repository content is the repository tables in the database. You can create repository content for a Repository Service if you did not create content when you created the service or if you deleted the repository content. You cannot create content for a Repository Service that already includes content.

*To create content after creating a Repository Service:*

1. In the Navigator, select a Repository Service that has no content associated with it.
2. In the Actions list, select Create Contents.

The page displays the options to create content.

Create contents for HR_80_Dev	OK	Cancel
<div>Create repository contents</div> <div> <div><input type="checkbox"/> This is a Global Data Repository</div> <div><input type="checkbox"/> Enable version control</div> </div>		

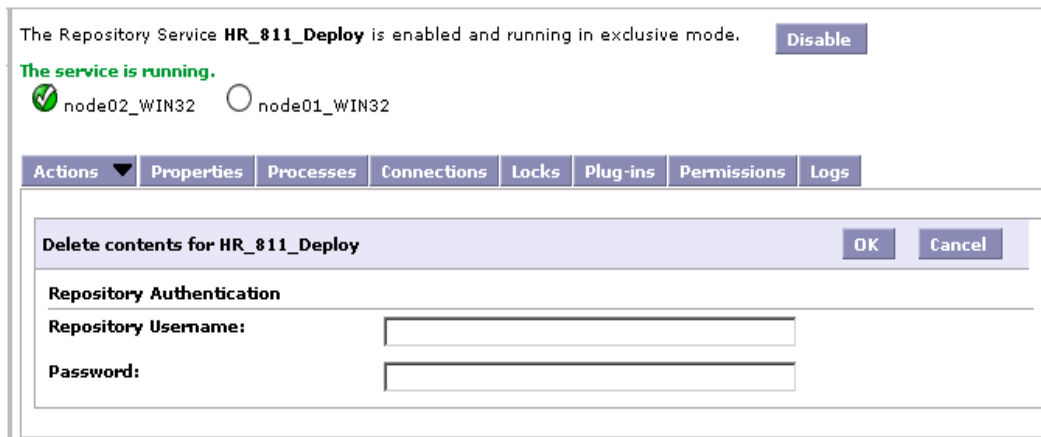
3. Optionally, choose to create a global repository.  
Select this option if you are certain you want to create a global repository. You can promote a local repository to a global repository at any time, but you cannot convert a global repository to a local repository.
4. Optionally, enable version control.  
You must have the team-based development option to enable version control. Enable version control if you are certain you want to use a versioned repository. You can convert a non-versioned repository to a versioned repository at any time, but you cannot convert a versioned repository to a non-versioned repository.
5. Click OK.

## Deleting Repository Content

Delete repository content when you want to delete all metadata and repository database tables from the repository. You might delete the repository content if the metadata is obsolete. Deleting repository content is an irreversible action. It is better to back up the repository before deleting it.

*To delete repository content:*

1. In the Navigator, select the repository from which you want to delete the content.
2. Change the operating mode of the Repository Service to exclusive.
3. In the Actions list, select Delete Contents.



The Repository Service **HR\_811\_Deploy** is enabled and running in exclusive mode. Disable

**The service is running.**

☒ node02\_WIN32 ☐ node01\_WIN32

**Actions** ▼ **Properties** **Processes** **Connections** **Locks** **Plug-ins** **Permissions** **Logs**

**Delete contents for HR\_811\_Deploy** OK Cancel

**Repository Authentication**

**Repository Username:**

**Password:**

4. Enter a repository user name and password. The repository user must have the Admin Repository privilege to delete repository content.
5. If the repository is a global repository, choose to unregister local repositories when you delete the content. The delete operation does not proceed if it cannot unregister the local repositories. For example, if a Repository Service for one of the local repositories is running in exclusive mode, you may need to unregister that repository before you attempt to delete the global repository.
6. Click OK.

## Backing up a Repository

Regularly back up repositories to prevent data loss due to hardware or software problems. When you back up a repository, the Repository Service saves the repository in a binary file, including the repository objects, connection information, and code page information. If you need to recover the repository, you can restore the content of the repository from this binary file.

*To back up a repository:*

1. In the Navigator, select the Repository Service for the repository you want to back up.

2. In the Actions list, select Back up Contents.

**Back up contents for HR\_80\_Dev** [OK] [Cancel]

**Repository Authentication**

Username:

Password:

**Backup file**

Filename:

Description:

☐ Replace existing file

**Advanced options**

☐ Skip workflow and session log

☐ Skip deployment group history

☐ Skip MX data

3. Enter a repository user name and password.
4. The repository user must have the Admin Repository or Super User privilege to perform backups.
5. Enter a file name and description for the repository backup file.
6. If you use the same file name that you used for a previous backup file, select whether or not to replace the existing file with the new backup file.
7. To overwrite an existing repository backup file, select Replace Existing File. If you specify a file name that already exists in the repository backup directory and you do not choose to replace the existing file, the Repository Service does not back up the repository.
8. Choose to skip or back up workflow and session logs, deployment group history, and MX data. You might want to skip these operations to increase performance when you restore the repository.
9. Click OK.

## Restoring a Repository

You can restore metadata from a repository binary backup file. When you restore a repository, you must have a database available for the repository. If a repository exists at the target database location, you must delete it before you restore a repository backup file. PowerCenter restores repositories from the current version. If you have a backup file from an earlier version of PowerCenter or PowerMart, you can use the earlier version to restore the repository. Verify that the repository license includes the license keys necessary to restore the repository backup file.

*To restore a repository:*



1. In the Navigator, select the Repository Service that manages the repository content you want to restore.
2. In the Actions list, select Restore Contents. The Restore Contents options appear.
3. Select a backup file to restore.  
If the repository backup file is associated with a security module, the restore security options appear.
4. In the restore security options appear, you must choose whether to restore the repository with the external module or unregister the security module when you restore.

The Repository Service **HR\_80\_Dev** is enabled and running in normal mode.

[Disable](#)

The repository has no content.

**The service is running.**

 sapphire  garnet

Actions...

Properties

Processes

Connections

Locks

Plug-ins

Permissions

Logs

Restore contents for HR\_80\_Dev

OK

Cancel

Select backup file

Select a backup file to restore repository contents

HR\_80\_Dev\_backup\_10-25-05.rep

☐ Restore as new

Advanced options

☐ Skip workflow and session log
 ☐ Skip deployment group history
 ☐ Skip MX data

If you choose to keep the external module, you must specify the repository administrator user name and password. Use the external module login that corresponds to the Administrator user for the repository.

If you choose to unregister the security module, the Administrator account password becomes "Administrator." The other user names are disabled. You must register the security module again to activate the user names.

5. Select whether or not to restore the repository as new. When you restore a repository as new, the Repository Service creates internal folder IDs for the folders in the restored repository. This enables you to copy folders and deployment groups between the original repository and the restored repository. If you do not restore the repository as new, you cannot copy folders and deployment groups between the original and restored repositories.

#### Restore security options

This backup contains a repository that uses an authentication module.

It can be either restored with the external module or without it.

- ☒ Keep the external module registered with the repository.
- Once restored, the repository will use the external authentication mechanism it was using when this backup was created.
- Provide the login to authenticate using this module:
- Login:
- Password:
- ☐ Unregister the security module.
- The built-in Administrator account password will be restored to 'Administrator' and all other users will be temporary disabled.

6. Optionally, choose to skip workflow and session logs, deployment group history, and MX data. Skipping the data improves performance.
7. Click OK.

## Licensing

A PowerCenter license enables you to perform the following tasks:

- **Run application services.** Application services include the Repository Service, Integration Service, SAP BW Service, and Web Services Hub.
- **Use PowerCenter features.** Features include connectivity, Metadata Exchange options, and other options, such as session on grid and high availability.

You can create a license object in a domain and assign the license to application services. You can create the license in the Administration Console using a license key file. The license key file contains an encrypted original key. You use the original key to create the license.

*To create a license:*

1. In the Administration Console, click Create > License.

The Create License window appears.

**Create License** [X]

**Name:**

**Description:**

**Location:**  [Select Folder](#)

**License File:**  [Browse...](#)

[Create](#) [Close](#)

2. Enter the options.
3. Click Create.



## Summary

---

- ❖ Repository is a relational database with unique Repository Server.
- ❖ Repository Service is a separate, multi-threaded process that retrieves, inserts, and updates metadata in the repository database tables.
- ❖ We can Create/Administer/Backup/Delete/Restore a Repository.

## Test your Understanding

---

1. What manages the connections to the PowerCenter Repository from Client Applications?
2. Where is a Repository Service created?
3. Where the Informatica objects and the passwords & privileges are are saved?
4. Repository is a \_\_\_\_\_ database.
5. How many Repository Services a Repository can have?

## Session 4: Repository Manager

### Learning Objectives

- ❑ Repository Manager Overview
- ❑ Understanding the Repository Manager Windows
- ❑ Connecting to a repository
- ❑ Managing Groups
- ❑ Managing Users
- ❑ Managing Privileges
- ❑ Managing Folders
- ❑ Viewing User Connections
- ❑ Viewing Locks
- ❑ Viewing Dependencies

### Repository Manager Overview

The Repository Manager is used to administer repositories. You can navigate through multiple folders and repositories, and complete the following tasks:

- **Manage users and groups.** Create, edit, and delete repository users and user groups. You can assign and revoke repository privileges and folder permissions.
- **Perform folder functions.** Create, edit, copy, and delete folders. Work you perform in the Designer and Workflow Manager is stored in folders. If you want to share metadata, you can configure a folder to be shared.
- **View metadata.** Analyze sources, targets, mappings, and shortcut dependencies, search by keyword, and view the properties of repository objects.

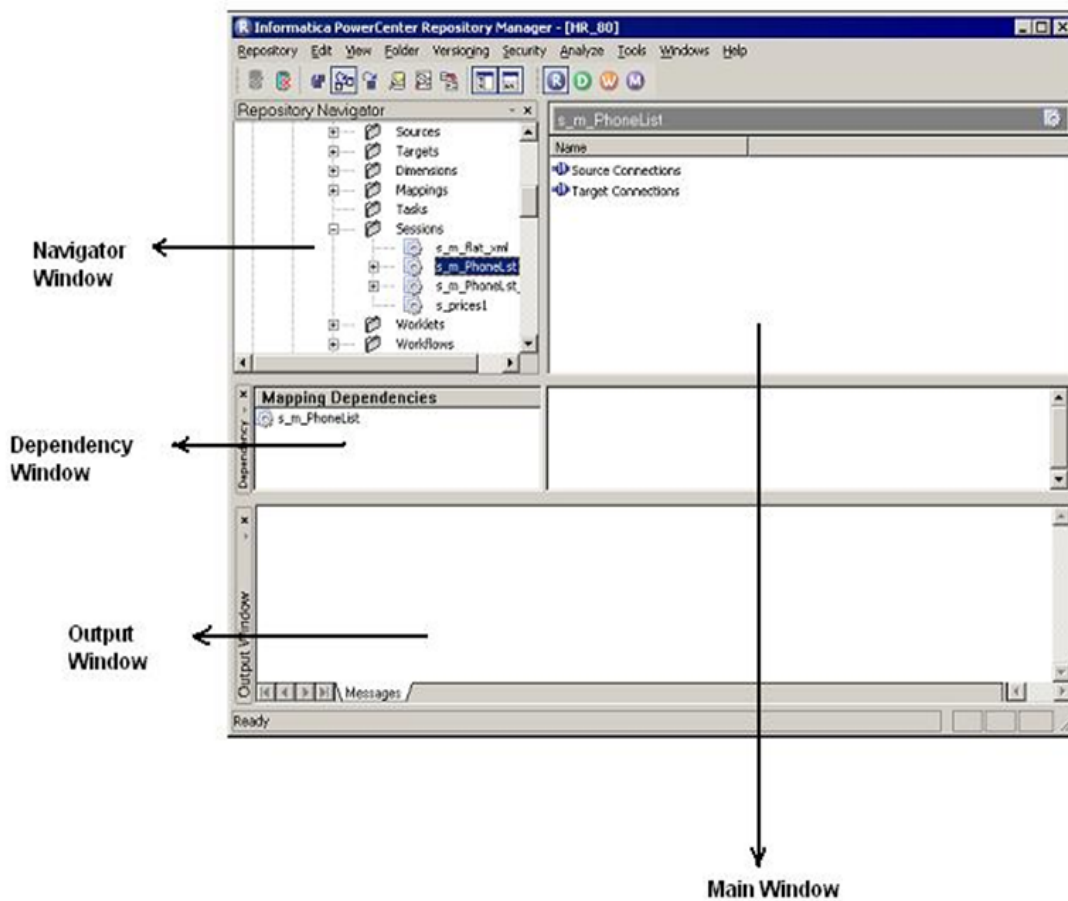
Use the Repository Manager to complete the following tasks:

- **Add domain connection information.** You can configure domain connection information.
- **Add and connect to a repository.** You can add repositories to the Navigator window and client registry and then connect to the repositories.
- **Work with PowerCenter domain and repository connections.** You can edit or remove domain connection information. You can connect to one repository or multiple repositories. You can export repository connection information from the client registry to a file. You can import the file on a different machine and add the repository connection information to the client registry.
- **Search for repository objects or keywords.** You can search for repository objects containing specified text. If you add keywords to target definitions, use a keyword to search for a target definition.
- **View Object dependencies.** Before you remove or change an object, you can view dependencies to see the impact on other objects.
- **Compare repository objects.** In the Repository Manager, you can compare two repository objects of the same type to identify differences between the objects.
- **Truncate session and workflow log entries.** You can truncate the list of session and workflow logs that the Integration Service writes to the repository. You can truncate all logs, or truncate all logs older than a specified date.
- **View user connections.** Use the Repository Manager to view and manage user connections.
- **Release locks.** Use the Repository Manager to release residual locks left by residual connections. Releasing a valid lock can result in repository inconsistencies, so release a lock only if the locked object is not in use.

- **Exchange metadata with other business intelligence tools.** You can export metadata to and import metadata from other business intelligence tools, such as Cognos ReportNet Framework Manager.

## Understanding the Repository Manager Windows

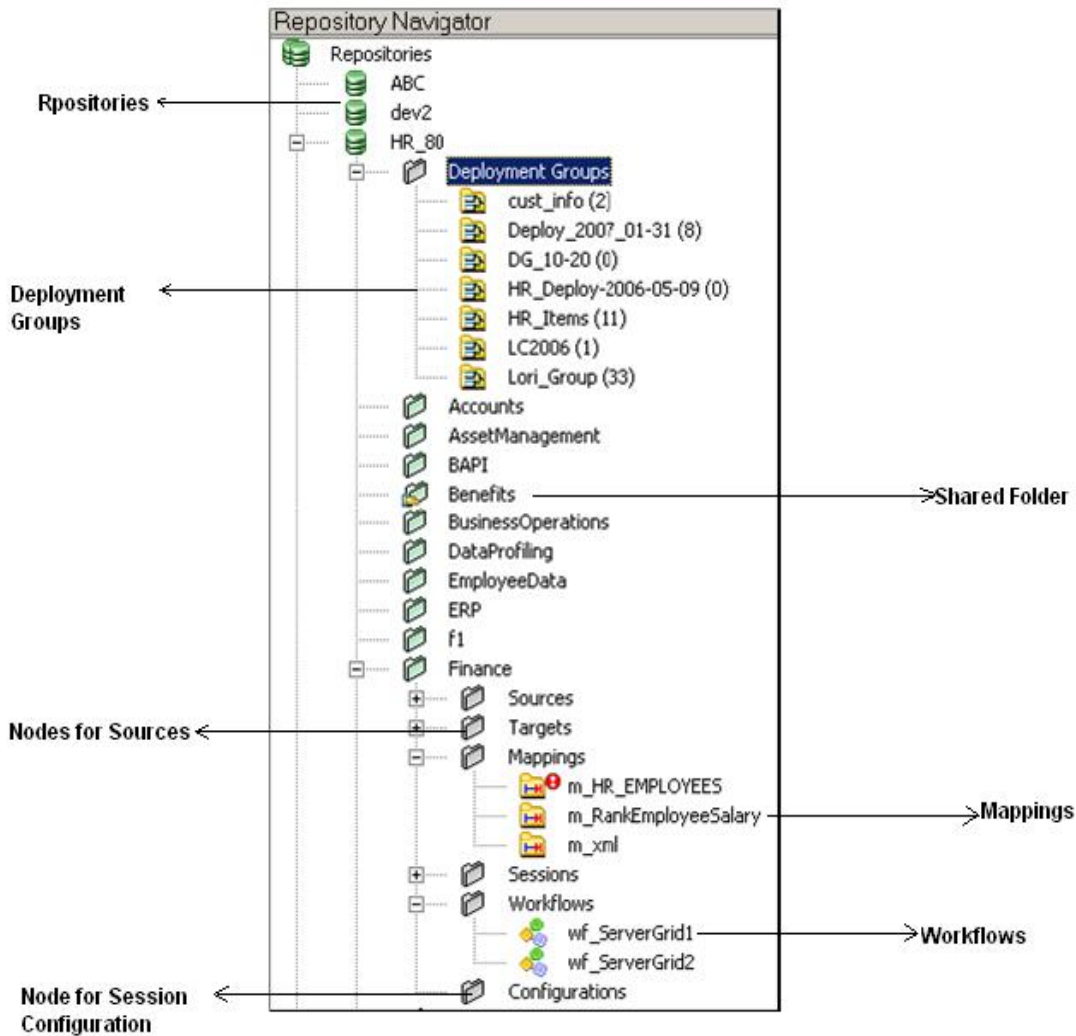
The Repository Manager can display four main windows: **Navigators window, Main window, Dependency window, and Output window.** You can dock and undock the Navigator, Dependency, and Output windows. You can also hide and display the Navigator, Dependency, and Output windows.



### Navigators Window

Use the Navigator window to connect to a repository and navigate through the folders and objects in the repository. The Navigator window displays the following types of objects:

- **Repositories.** PowerCenter repositories can be standalone, local, or global.
- **Deployment groups.** Deployment groups contain collections of objects for deployment to another repository in the repository domain.
- **Folders.** Folders can be shared or not shared.
- **Nodes.** Nodes contain sessions, sources, targets, transformations, mapplets, workflows, tasks, worklets, and mappings.
- **Repository objects.** Repository objects displayed in the Navigator can include sources, targets, transformations, mappings, mapplets, sessions, tasks, workflows, worklets, workflow logs, and session logs.



You can view the properties of the objects in the Navigator by selecting the object, and then clicking the View Properties button in the toolbar. You can also right-click the object in the Navigator and select Properties from the shortcut menu.

### Main Window

The Main window of the Repository Manager displays information about the object selected in the Navigator. For example, if you select a repository in the Navigator, the Main window displays all the folders in the repository along with additional folder information, such as whether the folder is shared or in use.

### Sorting and Organizing

You can sort items in the Main window by each of the columns. For example, to sort mappings by validity, select the mappings node, and then click the Valid column heading. Click the heading again to reverse the order in which the mappings are sorted.

You can also change the order in which the columns appear. For example, you might want the Valid column to appear first, on the left side of the Main window. To do this, drag the Valid column heading to the location. The Repository Manager displays the columns in the new order until you change the display.

*Note:* You can double-click an object in the Main window to view its properties.

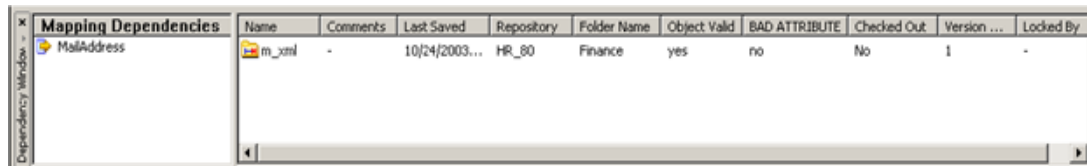
## Viewing Object Details

To view information about repository objects, select a node in the Navigator. Or, to view detailed information about a particular repository object, drill down on a node and select the object.

## Dependency Window

The Dependency window appears when you configure the Repository Manager to display dependencies. You can view dependencies by using the menu items or the dependency buttons on the toolbar. You can also view dependencies using the Dependency dialog box.

When you view dependencies, the left pane of the Dependency window lists the object that has dependency information, and the dependency information appears in the right pane.



The Dependency window can display the following types of dependencies:

- **Source-target dependencies.** When you view source-target dependencies, the Dependency window lists all sources or targets related to the selected object, including relevant information about those sources or targets. For example, if you select a target, you view all sources that provide data for that target, along with information about each source.
- **Mapping dependencies.** When you view mapping dependencies, the Dependency window lists all mappings containing the selected object, and relevant information about those mappings. For example, if you select a reusable transformation, you view all mappings using that transformation and information about each mapping.
- **Shortcut dependencies.** When you view shortcut dependencies, the Dependency window lists all shortcuts to the selected object and relevant details, such as the repository and folder in which the shortcut exists. When you edit an object referenced by shortcuts, use the Repository Manager to see which folders contain the shortcut.

When you open the Dependency window, it displays dependencies for the object selected in the Navigator. You can also view mapping object dependencies in the Designer.

## Output Window

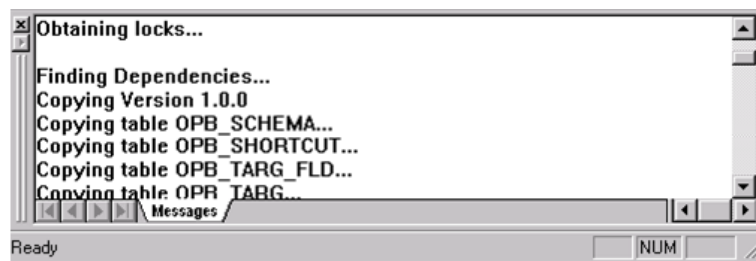
The Repository Manager displays status messages in the status bar. For complex operations, the Repository Manager displays detailed information in the Output window.

For example, when you connect to a repository, the status bar displays the following message:

*Accessing the repository...*

After you connect to the repository, the status bar displays the word *Ready*.

When you perform a complex operation, such as copying a folder, the Repository Manager displays details about the operation in the Output window.



The Repository Manager receives notifications when the following objects are created, modified, or deleted:

- Folders
- Users
- Groups

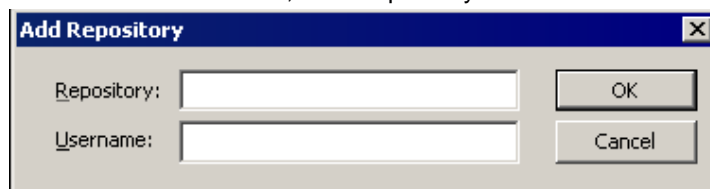
You must be connected to the repository to receive notifications about folders. You need the Super User privilege to receive notifications about users and groups.

## Connecting to a Repository

Add a repository when a repository already exists but does not appear in the Navigator. You can add a repository in the Repository Manager, the Designer, the Workflow Manager, or the Workflow Monitor. After you add a repository in one of the PowerCenter Client tools, it appears in the Navigator window of all the tools.

*To add a repository to the Navigator:*

1. In any of the PowerCenter Client tools, click Repository > Add.



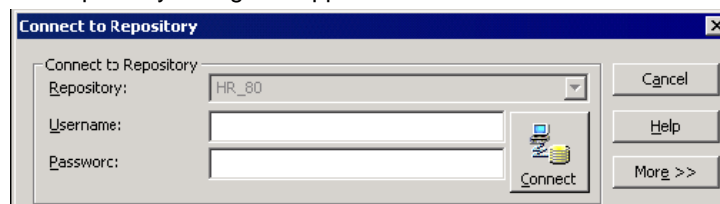
2. Enter the name of the repository and a valid repository user name.
3. Click OK.

The repository appears in the Navigator of the PowerCenter Client tools. Before you can connect to the repository for the first time, you must configure the connection information for the domain that the repository belongs to.

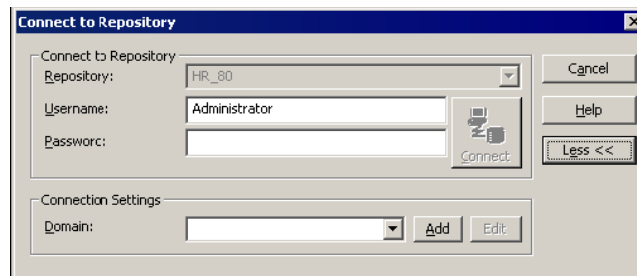
*To connect to a repository:*

1. Launch a PowerCenter Client tool.
2. Select the repository in the Navigator and click Repository > Connect, or double-click the repository.

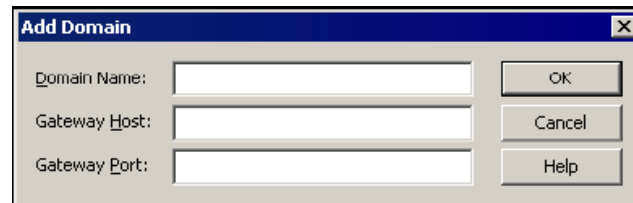
The Connect to Repository dialog box appears.



3. Enter a valid repository user name and password.
4. If you are connecting to a repository that you have connected to before, Go to step 11.  
-Or-  
If you want to select an existing domain connection for the repository, Go to step 9.  
-Or-  
If you need to enter the domain connection information for the repository, Complete steps 5 through 8.
5. If the Connect to Repository dialog box is not expanded, click More to view the connection settings



6. Click Add. The Add Domain dialog box appears.



7. Enter the domain name, gateway host name, and gateway port number. Click OK.
8. If the connection settings of the Connect to Repository dialog box are hidden, click More to view the connection settings.
9. In the connection settings, select the appropriate domain connection from the list.
10. Click Connect.

## Managing Groups

You can create custom user groups to manage users and repository privileges efficiently. After creating a new user group, you assign that group a set of privileges. Each repository user must be assigned to at least one user group. You can also assign users to multiple groups. This grants the user the privileges of each group.

When you assign a user to a group, the user:

- Receives all group privileges.
- Inherits any change to group privileges.
- Losses and gains privileges if you change the user group membership.

When you revoke a privilege from a group, you revoke the privilege from each user in the group. However, if a user has been granted the privilege individually (as with default privileges), you must revoke the privilege individually to restrict that user.

### Default Groups

When you create a repository, the Repository Manager creates two repository user groups. These two groups exist so you can immediately create users and begin developing repository objects. However, you can create additional groups and assign privileges and permissions to those groups.

The two default repository user groups:

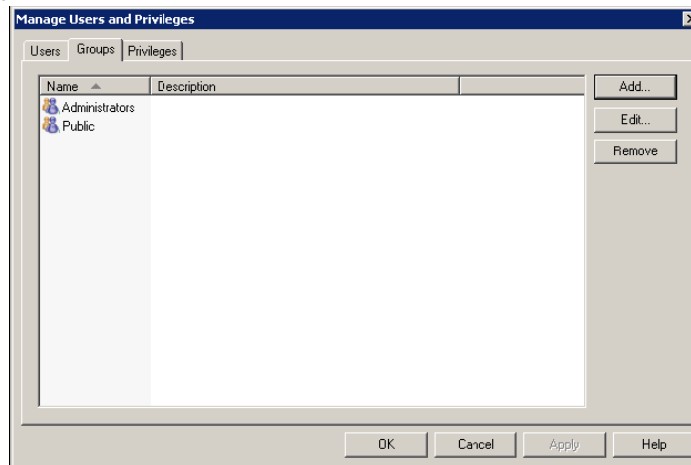
- **Administrators.** This group initially contains two users that are created by default. The default users are Administrator and the database user that created the repository. You cannot delete these users from the repository or remove them from the Administrators group.
- **Public.** The Repository Manager does not create any default users in the Public group.

You cannot delete these groups from the repository or change the default privileges.

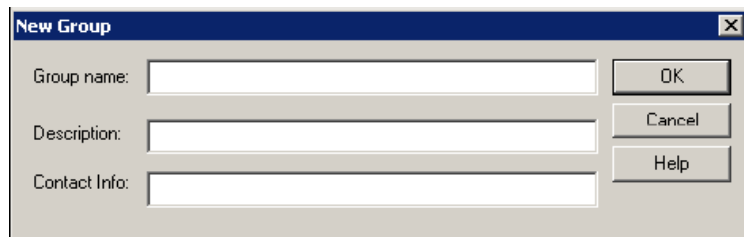
*To create a user group:*

1. In the Repository Manager, connect to a repository.
2. Click Security > Manage Users and Privileges.

- Click the Groups tab.



- Click Add.



- Enter the name of the group.
- Click OK. The new group now appears in the list of all groups defined for the repository.
- Click OK to return to the Repository Manager.  
You can now assign privileges and users to the group.

**Tip:** Before creating repository users, create user groups, and assign the appropriate privileges to those groups.

## Managing Users

In the Repository Manager, you create a user name for each user accessing the repository. Each user belongs to at least one user group. When you create a user, the Repository Manager assigns the user to the Public group. To change the group, you must first add the user to another group and then remove the user from the Public group. Each repository user needs a user name and password to access the repository. PowerCenter uses one of the following methods to authenticate users:

- PowerCenter default authentication
- Lightweight Directory Access Protocol (LDAP) authentication

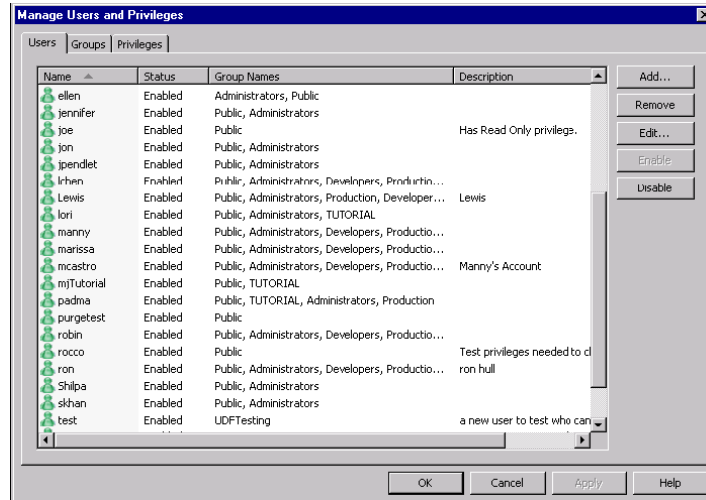
### Creating a User under Default Authentication

If you use default user authentication, enter user names in the repository and assign a password for each user name you create.

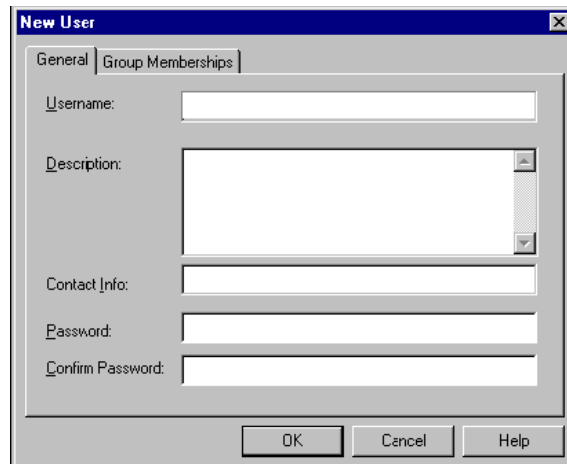
*To create a user under default authentication:*

- In the Repository Manager, connect to a repository.
- Click Security > Manage Users and Privileges.  
The Manage Users and Privileges dialog box appears.

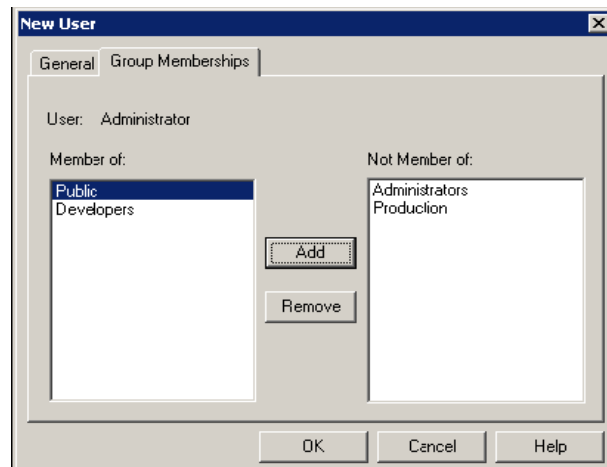




3. Click Add. The New User dialog box appears.
4. Enter a user name and password between 1 and 75 characters long, with no leading or trailing spaces. Use the optional Description and Contact Info fields to further identify the user.



5. Click Group Memberships. The Member Of list displays the groups that the user belongs to.
6. To add the user to a group, select the group in the Not Member Of list, and click Add. The group appears in the Member Of list.



7. To remove the user from a group, select the group in the Member Of list, and Click Remove.

Note: You cannot remove a user from a group when the user belongs to one group.

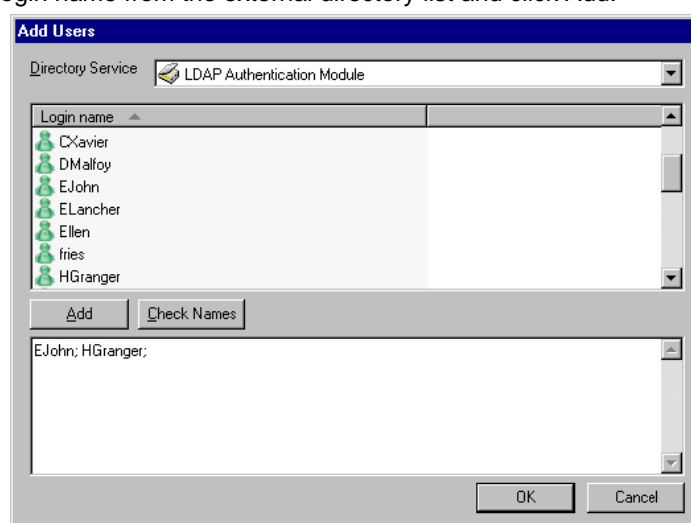
8. Click OK.

### Creating a User using an External Directory Service

If you use an external directory service for maintaining repository users, you add repository users from a list of login names from the external directory. You do not create repository user passwords.

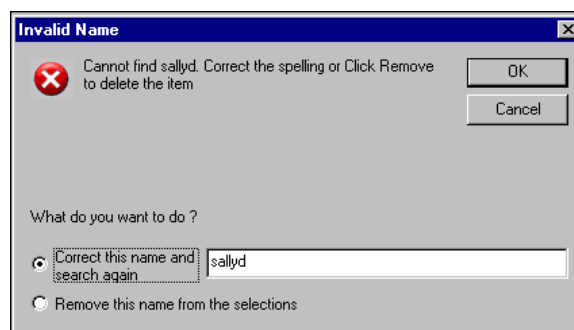
To add users with an external directory service:

1. In the Repository Manager, connect to a repository. Click Security > Manage Users and Privileges.
2. Click Add. The Add Users dialog box displays a list of names from the external directory service.
3. Select a login name from the external directory list and click Add.



The login name appears at the bottom of the dialog box. If you select more than one name, the names appear at the bottom separated by semicolons.

4. If any of the names you enter do not match a valid directory service login, you must correct the name or remove it from the selections on the Invalid Name dialog box. Click OK.



5. Click OK in the Add Names dialog box to add the selections to the user-name/login-name association. The Manage Users and Privileges window displays the user name, login name, and enabled status.

### Editing a User

You can edit users to maintain their group memberships and change their login status. If you use an external directory service you can change the user name-login association, so the repository

user name maps to a new login name. If you use default authentication, you can change a user password.

The following exceptions apply when you edit users:

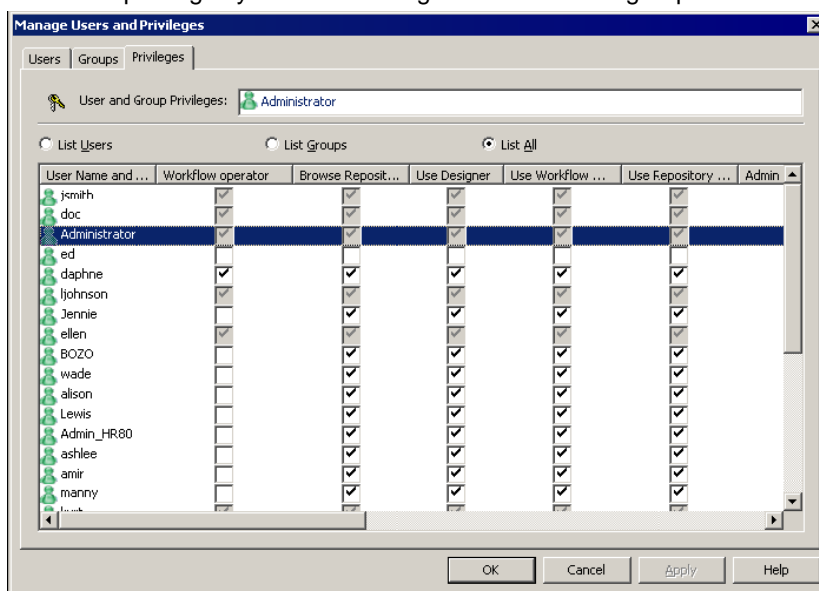
- You cannot edit passwords if you use an external directory for security.
- You cannot change group memberships for the Administrator user.
- You cannot edit any property for the Database user except contact information.
- You cannot change a user name. If a user name becomes obsolete, you can remove it from the repository.

## Managing Privileges

You can assign and revoke repository privileges of a user or group. When you change privileges for a group, you change the privileges granted to every user in the group. However, users retain the individually granted privileges until you revoke them. You cannot change the privileges of the default user groups or the default repository users.

*To assign a privilege to a user or group:*

1. In the Repository Manager, connect to a repository.
2. Click Security > Manage Users and Privileges. Select Privileges Tab.
3. Select the privileges you want to assign to each user or group.



4. Click OK to save the changes.

If you assign Super User privileges to a user, you cannot change the other privileges for that user. Super User privilege enables all privileges for the user.

## Managing Folders

Folders provide a way to organize and store all metadata in the repository, including mappings, schemas, and sessions. Folders are designed to be flexible and to help you logically organize the repository. Each folder has a set of configurable properties that enable you 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. You can create shared and non-shared folders.

*Note:* You must create a folder in a repository before you can connect to the repository using the Designer or Workflow Manager.

To create a folder:

1. In the Repository Manager, connect to the repository.
2. Click Folder > Create.

3. Enter the following information:

Folder Properties	Required/Optional	Description
Name	Required	Folder name. Do not use the period character ( . ) in folder names. Folder names with periods can cause errors when you run sessions.
Description	Optional	Description of the folder that appears in the Repository Manager.
Owner	Required	Owner of the folder. Any user in the repository can be the folder owner.
Owner's Group	Required	Repository user group to receive Owner's Group permissions. Groups to which the owner belongs can be selected.
Allow Shortcut	Optional	If selected, makes the folder shared.
Status	Required	Status applied to all objects in the folder. This option applies to versioned repositories.
Permissions	Required	Folder permissions for users in the repository.

4. Click OK. The folder appears in the Navigator. If the folder is shared, the folder displays an open hand icon.

## Shared Folders

You can configure folders in global and local repositories to be shared. After you designate a folder as shared, you can create shortcuts to objects in that folder. Use shortcuts in any other folder in the repository. If the shared folder is in a global repository, use shortcuts to that folder in any repository in the repository domain.

If a folder is not shared, you cannot create shortcuts to objects in the folder. However, you can still create copies of objects in non-shared folders.

Shared folders are useful when you want to use the same repository objects in several projects within the same repository. For example, each folder within a repository might represent a different development project. However, every project in the repository needs to store bug reports in a consistent format, so you might put the source definition for the BUG\_REPORTS table in a shared folder.

## Folder Permissions

Permissions allow repository users to perform tasks within a folder. With folder permissions, you control user access to the folder, and you control the tasks that users can perform. Folder permissions work closely with repository privileges. Privileges grant access to specific tasks while permissions allow you to read, write, and execute tasks within a specific folder.

Folders have the following permission types:

- **Read permission.** You can view the folder and objects in the folder.
- **Write permission.** You can create or edit objects in the folder.
- **Execute permission.** You can execute or schedule a workflow in the folder.

Any user with the Super User privilege has read, write, and execute all tasks across all folders in the repository and can perform any task using the PowerCenter Client tools.

## Permission Levels

You can grant three levels of security using folder permissions.

- **Owner.** The owner of the folder, as listed in the owner menu.
- **Owner's Group.** Each member of the group to which the owner belongs, including the owner. If the owner belongs to more than one group, you must select one of those groups in the folder dialog box. This becomes the Owner's Group.
- **Repository.** All groups and users in the repository.

Each permission level includes the members in the level above it. If you restrict permission from Owner or Owner's Group and then grant the same permission to Repository, Owner and Owner's Group retain the permission.

*Note:* All users in the Administrators group or with the Super User privilege have full access to all folders in the repository.

## Edit Folder Permissions

You can edit folder permissions and properties at any time. To create, edit, and delete folders, you must have the Manage Folder privilege.

*To edit folder permissions and properties:*

1. In the Repository Manager, connect to a repository.
2. Select the folder in the Navigator. If the folder does not appear in the Navigator, you do not have read permission for the folder.
3. Click Folder > Edit.
4. If necessary, select a new owner from the Owner menu.
5. If the owner belongs to more than one group, select a group from the Group menu.
6. Edit the folder permissions.
7. Click OK to save the changes.

## Comparing Folders

Before you copy or replace a folder in a repository, you may want to verify the contents of a folder or compare it with other folders. The Repository Manager lets you to quickly and accurately compare the objects in different folders using the Compare Folders Wizard. You must have read permission for each folder you want to compare. You must also have privileges to access the repositories containing the folders.

If you use a versioned repository, the Repository Manager uses the latest checked in version of an object for comparison.

In the Compare Folders Wizard, you can complete the following comparisons:

- Compare objects between two folders in the same repository.
- Compare objects between two folders in different repositories.

You can specify the following comparison criteria for each comparison:

- **Object types to compare.** You can specify the object types to compare and display between folders. The wizard compares objects based on specific object attributes.
- **Direction of comparison.** The wizard performs directional comparisons. A directional comparison checks the contents of one folder against the contents of the other. You can specify either one-way or two-way comparisons.

The wizard displays the following information:

- Similarities between objects
- Differences between objects
- Outdated objects

You can edit and save the result of the comparison. You can also compare two objects of the same type using Compare Objects Wizard. The wizard compares the attribute of each object in the source folder with the attribute of each object in the target folder. You can choose to compare objects based on the following criteria:

- **Different objects.** Object name and type exist in one folder but not the other.
- **Similar objects.** Object name, type, and modification date are the same in both folders.
- **Outdated objects.** Object modification date is older than objects with the same name.

The wizard does not compare the field attributes of the objects in the folders when performing the comparison. For example, if two folders have matching source names and column or port names but differing port or column attributes, such as precision or datatype, the wizard does not note these as difference.

The Compare Folders wizard displays similar objects in green text, unmatched objects denoting a difference in red text, and outdated objects in blue text. The Compare Folders Wizard always displays the total number of differences, similarities, and outdated objects found during the comparison, even if you do not choose to display differences or similarities in the edit field.

You can save the results of the comparison in the edit field to an .rtf or a .txt file. To retain the color and font attributes of the result, save it as an .rtf file. By default, the results file is saved in the My Documents directory.

Connect to the repositories containing the folders in the wizard.

*To compare folders:*

1. In the Repository Manager, click Folder > Compare.
2. Click Next.

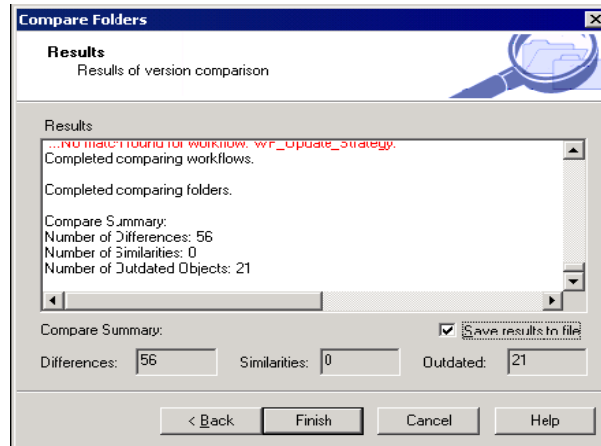
3. Connect to the repositories containing the folders you want to compare and select the folders for comparison. Click Next.

4. Select the object types you want to compare. Click Next.

5. Select the display options.

6. Click Next. The wizard always displays the number of differences, similarities, and outdated objects.
7. View the results of the comparison.

8. If you want to save the comparison results to a file, select Save results to file. Click Finish.



9. If you chose to save the results to a file, specify the file type, name, and directory. Click Save.

## Viewing User Connections

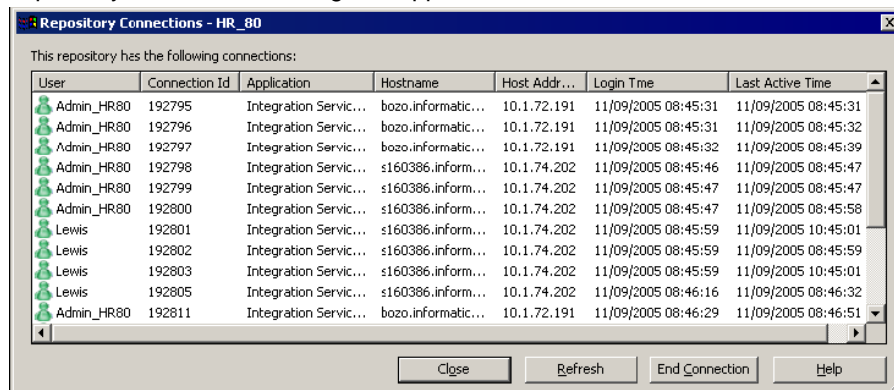
Use the Repository Manager to view connection details for users and applications connected to the repository. The following are the connection details viewable in Repository Manager:

- **User.** User name associated with the connection.
- **Connection ID.** Identification number assigned to the repository connection.
- **Application.** Repository client application associated with the connection.
- **Host Name.** Name of the machine running the application.
- **Host Address.** TCP/IP address of the machine associated with the connection.
- **Host Port.** Port number the machine hosting the repository client application uses to communicate with the repository.
- **Login Time.** Time the user connected to the repository.
- **Last Active Time.** Time of the last metadata transaction between the repository client application and the repository.

To view user connection details:

1. Launch the Repository Manager and connect to the repository.
2. Click Edit > Show User Connections.

The Repository Connections dialog box appears.



3. To sort the connections by column, click the column name.



## Viewing Locks

You can view locks and identify residual locks in the Repository Manager or in the Administration Console. The following are the object lock information available in both the Repository Manager and the Administration Console:

- **User.** User name locking the object.
- **Connection ID.** Identification number assigned to the repository connection.
- **Folder.** Folder in which the locked object is saved.
- **Object Type.** Type of object, such as folder, version, mapping, or source.
- **Object Name.** Name of the locked object.
- **Lock Type.** Type of lock: in-use, write-intent, or execute.
- **Lock Time.** Time the lock was created.
- **Hostname.** Name of the machine locking the object.
- **Application.** Application locking the object, such as Designer, Workflow Manager, Repository Manager, or Integration Service.

To show all repository locks:

1. In the Repository Manager, connect to a repository.
2. To view all locks in the repository, click Edit > Show locks.  
The Object Locks dialog box appears.

User	C...	Folder	Object Type	Object Name	Lock Type	Lock Time	Hostnar
Administr...	6	Developer Luo...	Workflow	wf_HR_ESC_RE...	execute	3/10/2009 10:31:0...	appub0i
Administr...	6	Developer Luo...	Workflow	wf_WMHR167_...	execute	3/10/2009 10:31:0...	appub0i
Administr...	6	Project RBT Inc...	Workflow	wf_ICMP_SCHE...	execute	3/11/2009 10:26:4...	appub0i
Administr...	6	Project WMFS IFP	Workflow	wkf_load_iir_...	execute	3/10/2009 9:00:02 PM	appub0i
Administr...	6	Project WMFS IFP	Workflow	wkf_nfs_anno...	execute	3/11/2009 4:00:04 ...	appub0i
Administr...	6	Project WMFS IFP	Workflow	wkf_rstr_stage...	execute	3/10/2009 9:00:03 PM	appub0i
Administr...	6	Project WMFS IFP	Workflow	wkf_suit_delta...	execute	3/10/2009 9:00:02 PM	appub0i
admin_R...	6	Project RBT Inc...	Workflow	wf_ICMP_JOB...	execute	3/11/2009 10:27:5...	appub0i
admin_R...	6	Project RBT Inc...	Workflow	wf_ICMP_PSFT...	execute	3/11/2009 10:25:0...	appub0i
u488482	292	Project DDR 3 S...	User Defined F...	DDR_ACCT_ID	in-use	3/11/2009 10:27:1...	*UNKNC
u488482	292	Project DDR 3 S...	User Defined F...	DDR_ACCT_O...	in-use	3/11/2009 10:27:0...	*UNKNC
u488482	292	Project DDR 3 S...	User Defined F...	DDR_ALL_0	in-use	3/11/2009 10:27:0...	*UNKNC

3. To sort the view of the locks by column, click the column name.
4. To view updated lock information, click Refresh.

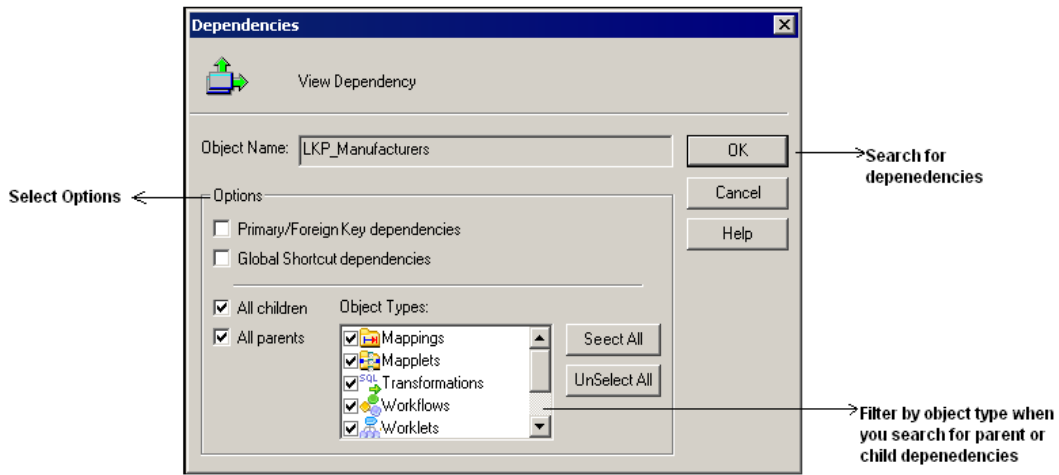
## Viewing Dependencies

Before changing or deleting the repository objects, you can view dependencies to see the impact on other objects. For example, before you remove a session, you can find out which workflows use the session. You can view dependencies for repository objects in the Repository Manager, Workflow Manager, and Designer tools.

In the Repository Manager, Workflow Manager, and Designer, you can view dependencies when you perform the following tasks:

- **View object history.** You can view dependencies from the View History window when you view the history of an object.
- **View checkouts.** You can view dependencies from the View Checkouts window when you search for persistent checkouts.
- **View query results.** You can view dependencies from the Query Results window when you search for object dependencies or run an object query.

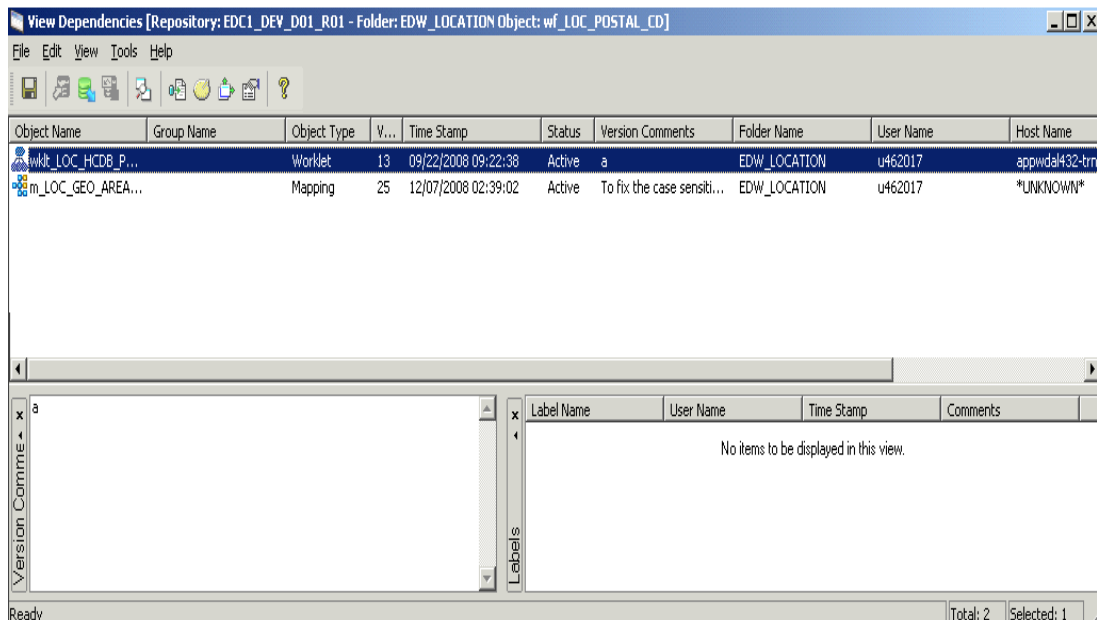
The Dependencies dialog box displays the object name, the object type, and a list of dependent objects and their properties.



When you search for dependencies, you can filter the results by selecting options and object types.

- **Primary/Foreign Key Dependencies.** View primary and source object dependencies where there is a primary key-foreign key relationship.
- **Global Shortcut Dependencies.** View global shortcuts across repositories. You can select this option when you search for parents, children, or primary key-foreign key dependencies.
- **All Children.** View the objects that the selected object uses. For example, if you search for the child objects for a workflow, the results might include sessions and worklets.
- **All Parents.** View the objects that use the selected object. For example, if you search for the parent objects for a session, the results might include a workflow or worklet.

The View Dependencies window also displays output, comments, and label information associated with the object. The Output window displays validation information, and the Comments window displays text entered during object check in or check out. The Labels window displays labels associated with the object and information associated with the label object. You can check in objects from the View Dependencies dialog box, but you cannot check in original objects for global shortcuts.



### Try It Out

---

1. Create a User Group.
2. Assign privileges "write and execute" to the created group.
3. Create Users and assign User Groups to them.
4. Create a folder without shortcut.
5. View locks
6. View Dependencies

### Summary

---

- ❖ The Repository Manager has 4 windows – Main, Navigator, dependency and output windows.
- ❖ Repository Manager allows the creation of groups.
- ❖ Groups can be assigned with the applicable privileges.
- ❖ Repository Manager allows the creation of users under the desired groups and folders with the applicable permissions and privileges.
- ❖ Comparison between two folders of the same or different repository can be done.
- ❖ Repository Manager shows all the dependency for the selected object.

### Test your Understanding

---

1. What are the 4 windows in the Repository Manager?
2. What are the two types of folders?
3. What are the types of dependencies available?
4. What are the default available groups?
5. What is the default group that gets assigned to when a new user is created in the Repository Manager?
6. What are the default permissions when a new folder is created?

## Session 5: Mapping Designer

### Learning Objectives

- ❑ Designer Overview
- ❑ Designer Tools
- ❑ Source Analyzer
- ❑ Working with Mappings
- ❑ Various Types of Transformations
- ❑ Mapping Parameter and Variables

### Designer Overview

The Designer has tools to help you build mappings and maplets so you can specify how to move and transform data between sources and targets. The Designer helps you create source definitions, target definitions, and transformations to build the mappings.

The Designer lets you work with multiple tools at one time and to work in multiple folders and repositories at the same time. It also includes windows so you can view folders, repository objects, and tasks. These tools play an important role in creating the mapping and converting the Business requirements into an ETL code.

You can configure general Designer settings, such as background color and font. You can also configure specific tool settings for each Designer tool.

The Designer provides the following tools:

- **Source Analyzer.** Import or create source definitions for flat file, XML, COBOL, Application, and relational sources.
- **Warehouse Designer.** Import or create target definitions.
- **Transformation Developer.** Used to create reusable transformations.
- **Maplet Designer.** Used to create maplets.
- **Mapping Designer.** Used to create mappings.

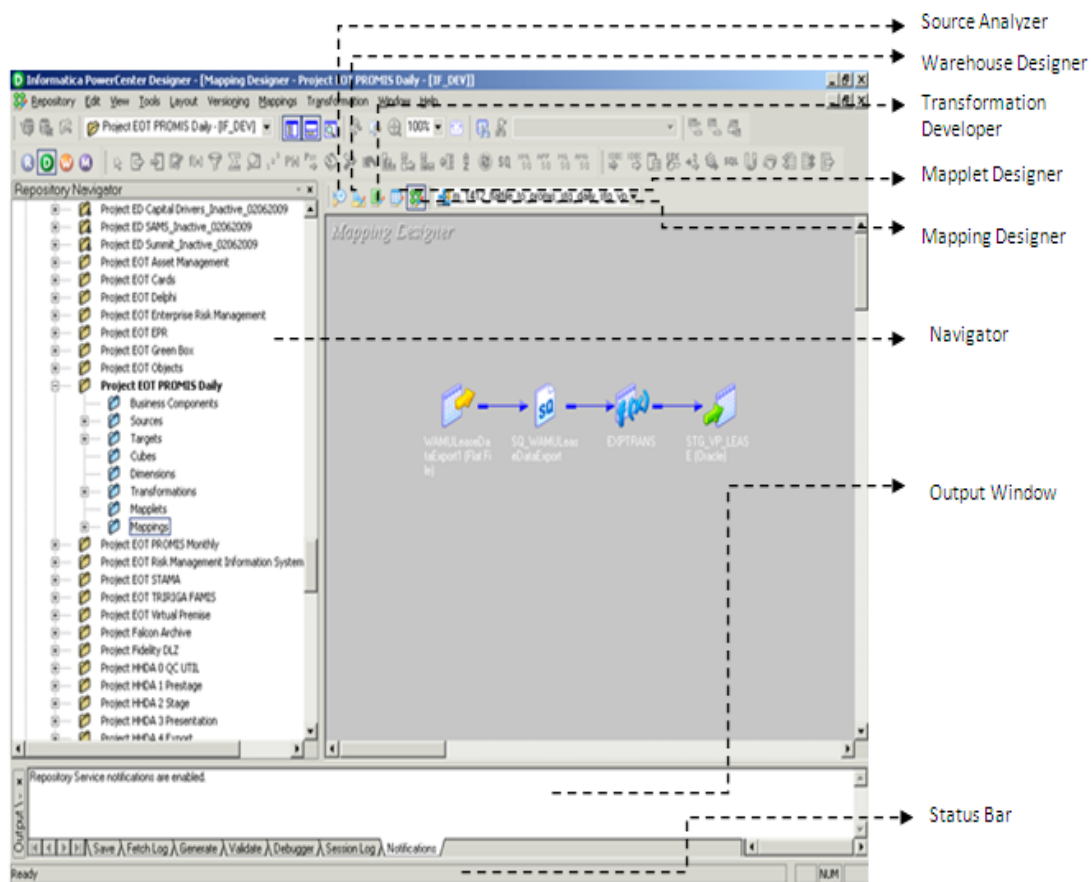
The Designer consists of the following windows:

- **Navigator.** Use to connect to and work in multiple repositories and folders. You can also copy and delete objects and create shortcuts using the Navigator.
- **Workspace.** Use to view or edit sources, targets, maplets, transformations, and mappings. You work with a single tool at a time in the workspace, which has two formats: default and workbook. You can view multiple versions of an object in the workspace.
- **Status bar.** Displays the status of the operation you perform.
- **Output.** Provides details when you perform certain tasks, such as saving work or validating a mapping. Right-click the Output window to access window options, such as printing output text, saving text to file, and changing the font size.
- **Overview.** An optional window to simplify viewing workbooks containing large mappings or a large number of objects. Outlines the visible area in the workspace and highlights selected objects in color. To open the Overview window, click View > Overview Window.
- **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.

You can complete the following tasks in each of the Designer tools:

- Add a repository.
- Print the workspace.
- View date and time an object was last saved.
- Open and close a folder.

- Create shortcuts.
- Check out and check in repository objects.
- Search for repository objects.
- Enter descriptions for repository objects.
- View older versions of objects in the workspace.
- Revert to a previously saved object version.
- Copy objects.
- Export and import repository objects.
- Work with multiple objects, ports, or columns.
- Rename ports.
- Use shortcut keys.
- You can also view object dependencies in the Designer.
- Comparison of various objects can also be done here.



## Source Analyzer

Source Analyzer can import or create source definitions for flat file, XML, COBOL, ERP, and relational sources. Specifies where the data is being extracted from. The Sources can be database objects (tables, views, and synonyms), flat files, XML files, or COBOL files that provide source data. Reads, analyzes, and "reverse engineers" schema information of operational databases and flat files. Sources store metadata information in the repository.

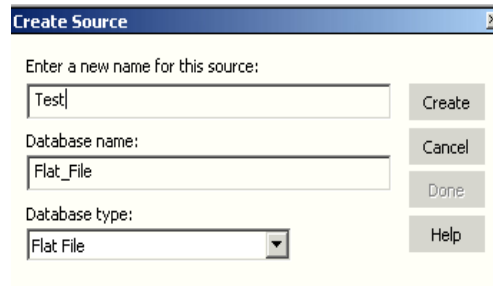
To extract data from a source, you must first define sources in the repository. Source definitions can be created or imported from the database. Mainly Flat File Source definitions are created in the Source analyzer with the structure of the Flat file.

You can import or create the following types of source definitions in the Source Analyzer:

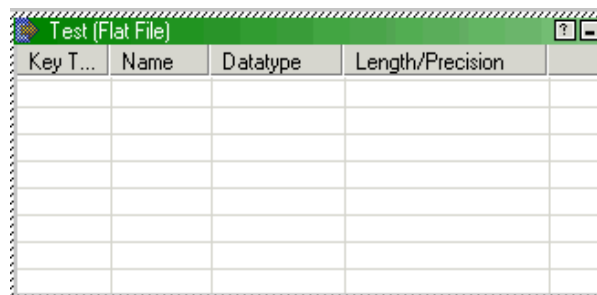
- Relational Tables, Views, and Synonyms
- Fixed-width and delimited flat files that do not contain binary data
- COBOL files
- XML files
- Data models using certain data modeling tools through Metadata Exchange for Data

*To Create the Source Definitions:*

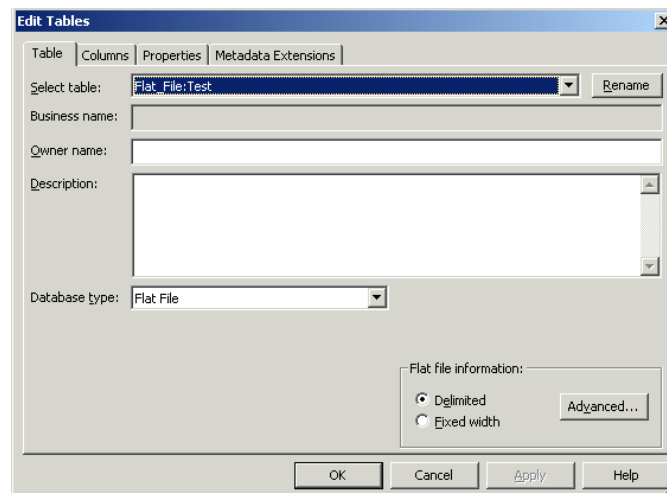
1. Click Sources > Create. Select the Database type and enter the name for Source.



2. Click Create > Done.



3. Double click the Source or Right-Click > Edit to configure the properties of the Source definition.

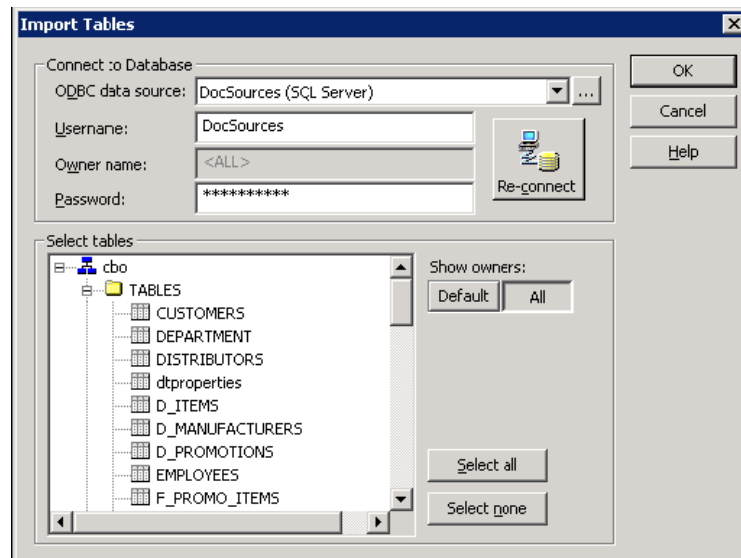


4. Select Columns Tab to define the Field Name, Data Type and Length of the fields.
5. Click OK.

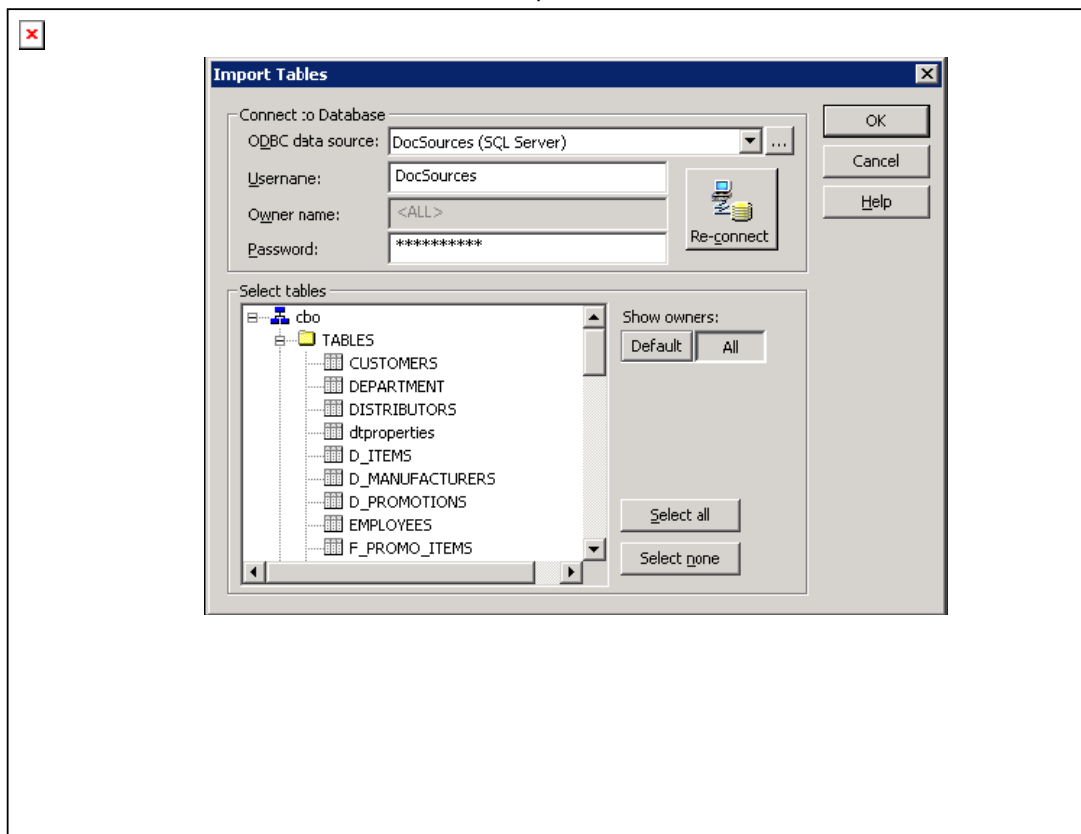
*To import the Source definitions of Relational Tables, Views or Synonyms:*

1. In the Designer, click Tools > Source Analyzer to open the Source Analyzer.
2. Click Sources > Import from Database.

3. Select the ODBC data source to access the database containing the source tables.
4. Enter the user name and password to connect to this database. Also, enter the name of the source table owner, if necessary.
5. Click Connect.



6. In the Select tables list, expand the database owner and the TABLES heading. If you click the All button, you can see all tables in the source database.
7. Select the tables you want to import it into the repository.
8. Click OK to import the source definitions into the repository.
9. The Designer displays the newly imported sources in the workspace. Click Layout > Scale to Fit to fit all the definitions in the workspace.



A new database definition (DBD) node appears under the Sources node in the project 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 DBD node, the list of all the imported sources is displayed.

You can import fixed-width and delimited flat file definitions that do not contain binary data. When importing the definition, the file must be in a directory local to the client machine. In addition, the Integration Service must be able to access all source files during the session.

When you create a file source, target, or lookup definition, you must define the properties of the file. The Flat File Wizard prompts you for the following file properties:

- File name and location
- File code page
- File type
- Column names and data types
- Number of header rows in the file
- Column size and null characters for fixed-width files
- Delimiter types, quote character, and escape character for delimited files

Every mapping requires at least one source qualifier that determines how the Integration Service reads the source data:

- **Source Qualifier transformation.** Used for Relational and Flat file sources.
- **Normalizer transformation.** Used for COBOL sources.
- **Application Source Qualifier transformation.** Used for Application sources.
- **XML Source Qualifier transformation.** Used for XML sources.

You can let the Designer create the source qualifier by default, so each time you drag a source into a mapping; the Designer adds a source qualifier and connects it to the source. Use the automatic source qualifier creation when you want to create one source qualifier for each source in the mapping. You can disable the automatic creation when you want to join data from different relational sources. You can then manually create and connect it to the source.

When you edit the source in the Source Analyzer, all instances of the source in mappings inherit the changes. Some changes might invalidate the mappings using the source.

## Warehouse Designer

---

Warehouse Designer helps to import or create target definitions for file and relational sources. Targets are the output destinations where the data will be written to. This can be database objects, flat files, or XML files to receive transformed data. During a session, the Informatica Server writes the transformed data to targets. Target definitions can be imported in the same way as source definitions in the Warehouse Designer.

You can create the following types of target definitions in the Warehouse Designer:

- **Relational.** Create a relational target for a particular database platform. Create a relational target definition when you want to use an external loader to the target database.
- **Flat file.** Create fixed-width and delimited flat file target definitions.
- **XML file.** Create an XML target definition to output data to an XML file.

You can create target definitions in the following ways:

- **Manually create a target definition.** Create a target definition in the Warehouse Designer.
- **Import the definition for an existing target.** Import the target definition from a relational target or a flat file. The Warehouse Designer uses a Flat File Wizard to import flat files.
- **Create a target definition based on a source definition.** Drag a source definition into the Warehouse Designer to make a target definition.
- **Create a target definition based on a transformation or maplet.** Drag a transformation into the Warehouse Designer to make a target definition.



- **Design several related targets.** Create several related target definitions at the same time. You can create the overall relationship, called a schema, and the target definitions, through wizards in the Designer. The Cubes and Dimensions Wizards follow common principles of data warehouse design to simplify the process of designing related targets.

## Transformation Developer

A transformation generates, modifies, or passes data through ports that you connect in a mapping or maplet. When you build a mapping or maplet, you add transformations and configure them to handle data according to your business purpose. The Designer provides a set of transformations that perform specific functions. Transformations in a mapping represent the operations the Informatica Server performs on data. Data passes into and out of transformations through ports that you connect in a mapping or maplet. Transformations can be active or passive.

When you build a mapping, you add transformations and configure them to handle data according to a business purpose. Complete the following tasks to incorporate a transformation into a mapping:

- **Create the transformation.** Create it in the Mapping Designer as part of a mapping, in the Maplet Designer as part of a maplet, or in the Transformation Developer as a reusable transformation.
- **Configure the transformation.** Each type of transformation has a unique set of options that you can configure.
- **Link the transformation to other transformations and target definitions.** Drag one port to another to link them in the mapping or maplet.

You can create transformations using the following Designer tools:

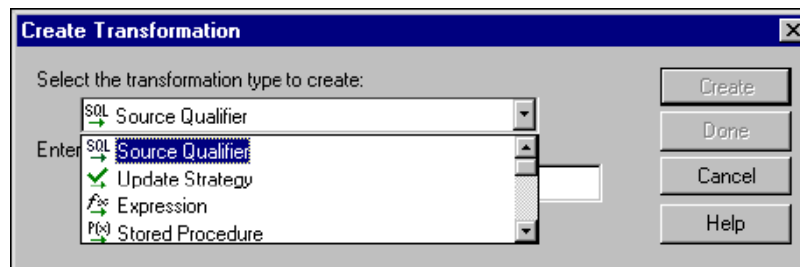
- **Mapping Designer.** Create transformations that connect sources to targets. These transformations created in a mapping cannot be used in other mappings unless you promote them to be reusable.
- **Transformation Developer.** Create reusable transformations that can be used in multiple mappings.
- **Maplet Designer.** Create and configure a set of transformations, called maplets that you use in multiple mappings. Use the same process to create a transformation in the Mapping Designer, Transformation Developer, and Maplet Designer.

To create a transformation:

1. Open the appropriate Designer tool.
2. On the Transformations toolbar, click the button corresponding to the transformation you want to create.

-Or-

Click Transformation > Create and select the type of transformation you want to create.



3. Drag across the workspace where you want to place the transformation. The new transformation appears in the workspace.

## Configuring Transformations

After you create a transformation, you can configure it. Every transformation contains the following common tabs:

- **Transformation.** Name the transformation or add a description.
- **Port.** Add and configure ports.
- **Properties.** Configure properties that are unique to the transformation.
- **Metadata Extensions.** Extend the metadata in the repository by associating information with individual objects in the repository.

Some transformations might include other tabs, such as the Condition tab, where you enter conditions in a Joiner or Normalizer transformation.

When you configure transformations, you might complete the following tasks:

- **Add ports.** Define the columns of data that move into and out of the transformation.
- **Add groups.** In some transformations, define input or output groups that define a row of data entering or leaving the transformation.
- **Enter expressions.** Enter SQL-like expressions in some transformations that transform the data.
- **Define local variables.** Define local variables in some transformations that temporarily store data.
- **Override default values.** Configure default values for ports to handle input nulls and output transformation errors.
- **Enter tracing levels.** Choose the amount of detail the Integration Service writes in the session log about a transformation.

## Source Qualifier Transformation

When you add a relational or a flat file source definition to a mapping, you need to connect it to a Source Qualifier transformation. The Source Qualifier transformation represents the rows that the Integration Service reads when it runs a session.

Use the Source Qualifier transformation to complete the following tasks:

- **Join data originating from the same source database.** You can join two or more tables with primary key-foreign key relationships by linking the sources to one Source Qualifier transformation.
- **Filter rows when the Integration Service reads source data.** If you include a filter condition, the Integration Service adds a WHERE clause to the default query.
- **Specify an outer join rather than the default inner join.** If you include a user-defined join, the Integration Service replaces the join information specified by the metadata in the SQL query.
- **Specify sorted ports.** If you specify a number for sorted ports, the Integration Service adds an ORDER BY clause to the default SQL query.
- **Select only distinct values from the source.** If you choose Select Distinct, the Integration Service adds a SELECT DISTINCT statement to the default SQL query.
- **Create a custom query to issue a special SELECT statement for the Integration Service to read source data.** For example, you might use a custom query to perform aggregate calculations.

**Edit Transformations**

Transformation Ports Properties Sources Metadata Extensions

Select transformation: **SQ SQ\_CUSTOMERS**

Transformation type: Source Qualifier

Transformation Attribute	Value
Sql Query	
User Defined Join	
Source Filter	
Number Of Sorted Ports	0
Tracing Level	Normal
Select Distinct	<input type="checkbox"/>
Pre SQL	
Post SQL	
Output is deterministic	<input checked="" type="checkbox"/>
Output is repeatable	Always

**Output is deterministic**

The source data is changing during the session run.

OK Cancel Apply Help

## Expression Transformation

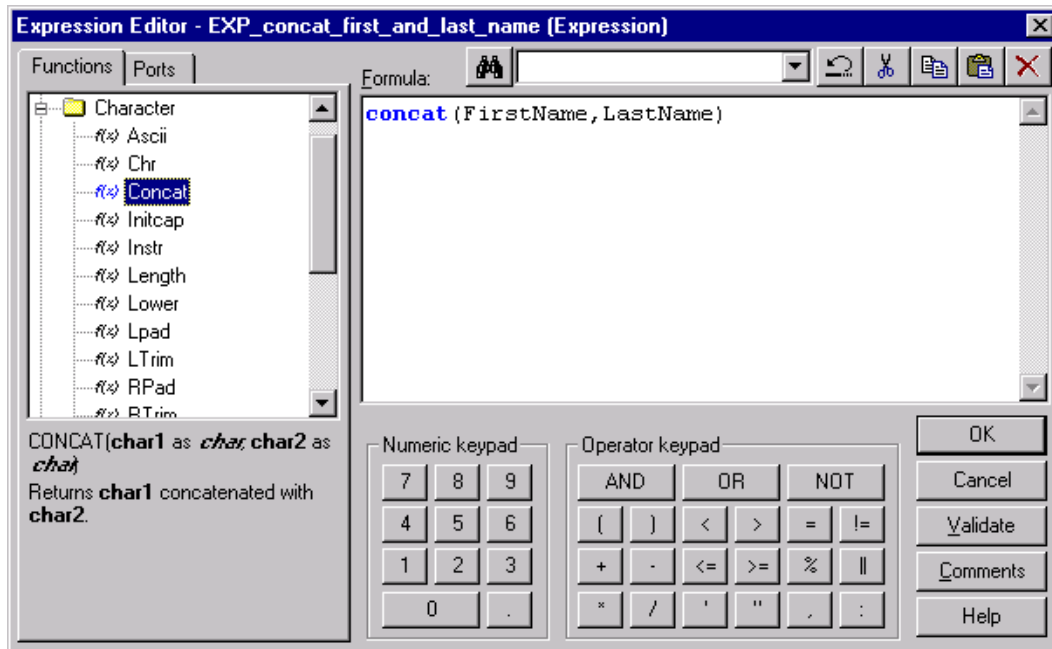
The Expression transformation is used to calculate values in a single row before you write to the target. For example, you might need to adjust employee salaries, concatenate first and last names, or convert strings to numbers. You use the Expression transformation to perform any non-aggregate calculations. You can also use the Expression transformation to test conditional statements before you output the results to target tables or other transformations.

The Following types of ports are used to perform the single row functions.

- **Input Ports or Input/Output Ports for each value used in the calculation.** For example, when calculating the total price for an order, determined by multiplying the unit price by the quantity ordered, one port provides the unit price and the other provides the quantity ordered.
- **Output port for the expression.** You enter the expression as a configuration option for the output port. The return value for the output port needs to match the return value of the expression. Expressions use the transformation language, which includes SQL-like functions, to perform calculations.
- **Local Variables port.** You can reference variables in an expression or use them to temporarily store data. Variables are an easy way to improve performance.

You might use variables to complete the following tasks:

- Temporarily store data.
- Simplify complex expressions.
- Store values from prior rows.
- Capture multiple return values from a stored procedure.
- Compare values.
- Store the results of an unconnected Lookup transformation.



### Adding Multiple Calculations

You can enter multiple expressions in a single Expression transformation. As long as you enter only one expression for each output port, you can create any number of output ports in the transformation. In this way, use one Expression transformation rather than creating separate transformations for each calculation that requires the same set of data.

For example, you might want to calculate several types of withholding taxes from each employee paycheck, such as local and federal income tax, Social Security and Medicare. Since all of these calculations require the employee salary, the withholding category, and/or the corresponding tax rate, you can create one Expression transformation with the salary and withholding category as input/output ports and a separate output port for each necessary calculation.

### Filter Transformation

Use the Filter transformation to filter out rows in a mapping. As an active transformation, the Filter transformation may change the number of rows passed through it. The Filter transformation allows rows that meet the specified filter condition to pass through. It drops rows that do not meet the condition. You can filter data based on one or more conditions.

A filter condition returns TRUE or FALSE for each row that the Integration Service evaluates, depending on whether a row meets the specified condition. For each row that returns TRUE, the Integration Services pass through the transformation. For each row that returns FALSE, the Integration Service drops and writes a message to the session log.

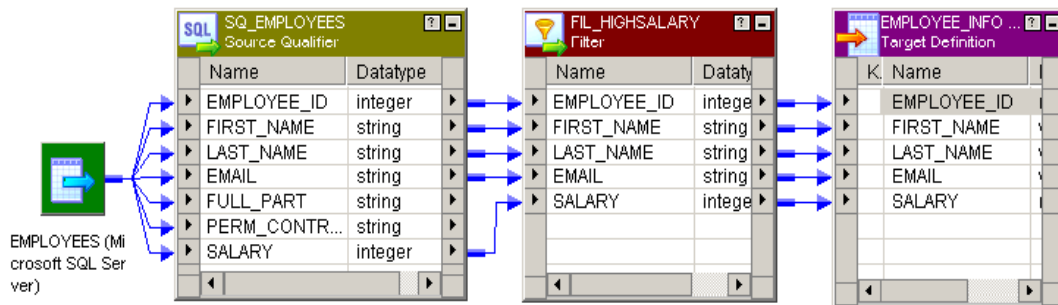
You cannot concatenate ports from more than one transformation into the Filter transformation. The input ports for the filter must come from a single transformation.

A Filter transformation contains the following tabs:

- **Transformation.** Enter the name and description of the transformation. The naming convention for a Filter transformation is FIL\_TransformationName. You can also make the transformation reusable.
- **Ports.** Create and configure ports.
- **Properties.** Configure the filter condition to filter rows. Use the Expression Editor to enter the filter condition. You can also configure the tracing level to determine the amount of transaction detail reported in the session log file.
- **Metadata Extensions.** Create a non-reusable metadata extension to extend the metadata of the transformation. Configure the extension name, data type, precision, and value. You

can also promote metadata extensions to reusable extensions if you want to make it available to all transformation transformations.

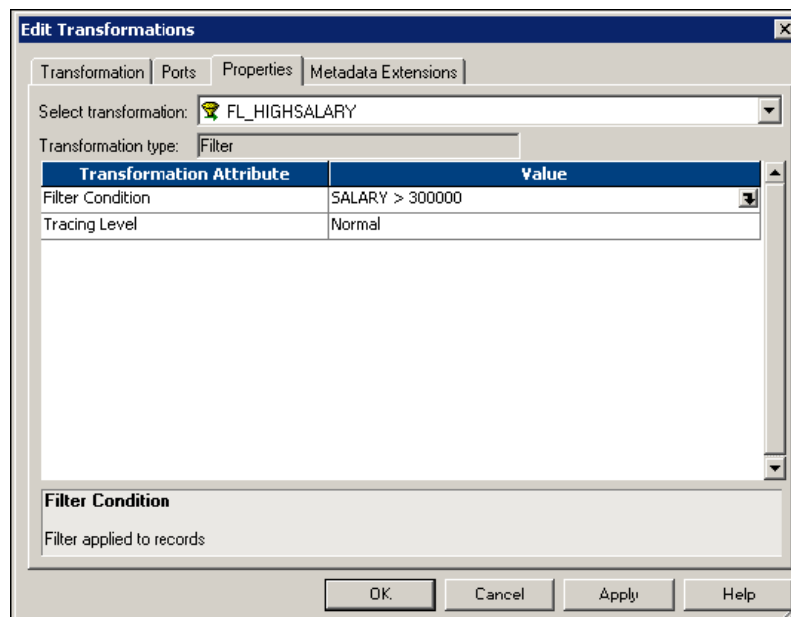
The Filter transformation used in the below mapping allows rows for employees that make salaries of \$30,000 or higher to pass through.



**Note:** Place the Filter transformation as close to the sources in the mapping as possible to maximize session performance. Rather than passing rows you plan to discard through the mapping, you can filter out unwanted data early in the flow of data from sources to targets.

### Specifying a Filter Condition in a Filter Transformation

The filter condition is an expression that returns TRUE or FALSE. Enter conditions using the Expression Editor available on the Properties tab. Any expression that returns a single value can be used as a filter.



For example, if you want to filter *out* rows for employees whose salary is less than \$30,000, you enter the following condition:

**SALARY > 30000**

You can specify multiple components of the condition, using the AND and OR logical operators. If you want to filter out employees who make less than \$30,000 and more than \$100,000, you enter the following condition:

**SALARY > 30000 AND SALARY < 100000**

You can also enter a constant for the filter condition. The numeric equivalent of FALSE is zero (0). Any non-zero value is the equivalent of TRUE. For example, the transformation contains a port

named NUMBER\_OF\_UNITS with a numeric datatype. You configure a filter condition to return FALSE if the value of NUMBER\_OF\_UNITS equals zero. Otherwise, the condition returns TRUE.

You do not need to specify TRUE or FALSE as values in the expression. TRUE and FALSE are implicit return values from any condition you set. If the filter condition evaluates to NULL, the row is treated as FALSE.

**Note:** The filter condition is case sensitive.

### Filtering Rows with Null Values

To filter rows containing null values or spaces, use the ISNULL and IS\_SPACES functions to test the value of the port. For example, if you want to filter out rows that contain NULL value in the FIRST\_NAME port, use the following condition:

*IIF (ISNULL (FIRST\_NAME), FALSE, TRUE)*

This condition states that if the FIRST\_NAME port is NULL, the return value is FALSE and the row should be discarded. Otherwise, the row passes through to the next transformation.

Use the following guidelines to efficiently use the filter transformation:

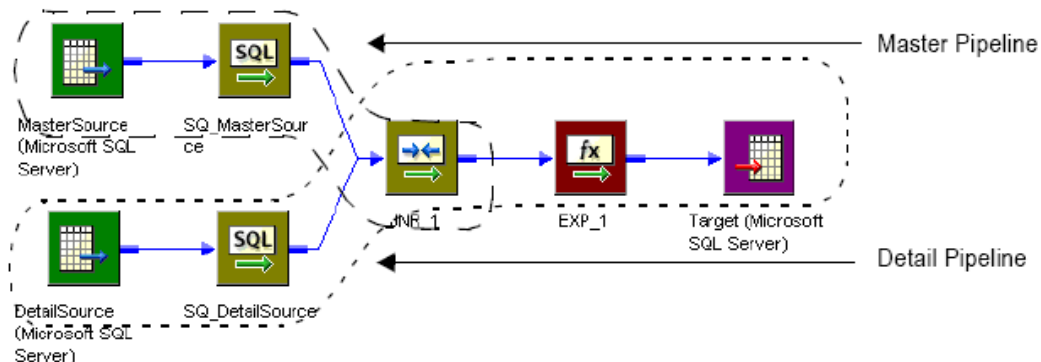
- **Use the Filter transformation early in the mapping.** To maximize session performance, keep the Filter transformation as close as possible to the sources in the mapping. Rather than passing rows that you plan to discard through the mapping, you can filter out unwanted data early in the flow of data from sources to targets.
- **Use the Source Qualifier transformation to filter.** The Source Qualifier transformation provides an alternate way to filter rows. Rather than filtering rows from within a mapping, the Source Qualifier transformation filters rows when read from a source. The main difference is that the source qualifier limits *the row set extracted from a source*, while the Filter transformation limits *the row set sent to a target*. Since a source qualifier reduces the number of rows used throughout the mapping, it provides better performance.

However, the Source Qualifier transformation only lets you filter rows from relational sources, while the Filter transformation filters rows from any type of source. Also, note that since it runs in the database, you must make sure that the filter condition in the Source Qualifier transformation only uses standard SQL. The Filter transformation can define a condition using any statement or transformation function that returns either a TRUE or FALSE value.

### Joiner Transformation

Use the Joiner transformation to join source data from two related heterogeneous sources residing in different locations or file systems. You can also join data from the same source. The Joiner transformation joins sources with at least one matching column. The Joiner transformation uses a condition that matches one or more pairs of columns between the two sources.

The two input pipelines include a master pipeline and a detail pipeline or a master and a detail branch. The master pipeline ends at the Joiner transformation, while the detail pipeline continues to the target.



To join more than two sources in a mapping, join the output from the Joiner transformation with another source pipeline. Add Joiner transformations to the mapping until you have joined all the source pipelines. The Joiner transformation accepts input from most transformations.

However, consider the following limitations on the pipelines you connect to the Joiner transformation:

- You cannot use a Joiner transformation when either input pipeline contains an Update Strategy transformation.
- You cannot use a Joiner transformation if you connect a Sequence Generator transformation directly before the Joiner transformation.

### Working with the Joiner Transformation

When you work with the Joiner transformation, you must configure the transformation properties, join type, and join condition. You can configure the Joiner transformation for sorted input to improve Integration Service performance. You can also configure the transformation scope to control how the Integration Service applies transformation logic. To work with the Joiner transformation, complete the following tasks:

- **Configure the Joiner transformation properties.** Properties for the Joiner transformation identify the location of the cache directory, how the Integration Service processes the transformation, and how the Integration Service handles caching.
- **Configure the join condition.** The join condition contains ports from both input sources that must match for the Integration Service to join two rows. Depending on the type of join selected, the Integration Service either adds the row to the result set or discards the row.
- **Configure the join type.** A join is a relational operator that combines data from multiple tables in different databases or flat files into a single result set. You can configure the Joiner transformation to use a Normal, Master Outer, Detail Outer, or Full Outer join type.
- **Configure the session for sorted or unsorted input.** You can improve session performance by configuring the Joiner transformation to use sorted input. To configure a mapping to use sorted data, you establish and maintain a sort order in the mapping so that the Integration Service can use the sorted data when it processes the Joiner transformation.
- **Configure the transaction scope.** When the Integration Service processes a Joiner transformation, it can apply transformation logic to all data in a transaction, all incoming data, or one row of data at a time.

Transformation Attribute	Value
Case Sensitive String Comparison	<input checked="" type="checkbox"/>
Cache Directory	\$PMCacheDir
Join Condition	ITEM_NAME1 = ITEM_NAME
Join Type	Normal Join
Null ordering in master	Null Is Highest Value
Null ordering in detail	Null Is Highest Value
Tracing Level	Normal
Joiner Data Cache Size	2000000
Joiner Index Cache Size	1000000
Sorted Input	<input checked="" type="checkbox"/>
Transformation Scope	All Input

**Sorted Input**  
Defines whether the input data are sorted.



## Defining a Join Condition

The join condition contains ports from both input sources that must match for the Integration Service to join two rows. Depending on the type of join selected, the Integration Service either adds the row to the result set or discards the row. The Joiner transformation produces result sets based on the join type, condition, and input data sources.

During a session, the Integration Service compares each row of the master source against the detail source. To improve performance for an unsorted Joiner transformation, use the source with fewer rows as the master source. To improve performance for a sorted Joiner transformation, use the source with fewer duplicate key values as the master.

You define one or more conditions based on equality between the specified master and detail sources.

For example, if two sources with tables called EMPLOYEE\_AGE and EMPLOYEE\_POSITION both contain employee ID numbers, the following condition matches rows with employees listed in both sources:

*EMP\_ID1 = EMP\_ID2*

The order of the ports in the condition can impact the performance of the Joiner transformation. If you use multiple ports in the join condition, the Integration Service compares the ports in the order you specify.

The Designer validates datatypes in a condition. Both ports in a condition must have the same datatype. If you need to use two ports in the condition with non-matching datatypes, convert the datatypes so they match.

Use the following guidelines when you define join conditions:

- The ports you use in the join condition must match the ports at the sort origin.
- When you configure multiple join conditions, the ports in the first join condition must match the first ports at the sort origin.
- When you configure multiple conditions, the order of the conditions must match the order of the ports at the sort origin, and you must not skip any ports.
- The number of sorted ports in the sort origin can be greater than or equal to the number of ports at the join condition.

## Null Handling

The Joiner transformation does not match null values. For example, if both EMP\_ID1 and EMP\_ID2 contain a row with a null value, the Integration Service does not consider them a match and does not join the two rows. To join rows with null values, replace null input with default values, and then join on the default values. If a result set includes fields that do not contain data in either of the sources, the Joiner transformation populates the empty fields with null values. If you know that a field will return a NULL and you do not want to insert NULLs in the target, you can set a default value on the Ports tab for the corresponding port.

## Join Type

You define the join type on the Properties tab in the transformation. The Joiner transformation supports the following types of joins:

- **Normal Join.** With a normal join, the Integration Service discards all rows of data from the master and detail source that do not match, based on the condition.
- **Master Outer Join.** A master outer join keeps all rows of data from the detail source and the matching rows from the master source. It discards the unmatched rows from the master source.
- **Detail Outer Join.** A detail outer join keeps all rows of data from the master source and the matching rows from the detail source. It discards the unmatched rows from the detail source.
- **Full Outer Join.** A full outer join keeps all rows of data from both the master and detail sources.

**Note:** A normal or master outer join performs faster than a full outer or detail outer join.



## Using Sorted Input

When you configure the Joiner transformation to use sorted data, the Integration Service improves performance by minimizing disk input and output. To configure a mapping to use sorted data, you establish and maintain a sort order in the mapping so the Integration Service can use the sorted data when it processes the Joiner transformation.

If you pass unsorted or incorrectly sorted data to a Joiner transformation configured to use sorted data, the session fails and the Integration Service logs the error in the session log file.

## Adding Joiner Transformations to the Mapping

When you add transformations between the sort origin and the Joiner transformation, use the following guidelines to maintain sorted data:

- Do not place any of the following transformations between the sort origin and the Joiner transformation:
  - » Custom
  - » Unsorted Aggregator
  - » Normalizer
  - » Rank
  - » Union transformation
  - » XML Parser transformation
  - » XML Generator transformation
  - » Maplet, if it contains one of the above transformations
- You can place a sorted Aggregator transformation between the sort origin and the Joiner transformation if you use the following guidelines:
  - » Configure the Aggregator transformation for sorted input.
  - » Use the same ports for the group by columns in the Aggregator transformation as the ports at the sort origin.
  - » The group by ports must be in the same order as the ports at the sort origin.
- When you join the result set of a Joiner transformation with another pipeline, verify that the data output from the first Joiner transformation is sorted.

**Tip:** You can place the Joiner transformation directly after the sort origin to maintain sorted data.

## Rank

Rank transformation allows you to select only the top or bottom rank of data with. Use a Rank transformation to return the largest or smallest numeric value in a port or group. You can also use a Rank transformation to return the strings at the top or the bottom of a session sort order. During the session, the Integration Service caches input data until it can perform the rank calculations.

The Rank transformation differs from the transformation functions MAX and MIN, in that it lets you select a group of top or bottom values, not just one value.

### Rank Caches

During a session, the Integration Service compares an input row with rows in the data cache. If the input row out-ranks a cached row, the Integration Service replaces the cached row with the input row. If you configure the Rank transformation to rank across multiple groups, the Integration Service ranks incrementally for each group it finds.

The Integration Service stores group information in an index cache and row data in a data cache. If you create multiple partitions in a pipeline, the Integration Service creates separate caches for each partition.

### Rank Transformation Properties

When you create a Rank transformation, you can configure the following properties:

- Enter a cache directory.
- Select the top or bottom rank.

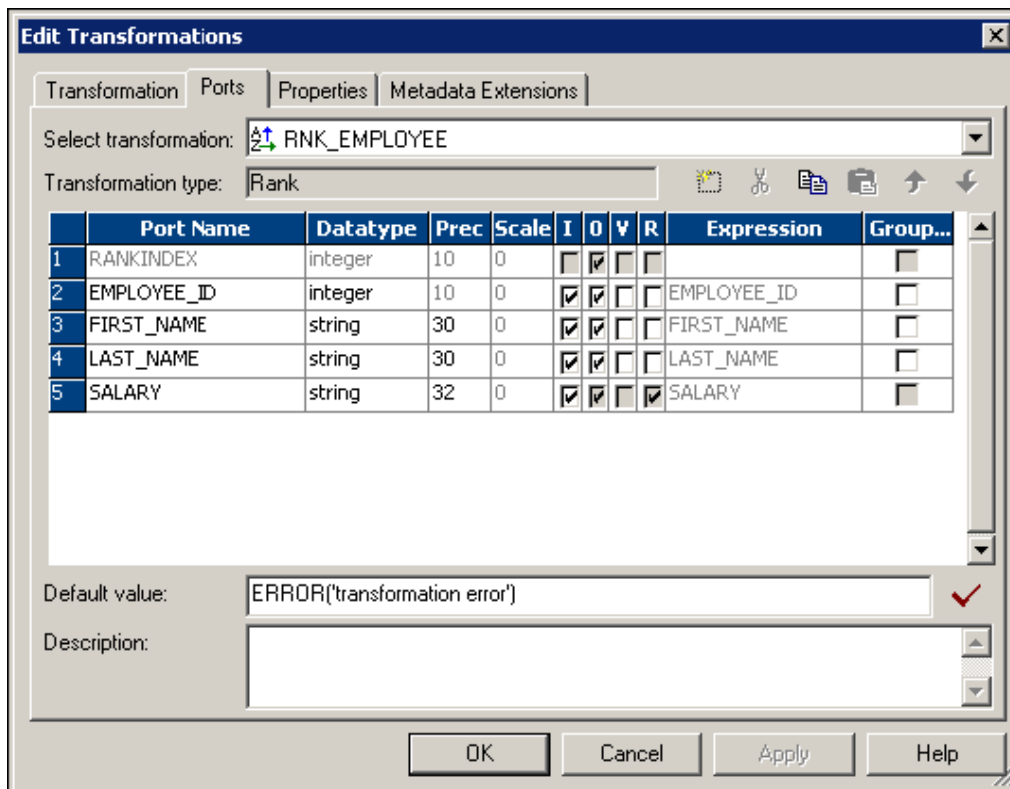
- Select the input/output port that contains values used to determine the rank. You can select only one port to define a rank.
- Select the number of rows falling within a rank.
- Define groups for ranks, such as the 10 least expensive products for each manufacturer.

### Ports in a Rank Transformation

The Rank transformation includes input or input/output ports connected to another transformation in the mapping. It also includes variable ports and a rank port. Use the rank port to specify the column you want to rank.

### Rank Index

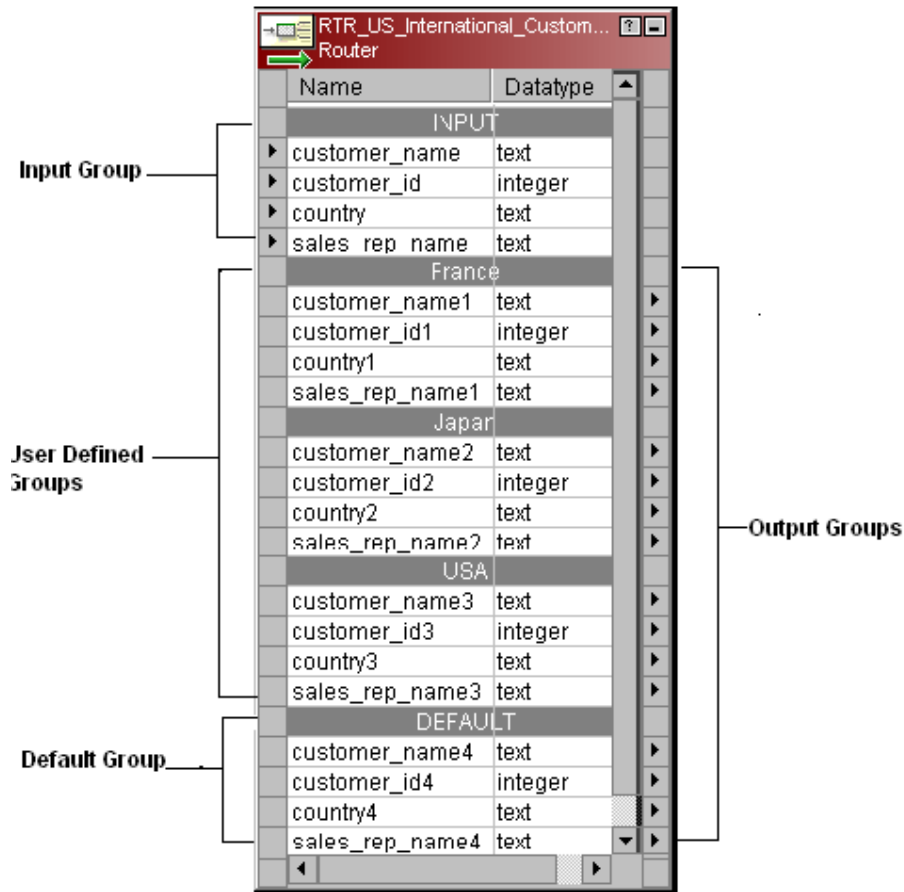
The Designer creates a RANKINDEX port for each Rank transformation. The Integration Service uses the Rank Index port to store the ranking position for each row in a group.



### Router Transformation

A Router transformation is similar to a Filter transformation because both transformations allow you to use a condition to test data. A Filter transformation tests data for one condition and drops the rows of data that do not meet the condition. However, a Router transformation tests data for one or more conditions and gives you the option to route rows of data that do not meet any of the conditions to a default output group.

If you need to test the same input data based on multiple conditions, use a Router transformation in a mapping instead of creating multiple Filter transformations to perform the same task. The Router transformation is more efficient. For example, to test data based on three conditions, you only need one Router transformation instead of three filter transformations to perform this task. Likewise, when you use a Router transformation in a mapping, the Integration Service processes the incoming data only once. When you use multiple Filter transformations in a mapping, the Integration Service processes the incoming data for each transformation.



A Router transformation has the following types of groups:

- **Input.** The Designer copies property information from the input ports of the input group to create a set of output ports for each output group.
- **Output.** You cannot modify or delete output ports or their properties.

There are two types of output groups:

- **User-defined groups.** You create a user-defined group to test a condition based on incoming data. A user-defined group consists of output ports and a group filter condition. You can create and edit user-defined groups on the Groups tab with the Designer. Create one user-defined group for each condition that you want to specify. The Integration Service uses the condition to evaluate each row of incoming data. It tests the conditions of each user-defined group before processing the default group, based on the order of the connected output groups. The Integration Service processes user-defined groups that are connected to a transformation or a target in a mapping. The Integration Service only processes user-defined groups that are not connected in a mapping if the default group is connected to a transformation or a target. If a row meets more than one group filter condition, the Integration Service passes this row multiple times.
- **Default group.** The Designer creates the default group after you create one new user-defined group. The Designer does not allow you to edit or delete the default group. This group does not have a group filter condition associated with it. If all of the conditions evaluate to FALSE, the Integration Service passes the row to the default group. If you want the Integration Service to drop all rows in the default group, do not connect it to a transformation or a target in a mapping. The Designer deletes the default group when you delete the last user-defined group from the list.

## Normalizer Transformation

Normalization is the process of organizing data to reduce redundancy. In database terms, this includes creating normalized tables and establishing relationships between the tables eliminating redundancy and inconsistent dependencies.

The Normalizer transformation normalizes records from COBOL and relational sources, allowing you to organize the data. A Normalizer transformation can appear anywhere in a pipeline when you normalize a relational source. Use a Normalizer transformation instead of the Source Qualifier transformation when you normalize a COBOL source. When you drag a COBOL source into the Mapping Designer workspace, the Mapping Designer creates a Normalizer transformation with input and output ports for every column in the source.

You primarily use the Normalizer transformation with COBOL sources, which are often stored in a denormalized format. The OCCURS statement in a COBOL file nests multiple records of information in a single record. Using the Normalizer transformation, you break out repeated data within a record into separate records. For each new record it creates, the Normalizer transformation generates a unique identifier. Use this key value to join the normalized records.

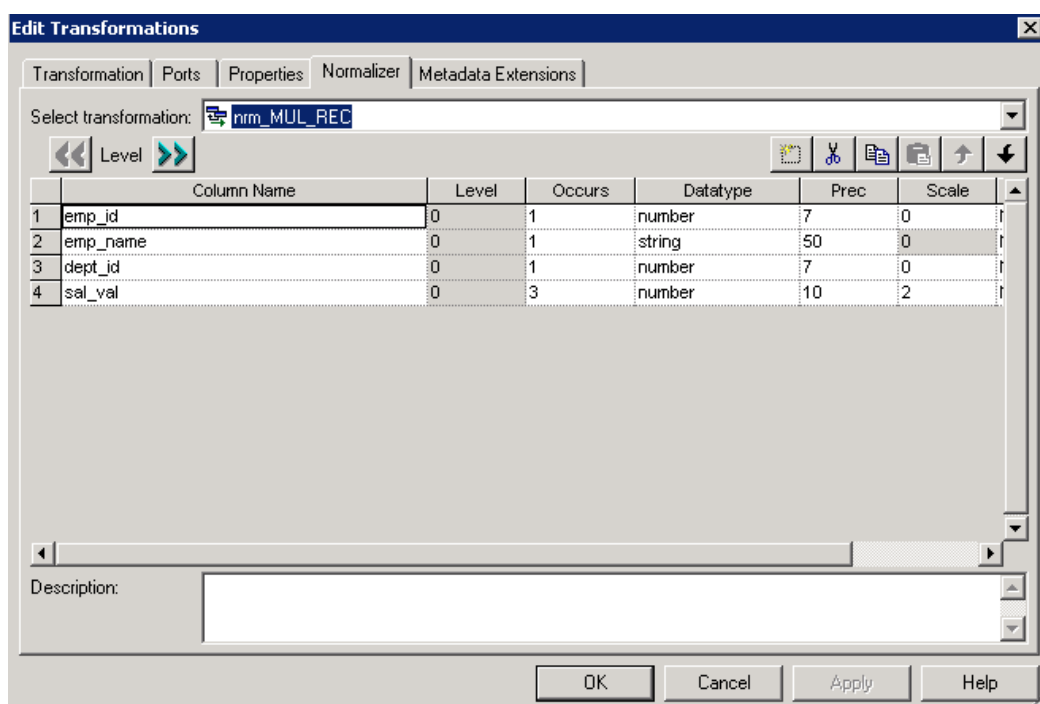
You can also use the Normalizer transformation with relational sources to create multiple rows from a single row of data.

### Normalizing Data in a Mapping

Although the Normalizer transformation is designed to handle data read from COBOL sources, you can also use it to denormalize data from any type of source in a mapping. You can add a Normalizer transformation to any data flow within a mapping to normalize components of a single record that contains denormalized data.

If you have denormalized data for which the Normalizer transformation has created key values, connect the ports representing the repeated data and the output port for the generated keys to a different pipeline branch in the mapping. Ultimately, you may want to write these values to different targets.

Use a single Normalizer transformation to handle multiple levels of denormalization in the same record. For example, a single record might contain two different detail record sets. Rather than using two Normalizer transformations to handle the two different detail record sets, you handle both normalizations in the same transformation.



## Normalizer Ports

When you create a Normalizer for a COBOL source, or in the mapping pipeline, the Designer identifies the OCCURS and REDEFINES statements and generates the following columns:

- **Generated key.** One port for each REDEFINES clause.  
The Designer generates a port for each REDEFINES clause to specify the generated key. Use the generated key as a primary key column in the target table and to create a primary-foreign key relationship. The naming convention for the Normalizer generated key is:  
*GK\_<redefined\_field\_name>*
- **Generated Column ID.** One port for each OCCURS clause.  
The Designer generates a port for each OCCURS clause to specify the positional index within an OCCURS clause. Use the generated column ID to create a primary-foreign key relationship. The naming convention for the Normalizer generated column ID is:  
*GCID\_<occurring\_field\_name>*

Use these ports for primary and foreign key columns. The Normalizer key and column ID columns are also useful when you want to pivot input columns into rows. You cannot delete these ports.

## Look Up Transformation

A Lookup transformation is used in a mapping to look up data in a flat file or a relational table, view, or synonym. You can import a lookup definition from any flat file or relational database to which both the Power Center Client and Integration Service can connect. You can also use multiple Lookup transformations in a mapping.

The Integration Service queries the lookup source based on the lookup ports in the transformation. It compares Lookup transformation port values to lookup source column values based on the lookup condition. It passes the result of the lookup to other transformations or a target.

Use the Lookup transformation to perform many tasks, including:

- **Get a related value.** For example, the source includes employee ID, but you want to include the employee name in the target table to make the summary data easier to read.
- **Perform a calculation.** Many normalized tables include values used in a calculation, such as gross sales per invoice or sales tax, but not the calculated value (such as net sales).
- **Update slowly changing dimension tables.** Use a Lookup transformation to determine whether rows already exist in the target.

On the Properties tab, you can configure properties, such as an SQL override for relational lookups, the lookup source name, and tracing level for the transformation. You can also configure caching properties on the Properties tab.

On the Condition tab, you can enter the condition or conditions you want the Integration Service to use to determine whether input data qualifies values in the lookup source or cache.

You can configure the Lookup transformation to complete the following types of lookups:

- **Connected or unconnected.** Connected and unconnected transformations receive input and send output in different ways.
- **Relational or flat file lookup.** When you create a Lookup transformation, you can choose to perform a lookup on a flat file or a relational table.
- **Cached or Uncached.** Sometimes you can improve session performance by caching the lookup table. If you cache the lookup, you can choose to use a dynamic or static cache. By default, the lookup cache remains static and does not change during the session. With a dynamic cache, the Integration Service inserts or updates rows in the cache during the session. When you cache the target table as the lookup, you can look up values in the target and insert them if they do not exist, or update them if they do.

## Connected and Unconnected Lookups

You can configure a connected Lookup transformation to receive input directly from the mapping pipeline, or you can configure an unconnected Lookup transformation to receive input from the result of an expression in another transformation.

### Connected Lookup Transformation

- A connected Lookup transformation receives input values directly from another transformation in the pipeline.
- For each input row, the Integration Service queries the lookup source or cache based on the lookup ports and the condition in the transformation.
- If the transformation is uncached or uses a static cache, the Integration Service returns values from the lookup query.
- If the transformation uses a dynamic cache, the Integration Service inserts the row into the cache when it does not find the row in the cache. When the Integration Service finds the row in the cache, it updates the row in the cache or leaves it unchanged. It flags the row as insert, update, or no change.
- The Integration Service passes return values from the query to the next transformation. If the transformation uses a dynamic cache, you can pass rows to a Filter or Router transformation to filter new rows to the target.

### Unconnected Lookup Transformation

The following steps describe the way the Integration Service processes an unconnected Lookup transformation:

1. An unconnected Lookup transformation receives input values from the result of a: LKP expression in another transformation, such as an Update Strategy transformation.
2. The Integration Service queries the lookup source or cache based on the lookup ports and condition in the transformation.
3. The Lookup transformation passes the return value into the: LKP expression.
4. The Integration Service returns one value into the return port of the Lookup transformation.
5. You can call the Lookup transformation more than once in a mapping.
6. A common use for unconnected Lookup transformations is to update slowly changing dimension tables.

### Lookup source

A flat file or a relational table could be a lookup source. When you create a Lookup transformation, you can import the lookup source from the following locations:

- Any relational source or target definition in the repository
- Any flat file source or target definition in the repository
- Any table or file that both the Integration Service and PowerCenter Client machine can connect to.

The lookup table can be a single table, or you can join multiple tables in the same database using a lookup SQL override. The Integration Service queries the lookup table or an in-memory cache of the table for all incoming rows into the Lookup transformation. The Integration Service can connect to a lookup table using a native database driver or an ODBC driver. However, the native database drivers improve session performance.

### Ports

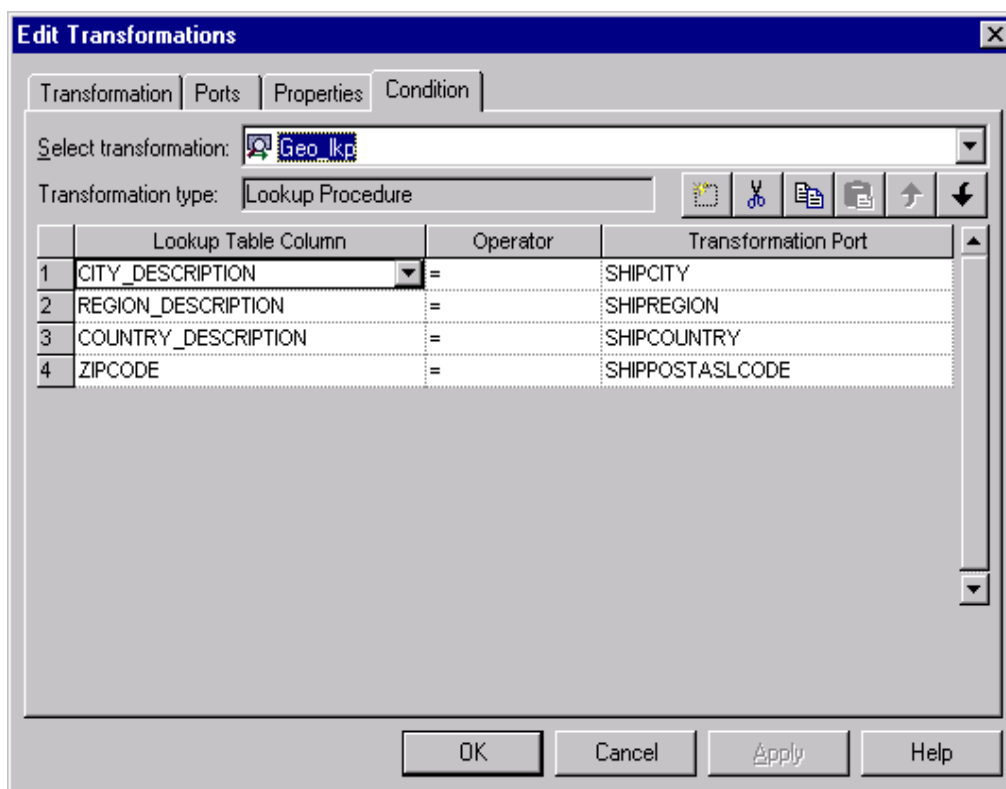
The Ports tab contains options similar to other transformations, such as port name, datatype, and scale. The following are the types of ports associated with a lookup transformation.

- **Input Ports.** Create an input port for each lookup port you want to use in the lookup condition. You must have at least one input or input/output port in each Lookup transformation. Used in both Connected and Unconnected.

- **Output port.** Create an output port for each lookup port you want to link to another transformation. You can designate both input and lookup ports as output ports. For connected lookups, you must have at least one output port. For unconnected lookups, use a lookup/output port as a return port (R) to designate a return value.
- **Lookup port.** The Designer designates each column in the lookup source as a lookup (L) and output port (O).
- **Return port.** Use only in unconnected Lookup transformations. Designates the column of data you want to return based on the lookup condition. You can designate one lookup/output port as the return port. The Lookup transformation also enables an associated ports property that you configure when you use a dynamic cache.

Use the following guidelines to configure lookup ports:

- If you delete lookup ports from a flat file session, the session fails.
- You can delete lookup ports from a relational lookup if you are certain the mapping does not use the lookup port. This reduces the amount of memory the Integration Service uses to run the session.
- To ensure datatypes match when you add an input port, copy the existing lookup ports.



## Sorter Transformation

You can sort data with the Sorter transformation. You can sort data in ascending or descending order according to a specified sort key. You can also configure the Sorter transformation for case-sensitive sorting, and specify whether the output rows should be distinct. The Sorter transformation is an active transformation. It must be connected to the data flow.

You can sort data from relational or flat file sources. You can also use the Sorter transformation to sort data passing through an Aggregator transformation configured to use sorted input.

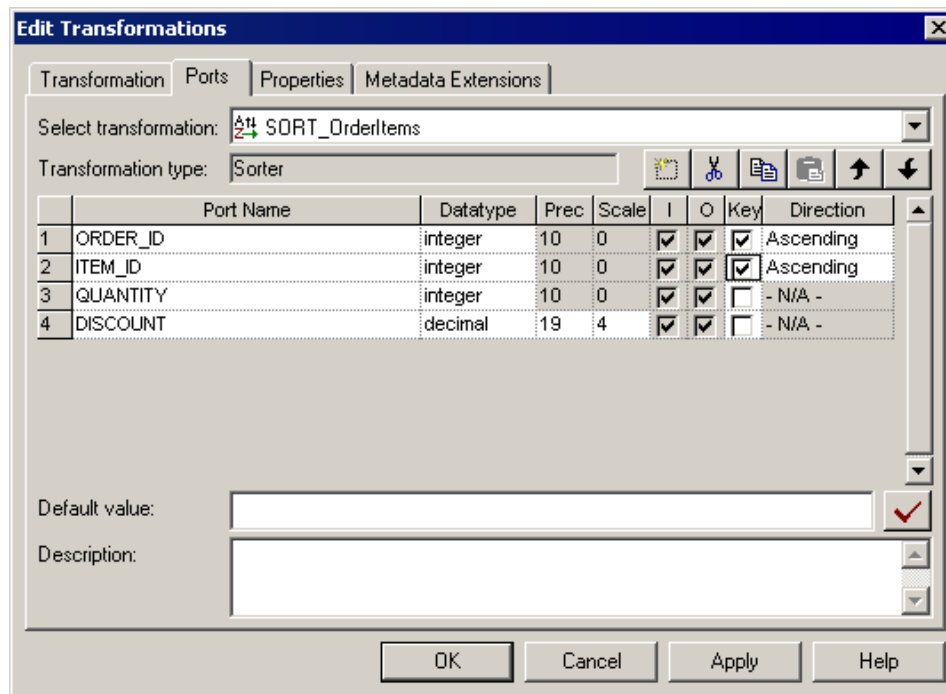
When you create a Sorter transformation in a mapping, you specify one or more ports as a sort key and configure each sort key port to sort in ascending or descending order. You can also configure the sort criteria the Integration Service applies to all sort key ports and the system resources it allocates to perform the sort operation.



The Sorter transformation contains only input/output ports. All data passing through the Sorter transformation is sorted according to a sort key.

You can specify more than one port as part of the sort key. When you specify multiple ports for the sort key, the Integration Service sorts each port sequentially. The order the ports appear in the Ports tab determines the succession of sort operations. The Sorter transformation treats the data passing through each successive sort key port as a secondary sort of the previous port.

At session run time, the Integration Service sorts data according to the sort order specified in the session properties. The sort order determines the sorting criteria for special characters and symbols.



## Aggregator Transformation

The Aggregator transformation lets you perform aggregate calculations, such as averages and sums. The Aggregator transformation is unlike the Expression transformation, in that you use the Aggregator transformation to perform calculations on groups. The Expression transformation permits you to perform calculations on a row-by-row basis only.

When using the transformation language to create aggregate expressions, use conditional clauses to filter rows, providing more flexibility than SQL language. The Integration Service performs aggregate calculations as it reads, and stores necessary data group and row data in an aggregate cache.

To configure ports in the Aggregator transformation, the following tasks need to be completed:

- Enter an expression in any output port, using conditional clauses or non-aggregate functions in the port.
- Create multiple aggregate output ports.
- Configure any input, input/output, output, or variable port as a group by port.
- Improve performance by connecting only the necessary input/output ports to subsequent transformations, reducing the size of the data cache.
- Use variable ports for local variables.
- Create connections to other transformations as you enter an expression.



## Aggregate Expressions

The Designer allows aggregate expressions only in the Aggregator transformation. An aggregate expression can include conditional clauses and non-aggregate functions. It can also include one aggregate function nested within another aggregate function, such as:

*MAX (COUNT (ITEM))*

The result of an aggregate expression varies depending on the group by ports used in the transformation. For example, when the Integration Service calculates the following aggregate expression with no group by ports defined, it finds the total quantity of items sold:

*SUM (QUANTITY)*

However, if you use the same expression, and you group by the ITEM port, the Integration Service returns the total quantity of items sold, by item. You can create an aggregate expression in any output port and use multiple aggregate ports in a transformation.

	Port Name	Datatype	Prec	Scale	I	O	V	Expression	GroupBy
1	ORDER_ID	integer	10	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ORDER_ID	<input type="checkbox"/>
2	STORE_ID	integer	10	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	STORE_ID	<input checked="" type="checkbox"/>
3	ITEM	string	15	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ITEM	<input checked="" type="checkbox"/>
4	QTY	integer	10	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	QTY	<input type="checkbox"/>
5	PRICE	decimal	8	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	PRICE	<input type="checkbox"/>
6	SALES_PER_STORE_ITEM	decimal	12	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	SUM(QTY*PRICE)	<input type="checkbox"/>

Default value: Misc

Description:

OK Cancel Apply Help

## Group By Ports

The Aggregator transformation lets you define groups for aggregations, rather than performing the aggregation across all input data. To define a group for the aggregate expression, select the appropriate input, input/output, output, and variable ports in the Aggregator transformation. You can select multiple groups by ports, creating a new group for each unique combination of groups.

When you group values, the Integration Service produces one row for each group. If you do not group values, the Integration Service returns one row for all input rows. The Integration Service typically returns the last row of each group (or the last row received) with the result of the aggregation. However, if you specify a particular row to be returned (for example, by using the FIRST function), the Integration Service then returns the specified row.

When selecting multiple group by ports in the Aggregator transformation, the Integration Service uses port order to determine the order by which it groups. Since group order can affect the results, order group by ports to ensure the appropriate grouping. For example, the results of grouping by ITEM\_ID then QUANTITY can vary from grouping by QUANTITY then ITEM\_ID, because the numeric values for quantity are not necessarily unique.

## Conditional Clauses

Use conditional clauses in the aggregate expression to reduce the number of rows used in the aggregation. The conditional clause can be any clause that evaluates to TRUE or FALSE. For example, use the following expression to calculate the total commissions of employees who exceeded their quarterly quota:

*SUM (COMMISSION, COMMISSION > QUOTA)*

## Non-Aggregate Functions

You can also use non-aggregate functions in the aggregate expression. The following expression returns the highest number of items sold for each item (grouped by item). If no items were sold, the expression returns 0.

*IIF( MAX( QUANTITY ) > 0, MAX( QUANTITY ), 0)*

## Null Values in Aggregate Functions

When you configure the Integration Service, you can choose how you want the Integration Service to handle null values in aggregate functions. You can choose to treat null values in aggregate functions as NULL or zero. By default, the Integration Service treats null values as NULL in aggregate functions.

## Non-Aggregate Expressions

Use non-aggregate expressions in group by ports to modify or replace groups. For example, if you want to replace 'AAA battery' before grouping, you can create a new group by output port, named CORRECTED\_ITEM, using the following expression:

*IIF (ITEM = 'AAA battery', battery, ITEM)*

## Default Values

Use default values in the group by port to replace null input values. This allows the Integration Service to include null item groups in the aggregation.

## Aggregate Caches

When you run a session that uses an Aggregator transformation, the Integration Service creates index and data caches in memory to process the transformation. If the Integration Service requires more space, it stores overflow values in cache files.

You can configure the index and data caches in the Aggregator transformation or in the session properties. Or, you can configure the Integration Service to determine the cache size at runtime.

## Using Sorted Input

You can improve Aggregator transformation performance by using the sorted input option. When you use sorted input, the Integration Service assumes all data is sorted by group. As the Integration Service reads rows for a group, it performs aggregate calculations. When necessary, it stores group information in memory. To use the Sorted Input option, you must pass sorted data to the Aggregator transformation. You can gain performance with sorted ports when you configure the session with multiple partitions.

When you do not use sorted input, the Integration Service performs aggregate calculations as it reads. However, since data is not sorted, the Integration Service stores data for each group until it reads the entire source to ensure all aggregate calculations are accurate. If you use sorted input and do not presort data correctly, you receive unexpected results.

Do not use sorted input if either of the following conditions are true:

- The aggregate expression uses nested aggregate functions.
- The session uses incremental aggregation.

Use the following guidelines to optimize the performance of an Aggregator transformation.

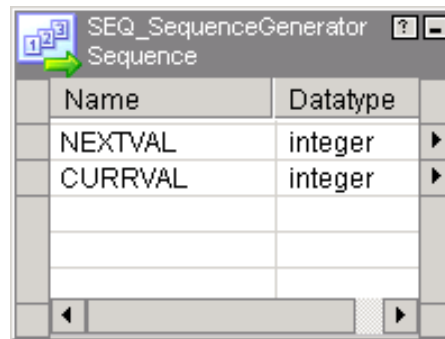
- **Use sorted input to decrease the use of aggregate caches.** Sorted input reduces the amount of data cached during the session and improves session performance. Use this option with the Sorter transformation to pass sorted data to the Aggregator transformation.

- **Limit connected input/output or output ports.** Limit the number of connected input/output or output ports to reduce the amount of data the Aggregator transformation stores in the data cache.
- **Filter before aggregating.** If you use a Filter transformation in the mapping, place the transformation before the Aggregator transformation to reduce unnecessary aggregation.
- If you use sorted input and do not sort data correctly, the session fails.

## Sequence Generator Transformation

The Sequence Generator transformation generates sequence of numeric values. Use the Sequence Generator to create unique primary key values, replace missing primary keys, or cycle through a sequential range of numbers.

The Sequence Generator transformation is a connected transformation. It contains two output ports that you can connect to one or more transformations. The Integration Service generates a block of sequence numbers each time a block of rows enters a connected transformation. You can make a Sequence Generator reusable, and use it in multiple mappings. You might reuse a Sequence Generator when you perform multiple loads to a single target.



### Sequence Generator Ports

The Sequence Generator transformation provides two output ports: NEXTVAL and CURRVAL. You cannot edit or delete these ports. Likewise, you cannot add ports to the transformation.

- **NEXTVAL.** Connect NEXTVAL to multiple transformations to generate unique values for each row in each transformation. Use the NEXTVAL port to generate sequence numbers by connecting it to a transformation or target. You connect the NEXTVAL port to a downstream transformation to generate the sequence based on the Current Value and Increment By properties.

For example, you might connect NEXTVAL to two target tables in a mapping to generate unique primary key values. The Integration Service creates a column of unique primary key values for each target table. The column of unique primary key values is sent to one target table as a block of sequence numbers. The second target receives a block of sequence numbers from the Sequence Generator transformation only after the first target table receives the block of sequence numbers.

- **CURRVAL.** CURRVAL is NEXTVAL plus the Increment By value. You typically only connect the CURRVAL port when the NEXTVAL port is already connected to a downstream transformation. When a row enters the transformation connected to the CURRVAL port, the Integration Service passes the last-created NEXTVAL value plus the Increment By value.

If you connect the CURRVAL port without connecting the NEXTVAL port, the Integration Service passes a constant value for each row. When you connect the CURRVAL port in a Sequence Generator transformation, the Integration Service processes one row in each block. You can optimize performance by connecting only the NEXTVAL port in a mapping.

**Edit Transformations**

Transformation | Ports | **Properties** | Metadata Extensions

Select transformation: 123 SEQ\_SequenceGenerator

Transformation type: Sequence (Reusable)

Transformation Attribute	Value
Start Value	0
Increment By	1
End Value	2147483647
Current Value	1
Cycle	<input type="checkbox"/>
Number of Cached Values	1000
Reset	<input type="checkbox"/>
Tracing Level	Normal

**Start Value**

First value generated for the sequence

OK Cancel Apply Help

## Update Strategy Transformation

When you design a data warehouse, you need to decide what type of information to store in targets. As part of the target table design, you need to determine whether to maintain all the historic data or just the most recent changes.

Use the following steps to define an update strategy:

1. To control how rows are flagged for insert, update, delete, or reject within a mapping, add an Update Strategy transformation to the mapping.
2. Define how to flag rows when you configure a session. You can flag all rows for insert, delete, or update, or you can select the data driven option, where the Integration Service follows instructions coded into Update Strategy transformations within the session mapping.
3. Define insert, update, and delete options for each target when you configure a session.

### Flagging Rows within a Mapping

For the greatest degree of control over the update strategy, add Update Strategy transformations to a mapping. The most important feature of this transformation is its update strategy expression, used to flag individual rows for insert, delete, update, or reject.

Frequently, the update strategy expressions use the IIF or DECODE function from the transformation language to test each row to see if it meets a particular condition. If it does, you can then assign each row a numeric code to flag it for a particular database operation. For example, the following IIF statement flags a row for reject if the entry date is after the apply date. Otherwise, it flags the row for update:

*IIF ((ENTRY\_DATE > APPLY\_DATE), DD\_REJECT, DD\_UPDATE)*

Operation	Constant	Numeric Value
Insert	DD_INSERT	0
Update	DD_UPDATE	1
Delete	DD_DELETE	2
Reject	DD_REJECT	3

## Specifying an Operation for All Rows

When you configure a session, you can select a single database operation for all rows using the Treat Source Rows As setting.

- **Insert.** Treat all rows as inserts. If inserting the row violates a primary or foreign key constraint in the database, the Integration Service rejects the row.
- **Delete.** Treat all rows as deletes. For each row, if the Integration Service finds a corresponding row in the target table (based on the primary key value), the Integration Service deletes it. Note that the primary key constraint must exist in the target definition in the repository.
- **Update.** Treat all rows as updates. For each row, the Integration Service looks for a matching primary key value in the target table. If it exists, the Integration Service updates the row. The primary key constraint must exist in the target definition.
- **Data Driven.** Integration Service follows instructions coded into Update Strategy and Custom transformations within the session mapping to determine how to flag rows for insert, delete, update, or reject.

If the mapping for the session contains an Update Strategy transformation, this field is marked Data Driven by default.

If you do not choose Data Driven when a mapping contains an Update Strategy or Custom transformation, the Workflow Manager displays a warning. When you run the session, the Integration Service does not follow instructions in the Update Strategy or Custom transformation in the mapping to determine how to flag rows.

## Specifying Operations for Individual Target Tables

Once you determine how to treat all rows in the session, you also need to set update strategy options for individual targets. Define the update strategy options in the Transformations view on Mapping tab of the session properties.

You can set the following update strategy options:

- **Insert.** Select this option to insert a row into a target table.
- **Delete.** Select this option to delete a row from a table.
- **Update.** You have the following options in this situation:
  - **Update as Update.** Update each row flagged for update if it exists in the target table.
  - **Update as Insert.** Insert each row flagged for update.
  - **Update else Insert.** Update the row if it exists. Otherwise, insert it.
- **Truncate table.** Select this option to truncate the target table before loading data.

## Forwarding Rejected Rows

You can configure the Update Strategy transformation to either pass rejected rows to the next transformation or drop them. By default, the Integration Service forwards rejected rows to the next transformation. The Integration Service flags the rows for reject and writes them to the session reject file. If you do not select Forward Rejected Rows, the Integration Service drops rejected rows and writes them to the session log file.

If you enable row error handling, the Integration Service writes the rejected rows and the dropped rows to the row error logs. It does not generate a reject file. If you want to write the dropped rows to the session log in addition to the row error logs, you can enable verbose data tracing.

## Aggregator and Update Strategy Transformations

When you connect Aggregator and Update Strategy transformations as part of the same pipeline, you have the following options:

- Position the Aggregator before the Update Strategy transformation. In this case, you perform the aggregate calculation, and then use the Update Strategy transformation to flag rows that contain the results of this calculation for insert, delete, or update.
- Position the Aggregator after the Update Strategy transformation. Here, you flag rows for insert, delete, update, or reject before you perform the aggregate calculation. How you flag

a particular row determines how the Aggregator transformation treats any values in that row used in the calculation. For example, if you flag a row for delete and then later use the row to calculate the sum, the Integration Service subtracts the value appearing in this row. If the row had been flagged for insert, the Integration Service would add its value to the sum.

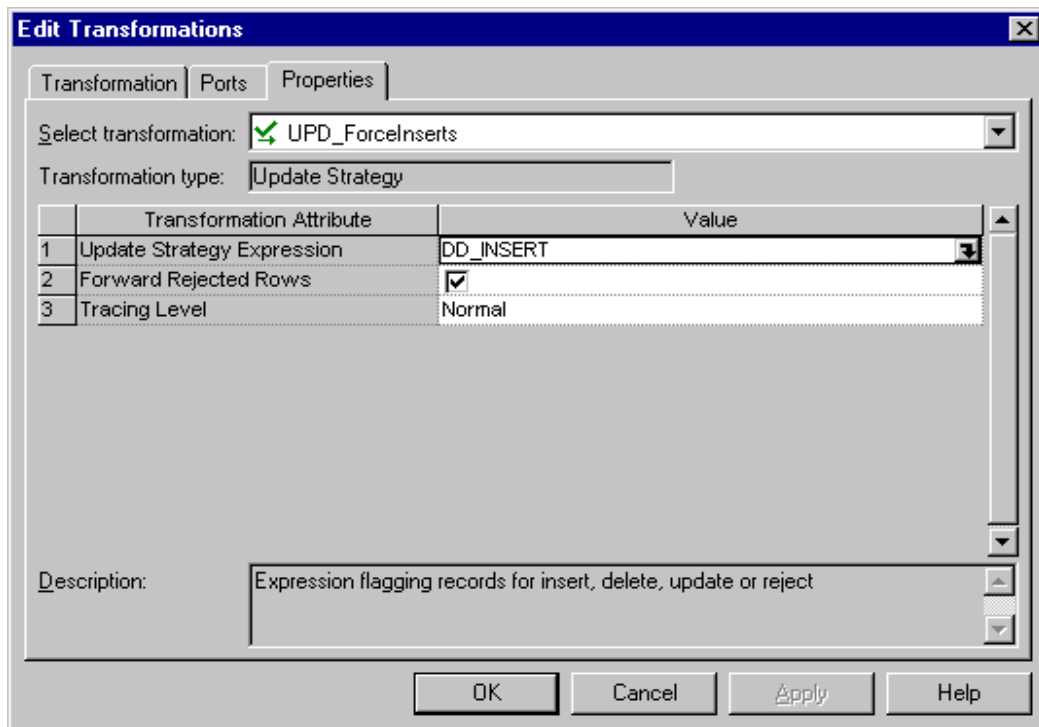
### Lookup and Update Strategy Transformations

When you create a mapping with a Lookup transformation that uses a dynamic lookup cache, you must use Update Strategy transformations to flag the rows for the target tables. When you configure a session using Update Strategy transformations and a dynamic lookup cache, you must define certain session properties. You must define the Treat Source Rows As option as Data Driven on the Properties tab in the session properties.

You must also define the following update strategy target table options:

- Select Insert
- Select Update as Update
- Do not select Delete

These update strategy target table options ensure that the Integration Service updates rows marked for update and inserts rows marked for insert.



### Update Strategy Checklist

Choosing an update strategy requires setting the right options within a session and possibly adding Update Strategy transformations to a mapping. This section summarizes what you need to implement different versions of an update strategy.

- **Only perform inserts into a target table.** When you configure the session, select Insert for the Treat Source Rows As session property. Also, make sure that you select the Insert option for all target instances in the session.
- **Delete all rows in a target table.** When you configure the session, select Delete for the Treat Source Rows As session property. Also, make sure that you select the Delete option for all target instances in the session.
- **Only perform updates on the contents of a target table.** When you configure the session, select Update for the Treat Source Rows As session property. When you



configure the update options for each target table instance, make sure you select the Update option for each target instance.

- **Perform different database operations with different rows destined for the same target table.** Add an Update Strategy transformation to the mapping. When you write the transformation update strategy expression, use either the DECODE or IIF function to flag rows for different operations (insert, delete, update, or reject). When you configure a session that uses this mapping, select Data Driven for the Treat Source Rows As session property. Make sure that you select the Insert, Delete, or one of the Update options for each target table instance.
- **Reject data.** Add an Update Strategy transformation to the mapping. When you write the transformation update strategy expression, use DECODE or IIF to specify the criteria for rejecting the row. When you configure a session that uses this mapping, select Data Driven for the Treat Source Rows As session property.

## Stored Procedure

A Stored Procedure transformation is an important tool for populating and maintaining databases. Database administrators create stored procedures to automate tasks that are too complicated for standard SQL statements.

A stored procedure is a precompiled collection of Transact-SQL, PL-SQL or other database procedural statements and optional flow control statements, similar to an executable script. Stored procedures are stored and run within the database. You can run a stored procedure with the EXECUTE SQL statement in a database client tool, just as you can run SQL statements. Unlike standard SQL, however, stored procedures allow user-defined variables, conditional statements, and other powerful programming features.

Not all databases support stored procedures, and stored procedure syntax varies depending on the database. You might use stored procedures to complete the following tasks:

- Check the status of a target database before loading data into it.
- Determine if enough space exists in a database.
- Perform a specialized calculation.
- Drop and recreate indexes.

Database developers and programmers use stored procedures for various tasks within databases, since stored procedures allow greater flexibility than SQL statements. Stored procedures also provide error handling and logging necessary for critical tasks. Developers create stored procedures in the database using the client tools provided with the database.

The stored procedure must exist in the database before creating a Stored Procedure transformation, and the stored procedure can exist in a source, target, or any database with a valid connection to the Integration Service.

You might use a stored procedure to perform a query or calculation that you would otherwise make part of a mapping. For example, if you already have a well-tested stored procedure for calculating sales tax, you can perform that calculation through the stored procedure instead of recreating the same calculation in an Expression transformation.

## Input and Output Data

One of the most useful features of stored procedures is the ability to send data to the stored procedure, and receive data from the stored procedure. There are three types of data that pass between the Integration Service and the stored procedure:

- Input/output parameters
- Return values
- Status codes

Some limitations exist on passing data, depending on the database implementation, which are discussed throughout this chapter. Additionally, not all stored procedures send and receive data. For example, if you write a stored procedure to rebuild a database index at the end of a session, you cannot receive data, since the session has already finished.

## Input/Output Parameters

For many stored procedures, you provide a value and receive a value in return. These values are known as input and output parameters. For example, a sales tax calculation stored procedure can take a single input parameter, such as the price of an item. After performing the calculation, the stored procedure returns two output parameters, the amount of tax, and the total cost of the item including the tax. The Stored Procedure transformation sends and receives input and output parameters using ports, variables, or by entering a value in an expression, such as 10 or SALES.

## Return Values

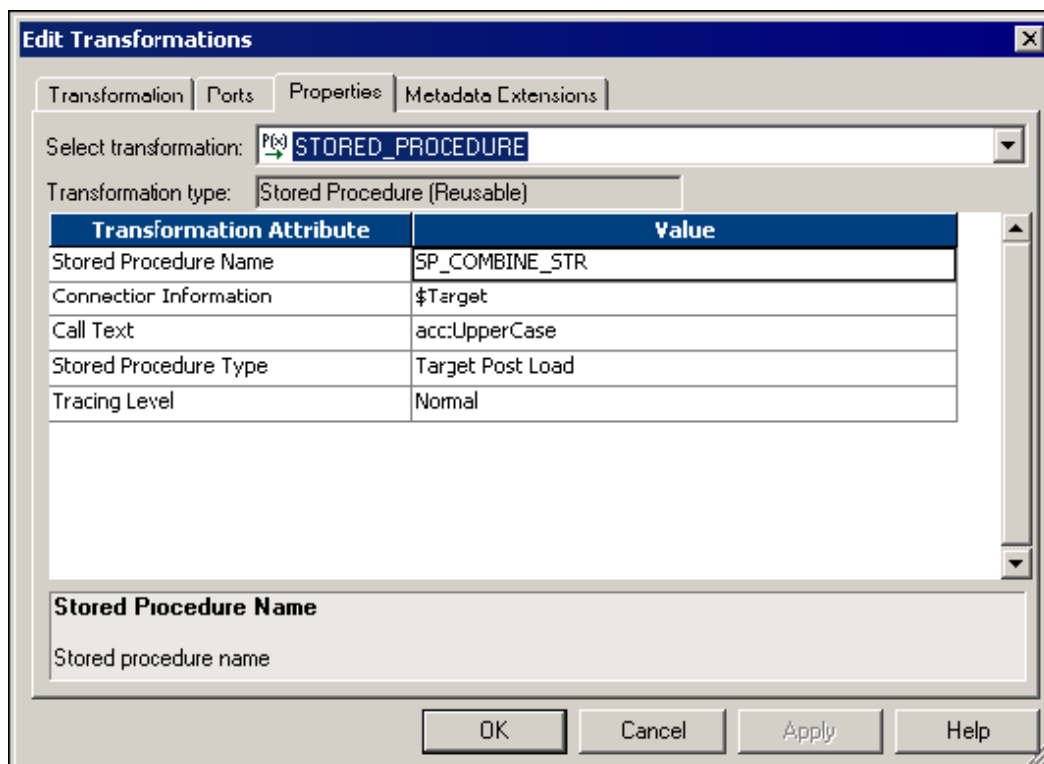
Most databases provide a return value after running a stored procedure. Depending on the database implementation, this value can either be user-definable, which means that it can act similar to a single output parameter, or it may only return an integer value.

The Stored Procedure transformation captures return values in a similar manner as input/output parameters, depending on the method that the input/output parameters are captured. In some instances, only a parameter or a return value can be captured. If a stored procedure returns a result set rather than a single return value, the Stored Procedure transformation takes only the first value returned from the procedure.

**Note:** An Oracle stored function is similar to an Oracle stored procedure, except that the stored function supports output parameters or return values. In this chapter, any statements regarding stored procedures also apply to stored functions, unless otherwise noted.

## Status Codes

Status codes provide error handling for the Integration Service during a workflow. The stored procedure issues a status code that notifies whether or not the stored procedure completed successfully. You cannot see this value. The Integration Service uses it to determine whether to continue running the session or stop. You configure options in the Workflow Manager to continue or stop the session in the event of a stored procedure error.





## Connected and Unconnected

Stored procedures run in either connected or unconnected mode. The mode you use depends on what the stored procedure does and how you plan to use it in a session. You can configure connected and unconnected Stored Procedure transformations in a mapping.

- **Connected.** The flow of data through a mapping in connected mode also passes through the Stored Procedure transformation. All data entering the transformation through the input ports affects the stored procedure. You should use a connected Stored Procedure transformation when you need data from an input port sent as an input parameter to the stored procedure, or the results of a stored procedure sent as an output parameter to another transformation.
- **Unconnected.** The unconnected Stored Procedure transformation is not connected directly to the flow of the mapping. It either runs before or after the session, or is called by an expression in another transformation in the mapping.

## Custom Transformation

Custom transformations operate in conjunction with procedures you create outside of the Designer interface to extend PowerCenter functionality. You can create a Custom transformation and bind it to a procedure that you develop using the functions. Each Custom transformation specifies a module and a procedure name. You can create a Custom transformation based on an existing shared library or a DLL containing the procedure or you can create a Custom transformation as the basis for creating the procedure. When you create a Custom transformation to use with an existing shared library or DLL, make sure you define the correct module and procedure name. When you create a Custom transformation as the basis for creating the procedure, select the transformation and generate the code. The Designer uses the transformation properties when it generates the procedure code. It generates code in a single directory for all transformations sharing a common module name.

The Designer generates the following files:

- **m\_<module\_name>.c.** Defines the module.
- **p\_<procedure\_name>.c.** Defines the procedure in the module.
- **makefile.aix, makefile.aix64, makefile.hp, makefile.hp64, makefile.hpparisc64, makefile.linux, makefile.sol and makefile.sol64.** Make files for the UNIX platforms.

Transformation Attribute	Value
Language	
Module Identifier	XML_Mod
Function Identifier	parse_proc
Runtime Location	\$PMEExtProcDir
Tracing Level	Normal
Is Partitionable	<input type="checkbox"/>
Inputs Must Block	<input checked="" type="checkbox"/>
Is Active	<input checked="" type="checkbox"/>
Update Strategy Transformation	<input type="checkbox"/>
Transformation Scope	All Input
Generate Transaction	<input type="checkbox"/>
Output Is Repeatable	Never
Requires Single Thread Per Partition	<input checked="" type="checkbox"/>

**Language**  
Programming language used to develop this Custom Transformation.

Use the following rules and guidelines while using Custom transformations:

- Custom transformations are connected transformations. You cannot reference a Custom transformation in an expression.
- You can include multiple procedures in one module. For example, you can include an XML writer procedure and an XML parser procedure in the same module.
- You can bind one shared library or DLL to multiple Custom transformation instances if you write the procedure code to handle multiple Custom transformation instances.
- When you write the procedure code, you must make sure it does not violate basic mapping rules.
- The Custom transformation sends and receives high precision decimals as high precision decimals.
- Use multi-threaded code in Custom transformation procedures.

## XML Transformation

---

### XML Source Qualifier Transformation

An XML Source Qualifier transformation can be added to a mapping by dragging an XML source definition to the Mapping Designer workspace or by manually creating one. When you add an XML source definition to a mapping, you need to connect it to an XML Source Qualifier transformation. The XML Source Qualifier transformation defines the data elements that the Integration Service reads when it executes a session. It determines how the PowerCenter reads the source data.

### XML Parser Transformation

An XML Parser transformation is used to extract XML inside a pipeline. The XML Parser transformation lets you extract XML data from messaging systems, such as TIBCO or MQ Series, and from other sources, such as files or databases. The XML Parser transformation functionality is similar to the XML source functionality, except it parses the XML in the pipeline. For example, you might want to extract XML data from a TIBCO source and pass the data to relational targets. The XML Parser transformation reads XML data from a single input port and writes data to one or more output ports.

### XML Generator Transformation

An XML Generator transformation is used to create XML inside a pipeline. The XML Generator transformation lets you read data from messaging systems, such as TIBCO and MQ Series, or from other sources, such as files or databases. The XML Generator transformation functionality is similar to the XML target functionality, except it generates the XML in the pipeline. For example, you might want to extract data from relational sources and pass XML data to targets. The XML Generator transformation accepts data from multiple ports and writes XML through a single output port.

## Mapplet Designer

---

You can design a mapplet to contain sets of transformation logic to be reused in multiple mappings. It allows reusing transformation logic and can contain as many transformations as needed.

Mapplets can:

- Include source definitions
- Accept data from sources in a mapping
- Include multiple transformations
- Pass data to multiple pipelines
- Contain unused ports

## Mapping Designer

A mapping specifies how to move and transform data from sources to targets. Mappings include source and target definitions and transformations. When the Informatica Server runs a session, it uses the instructions configured in the mapping to read, transform, and write data.

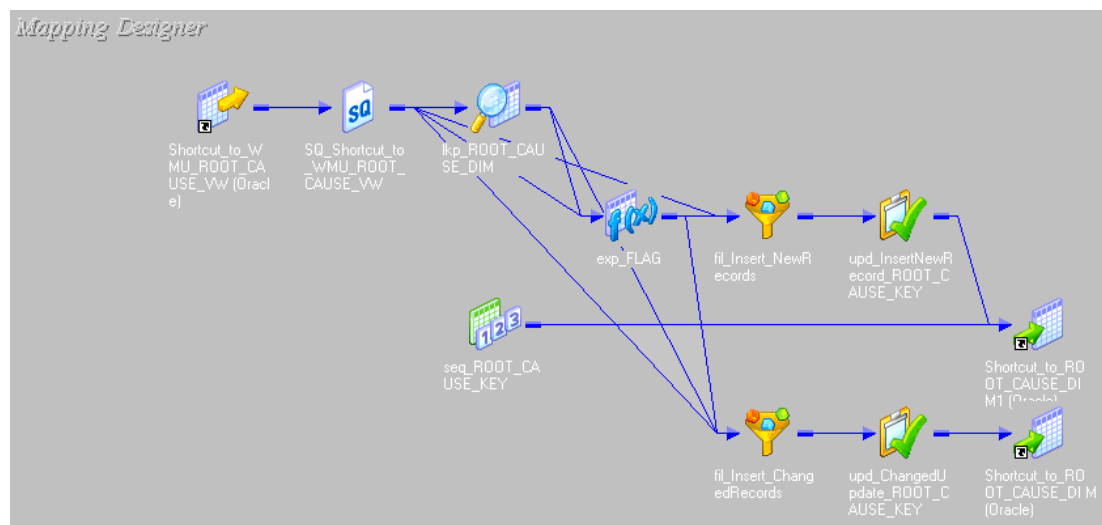
Every mapping must contain the following components:

- Source definition
- Transformation
- Target Definition
- Connectors

A mapping can also contain one or more mapplets. A mapplet is a set of transformations that you build in the Mapplet Designer and can use in multiple mappings.

When you add an object to a mapping, you configure the properties according to the way you want the Integration Service to transform the data. You also connect the mapping objects according to the way you want the Integration Service to move the data. You connect the objects through ports.

Below figure shows a normal mapping in an iconic view:



Use the following procedure as a guideline when you develop a mapping:

1. **Verify that all source, target, and reusable objects are created.** Create source and target definitions. If you want to use mapplets, you must create them also. You can create reusable transformations in the Transformation Developer, or you can create them while you develop a mapping.
2. **Create the mapping.** Create a mapping by dragging a source, target, mapplet, or reusable transformation into the Mapping Designer workspace, or you can click Mapping > Create from the menu.
3. **Add sources and targets.** Add sources and targets to the mapping.
4. **Add transformations and transformation logic.** Add transformations to the mapping and build transformation logic into the transformation properties.
5. **Connect the mapping.** Connect the mapping objects to create a flow of data from sources to targets, through mapplets and transformations that add, remove, or modify data along this flow.
6. **Validate the mapping.** Validate the mapping to identify connection or transformation errors.

7. **Save the mapping.** When you save the mapping, the Designer validates it, identifying any errors. The Designer displays validation messages in the Output window. A mapping with errors is invalid, and you cannot run a session against it until you validate it.

## Mapping Parameters and Variables

In the Designer, use mapping parameters and variables to make mappings more flexible. Mapping parameters and variables represent values in mappings and maplets. If you declare mapping parameters and variables in a mapping, you can reuse a mapping by altering the parameter and variable values of the mapping in the session. This can reduce the overhead of creating multiple mappings when only certain attributes of a mapping need to be changed. When you use a mapping parameter or variable in a mapping, first you declare the mapping parameter or variable for use in each maplet or mapping. Then, you define a value for the mapping parameter or variable before you run the session.

### Mapping Parameters

A mapping parameter represents a constant value that you can define before running a session. A mapping parameter retains the same value throughout the entire session. When you use a mapping parameter, you declare and use the parameter in a mapping or maplet. Then define the value of the parameter in a parameter file. During the session, the Integration Service evaluates all references to the parameter to that value.

When you want to use the same value for a mapping parameter each time you run the session, use the same parameter file for each session run. When you want to change the value of a mapping parameter between sessions you can either:

- Update the parameter file between sessions.
- Create a different parameter file and configure the session to use the new file.
- Remove the parameter file from the session properties so that the session uses the configured initial value of the parameter in the mapping.

### Create a Mapping Parameter

You can create mapping parameters for any mapping or maplet. You can declare as many mapping parameters as you need. Once declared, use the parameter in the mapping or maplet.

*To create a mapping parameter:*

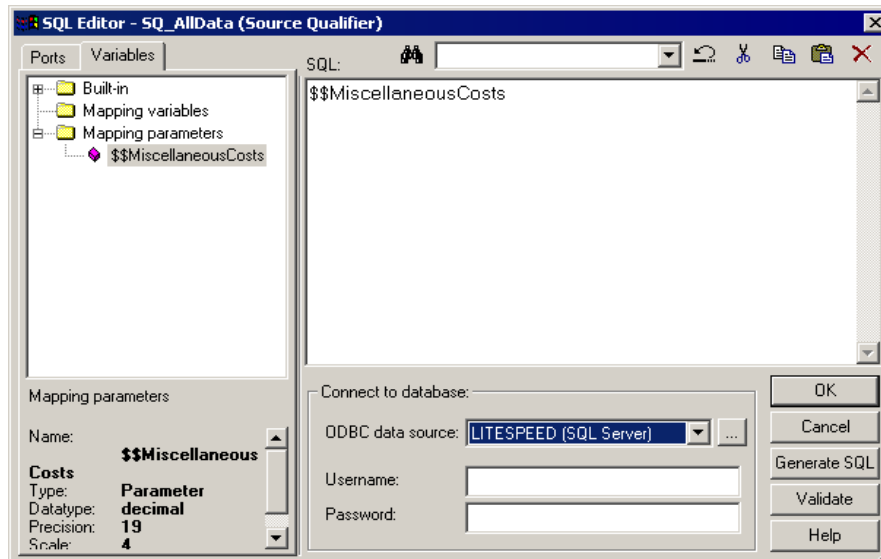
1. In the Mapping Designer, click Mappings > Parameters and Variables. Or In the Maplet Designer, click Maplet > Parameters and Variables.
2. Click the Add button.
3. Enter the required information and click OK.

Name	Type	Datatype	Prec	Scale	Aggrega...
\$MiscellaneousCosts	Parameter	decimal	19	4	

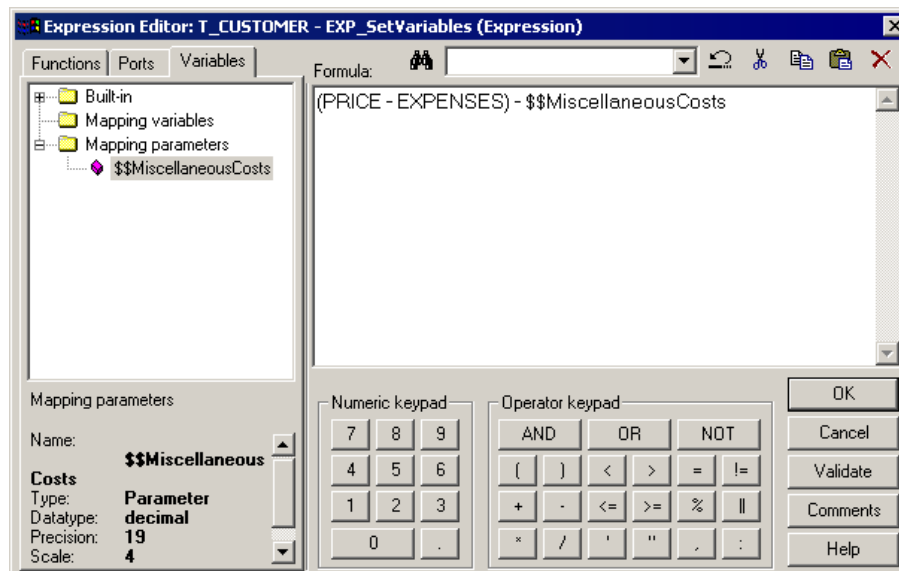
## Use a Mapping Parameter

After you create a parameter, use it in the Expression Editor of any transformation in a mapping or maplet. You can also use it in Source Qualifier transformations and reusable transformations. In a Source Qualifier transformation, mapping parameters appear on the Variables tab in the SQL Editor. When using mapping parameters in a Source Qualifier transformation, follow these guidelines:

- Enclose string parameters in string identifiers appropriate to the source system.
- When necessary, change the format of the datetime parameter to match the format in the source.



You can also use mapping parameters in the Expression Editor. When you use mapping parameters in the Expression Editor, do not enclose string parameters in string identifiers. The Integration Service handles parameters just like any other port identifiers.



## Define a Parameter Value

Before you run a session, define values for mapping parameters in the parameter file. When you do not define a parameter value, the Integration Service uses the initial value for the parameter. If the initial value is not defined, the Integration Service uses the default value for the parameter datatype.

## Mapping Variables

In the Designer, you can create mapping variables in a mapping or maplet. After you create a mapping variable, it appears in the Expression Editor. You can then use it in any expression in the mapping or maplet. You can also use mapping variables in a source qualifier filter, user-defined join, or extract override, and in the Expression Editor of reusable transformations.

Unlike mapping parameters, mapping variables are values that can change between sessions. The Integration Service saves the latest value of a mapping variable to the repository at the end of each successful session. During the next session run, it evaluates all references to the mapping variable to the saved value. You can override a saved value with the parameter file. You can also clear all saved values for the session in the Workflow Manager.

The Integration Service holds two different values for a mapping variable during a session run:

- Start value of a mapping variable
- Current value of a mapping variable

You can create a mapping variable for any mapping or maplet. You can create as many variables as you need. Once created, use the variable in the mapping or maplet.

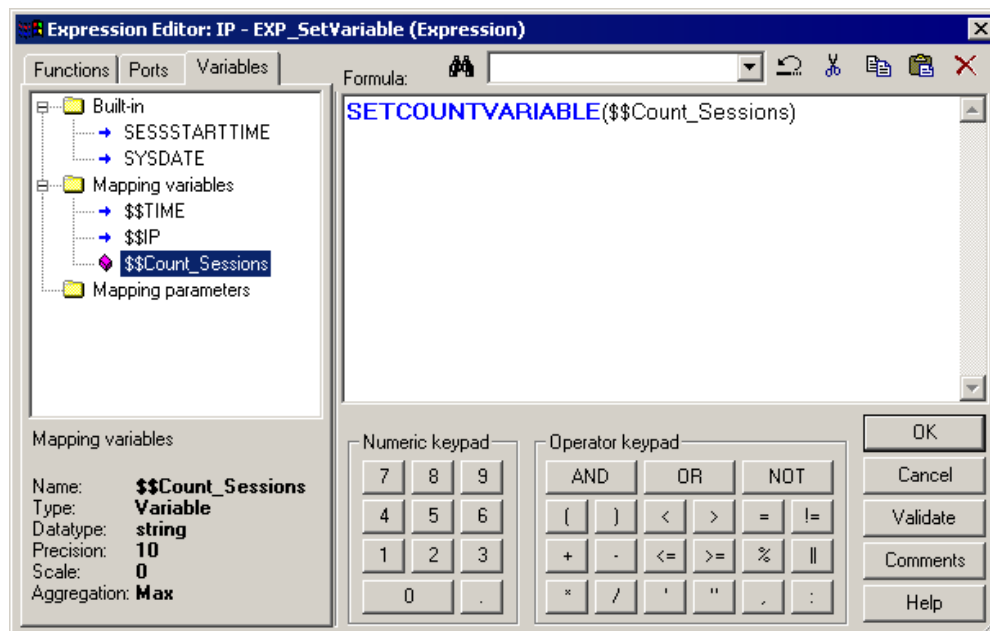
### Set a Mapping Variable Value

After you declare a variable, use it in any expression in the mapping or maplet. You can also use a mapping variable in a Source Qualifier transformation or reusable transformation. In other transformations in a maplet or mapping, mapping variables appear in the Expression Editor. When you write expressions that use mapping variables, you do not need string identifiers for string variables. When you validate the expression, the Designer treats the variable as an Integer datatype.

You can also use mapping variables in transformation overrides in the session properties. You can override properties such as a filter or user-defined join in a Source Qualifier transformation.

When you use a mapping variable, you have to determine how to set the value of the mapping variable. Use a variable function to set a variable value. Use a variable function in any of the following transformations:

- Expression
- Filter
- Router
- Update Strategy



## Override or Clear Saved Values

After a successful session run, the Integration Service saves the final value of each variable in the repository. When you do not want to use that value for the next session run, you can override it in the parameter file.

When you do not want to use any of the variable values saved for a session, you can clear all saved values. You can clear variable values for a session using the Workflow Manager. After you clear variables values from the repository, the Integration Service runs the session as if for the first time.

## Default Values

When the Integration Service needs an initial value, and you did not declare an initial value for the parameter or variable, the Integration Service uses a default value based on the datatype of the parameter or variable.

Data	Default Value
String	Empty string.
Numeric	0
Datetime	1/1/1753 A.D. or 1/1/1 when the Integration Service is configured for compatibility with 4.0.

## Parameter Files

A parameter file is a list of parameters and associated values for a workflow, worklet, or session. Parameter files provide flexibility to change these variables each time you run a workflow or session. You can create multiple parameter files and change the file you use for a session or workflow. You can create a parameter file using a text editor such as WordPad or Notepad.

A parameter file contains the following types of parameters and variables:

- **Service variable.** Defines a service variable for an Integration Service.
- **Service process variable.** Defines a service process variable for an Integration Service that runs on a specific node.
- **Workflow variable.** References values and records information in a workflow. For example, use a workflow variable in a Decision task to determine whether the previous task ran properly.
- **Worklet variable.** References values and records information in a worklet. Use predefined worklet variables in a parent workflow, but you cannot use workflow variables from the parent workflow in a worklet.
- **Session parameter.** Defines a value that can change from session to session, such as a database connection or file name.
- **Mapping parameter.** Defines a value that remains constant throughout a session, such as a state sales tax rate.
- **Mapping variable.** Defines a value that can change during the session. The Integration Service saves the value of a mapping variable to the repository at the end of each successful session run and uses that value the next time you run the session.

Use one of the following methods to choose the parameter file the Integration Service uses with a workflow or session:

- Enter the parameter file name and directory in the workflow or session properties.
- Start the workflow using *pmcmd*, and enter the parameter file name and directory in the command line.

## Using a Parameter File

Parameter files contain several sections preceded by a heading. The heading identifies the Integration Service, Integration Service process, workflow, worklet, or session to which you want to



assign parameters or variables. You assign parameters and variables directly below this heading in the file, entering each parameter or variable on a new line. Enter the parameter or variable in the form *name=value*. The Integration Service interprets all characters between the equals sign and the end of the line as the parameter value. You can list parameters and variables in any order for each section. You can define service variables, service process variables, workflow variables, session parameters, mapping parameters, and mapping variables in any section in the parameter file.

### Sample Parameter File

The following text is an excerpt from a parameter file that contains service variables for one Integration Service and parameters for four workflows:

```
[Service:IntSvs_01]
$PMSuccessEmailUser=pcadmin@mail.com
$PMFailureEmailUser=pcadmin@mail.com
[HET_TGTS.WF:wf_TCOMMIT_INST_ALIAS]
$$platform=unix
[HET_TGTS.WF:wf_TGTS_ASC_ORDR.ST:s_TGTS_ASC_ORDR]
$$platform=unix
$DBConnection_ora=qasrvrk2_hp817
[ORDERS.WF:wf_PARAM_FILE.WT:WL_PARAM_Lvl_1]
$$DT_WL_lv1=02/01/2005 01:05:11
$$Double_WL_lv1=2.2
[ORDERS.WF:wf_PARAM_FILE.WT:WL_PARAM_Lvl_1.WT:NWL_PARAM_Lvl_2]
$$DT_WL_lv2=03/01/2005 01:01:01
$$Int_WL_lv2=3
$$String_WL_lv1_2=cccccc
```

### Guidelines for Creating Parameter Files

Use the following rules and guidelines when you create parameter files:

- **Capitalize folder and session names the same as they appear in the Workflow Manager.** Folder and session names are case sensitive in the parameter file. Service and node names are not case sensitive.
- **Define service and service process variables properly.** Service and service process variables must begin with \$PM. If they do not, the Integration Service does not recognize them as service or service process variables.
- **You can define a service and service process variables for workflows, worklets, and sessions.** If you define a service or service process variable in a workflow, worklet, or session section of the parameter file, the variable applies to any service process that runs the task.
- **List all necessary mapping parameters and variables.** Mapping parameter and variable values become start values for parameters and variables in a mapping. Mapping parameter and variable names are not case sensitive.
- **Enter folder names for non-unique session names.** When a session name exists more than once in a repository, enter the folder name to indicate the location of the session.
- **Use multiple parameter files.** You assign parameter files to workflows, worklets, and sessions individually. You can specify the same parameter file for all of these tasks or create multiple parameter files.
- **Create a parameter file section for each session.** To include parameter and variable information for more than one session in the parameter file, create a section for each session. The folder name is optional. The following parameter file example has multiple sections:

```
[folder_name.session_name]
parameter_name=value
variable_name=value
```



```
mapplet_name.parameter_name=value
[folder2_name.session_name]
parameter_name=value
variable_name=value
mapplet_name.parameter_name=value
```

- **Specify headings in any order.** You can place headings in any order in the parameter file. However, if you define the same heading more than once in the file, the Integration Service uses the parameter or variable values below the first instance of the heading.
- **Specify parameters and variables in any order.** You can specify the parameters and variables in any order below a heading.
- **When defining parameter values, do not use unnecessary line breaks or spaces.** The Integration Service might interpret additional spaces as part of a value.
- **List all session parameters.** Session parameters do not have default values. An undefined session parameter fails a session. Session parameter names are not case sensitive.
- **Override initial values of workflow variables if necessary.** If a workflow contains an Assignment task that changes the value of a workflow variable, the next session in the workflow uses the latest value of the variable as the initial value for the session. To override the initial value, define a new value for the variable in the session section of the parameter file.
- **Use correct date formats for datetime values.** Use the following date formats for datetime values:
  - » MM/DD/RR
  - » MM/DD/RR HH24:MI:SS
  - » MM/DD/YYYY
  - » MM/DD/YYYY HH24:MI:SS
- **Do not enclose parameters or variables in quotes.** The Integration Service interprets everything after the equal sign as part of the value.
- **Precede parameters and variables in mapplets with the mapplet name as follows:**

```
mapplet_name.parameter_name=value
mapplet2_name.variable_name=value
```

### Configuring the Parameter File Location

You can specify the parameter file name and directory in the workflow or session properties.

To enter a parameter file in the workflow properties:

1. Open a Workflow in the Workflow Manager.
2. Click Workflows > Edit. The Edit Workflow dialog box appears.
3. Click the Properties tab.
4. Enter the parameter directory and name in the Parameter Filename field.
5. Click OK.

You can enter either a direct path or a process variable directory. Use the appropriate delimiter for the Integration Service operating system. If you configured the PowerCenter environment for high availability, include the server in the path.

To enter a parameter file in the session properties:

1. Open a session in the Workflow Manager. The Edit Tasks dialog box appears.
2. Click the Properties tab and open the General Options settings.
3. Enter the parameter directory and name in the Parameter Filename field.
4. You can enter either a direct path or a process variable directory. Use the appropriate delimiter for the Integration Service operating system.
5. Click OK.

### Try It Out

---

1. Create a delimited flat file source, a relational table source and join the sources using a joiner transformation.
2. Create a lookup transformation with dynamic lookup cache and include it in the mapping
3. Create an update strategy transformation in the mapping and flag rows based on the lookup values.
4. Create an aggregator transformation with sorted input enabled and use a sorter transformation to provide sorted data.
5. Define Mapping Parameters and use them to apply source filter in the source qualifier of the relational table source

### Summary

---

- ❖ PowerCenter Designer is used to create source definitions, target definitions, and transformations to build the mappings.
- ❖ Source Analyzer, Warehouse Designer, Transformation Developer, Maplet Designer and Mapping Designer are the tools present in Informatica Designer Tool.
- ❖ Transformations are of two types – Active and Passive.
- ❖ Expression, Sorter and Lookup Transformations are examples of Passive Transformation.
- ❖ Aggregator, Joiner, Filter, Router and Normalizer Transformations are examples of Active Transformation.
- ❖ Mapping Parameters and Variables add flexibility to the mappings.
- ❖ Parameter Files are used to provide values externally to Mapping Parameters and Variables during runtime.

### Test your Understanding

---

1. What is the source qualifier used to read COBOL sources?
2. How will you configure the update strategy of a particular Target table?
3. Can a Lookup Transformation with Dynamic Cache be used as unconnected?
4. Differentiate Mapping Parameter, Mapping Variable and Variable Ports.
5. Differentiate CURRVAL and NEXTVAL ports of Sequence Generator Transformation.
6. What is Persistent Lookup Cache? Where it is used?
7. Differentiate Source Qualifier Join and Joiner Transformation.

## Session 6: Workflow Manager

### Learning Objectives

- Manager Overview
- Source and Target Connection
- Tasks
- Sessions
- Workflows

### Manager Overview

Workflow Manager is used to define a set of instructions called a workflow to execute mappings we build in the Designer. A workflow contains a session and any other task you may want to perform when you run a session. Tasks can include a session, email notification, or scheduling information. These tasks are connected with links in the workflow. You can also create a worklet in the Workflow Manager. A worklet is an object that groups a set of tasks. A worklet is similar to a workflow, but without scheduling information.

The Workflow Manager contains the following tools:

- **Task Developer.** To create reusable tasks we want to run in the workflow.
- **Workflow Designer.** To create a workflow by connecting tasks with links. You can also create non-reusable tasks in the Workflow Designer as we develop the workflow.
- **Worklet Designer.** To create a worklet containing a set of tasks.

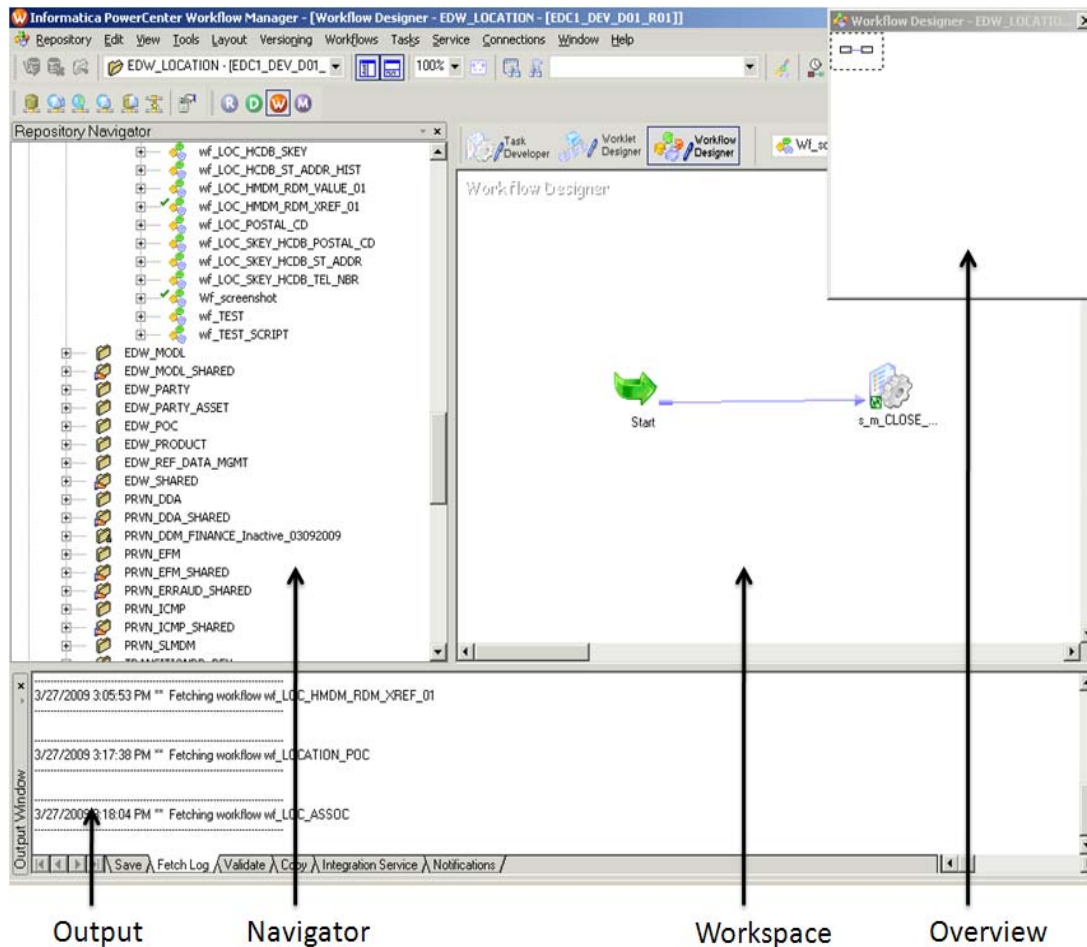
### Workflow Manager Windows

The Workflow Manager displays the following windows to help you create and organize workflows:

- **Navigator.** You can connect to and work in multiple repositories and folders. In the Navigator, the Workflow Manager displays a red icon over invalid objects.
- **Workspace.** You can create, edit, and view tasks, workflows, and worklets.
- **Output.** Contains tabs to display different types of output messages. The Output window contains the following tabs:
  - » **Save.** Displays messages when you save a workflow, worklet, or task. The Save tab displays a validation summary when you save a workflow or a worklet.
  - » **Fetch Log.** Displays messages when the Workflow Manager fetches objects from the repository.
  - » **Validate.** Displays messages when you validate a workflow, worklet, or task.
  - » **Copy.** Displays messages when you copy repository objects.
  - » **Server.** Displays messages from the Integration Service.
  - » **Notifications.** Displays messages from the Repository Service.
- **Overview.** An optional window that lets you easily view large workflows in the workspace. Outlines the visible area in the workspace and highlights selected objects in color. Click View > Overview Window to display this window.

### Source and Target Connection

A connection object is a global object that defines a connection in the repository. You can create and modify connection objects in the Workflow Manager. When you create a connection object, you define values for the connection properties. The properties vary depending on the type of connection you create. You can create, assign permissions, edit, and delete for all connection objects. For relational database connections, you can also copy and replace connection objects.



To create and edit database, FTP, and external loader connections, you must have one of the following privileges:

- Super User
- Manage Connection

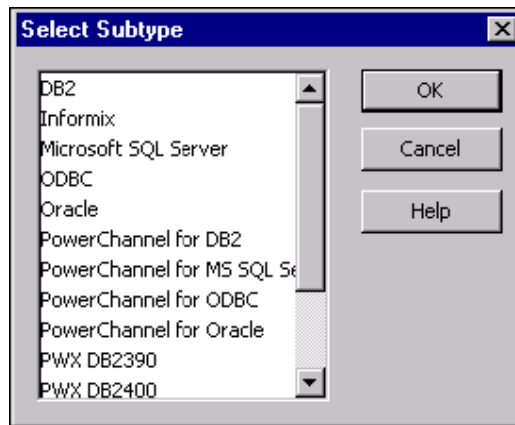
You can configure the following connection information in the Workflow Manager:

- **Relational database connections.** Connections for each source, target, Lookup transformation, and Stored Procedure transformation database. You must create connections to a database before you can create a session that accesses the database.
- **FTP connections.** File Transfer Protocol (FTP) connections object to read files from a FTP host.
- **External loader connections.** An external loader connection to load information directly from a file or pipe rather than running the SQL commands to insert the same data into the database.
- **Queue connections.** Database connections for message queues.
- **Source and target application connections.** Connections to source and target applications. When you create or modify a session that reads from or writes to an application, you can select configured source and target application connections. When you create a connection, the connection properties you need depends on the application.

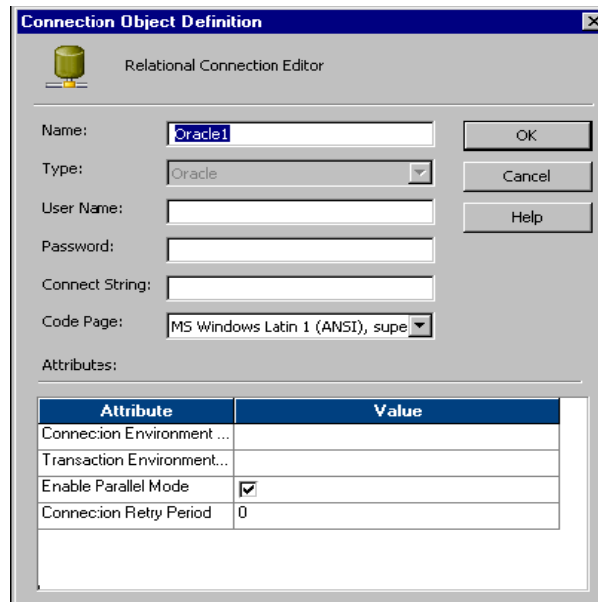
*To create a relational database connection:*

1. In the Workflow Manager, connect to a repository.
2. Click Connections > Relational. The Relational Connection Browser dialog box appears, listing all the source and target database connections.

- Click New. The Select Subtype dialog box appears.



- In Select Subtype dialog box, select the type of database connection you want to create and Click OK.



- Specify Connection Name, Username, password, connection string, code page and other Database related attributes. Click OK.

The new database connection appears in the Connection Browser list.

*To create an external database connection:*

- Click Connections > Loader in the Workflow Manager.
- The Loader Connection Browser dialog box appears. Click New.
- Select an external loader type, and then click OK. The Loader Connection Editor dialog box appears.
- Enter a name for the external loader connection, database user name, password, connect string and the loader properties. Click OK

Attribute	Value
Error Limit	0
Load Mode	Append
Load Method	Use Conventional Path
Enable Parallel Load	Enable Parallel Load
Rows Per Commit	10000
External Loader Executa...	sqlload

## Tasks

The Workflow Manager contains many types of tasks to help you build workflows and worklets. You can create reusable tasks in the Task Developer. Or, create and add tasks in the Workflow or Worklet Designer as you develop the workflow. The Workflow Manager validates tasks attributes and links. If a task is invalid, the workflow becomes invalid. Workflows containing invalid sessions may still be valid.

**Types of Tasks.** The Workflow Manager supports the following types of tasks.

- **Assignment.** Assigns a value to a workflow variable. Non reusable.
- **Command.** Specifies shell commands to run during the workflow. Reusable.
- **Control.** Stops or aborts workflow. Non reusable.
- **Decision.** Specifies a condition to evaluate in the workflow. Non reusable.
- **Email.** Sends email during the workflow. Reusable
- **Event Raise.** Represents the location of the user defined event. Non reusable.
- **Event Wait.** Waits for a pre/user defined event to occur. Non reusable.
- **Session.** Set of instructions to run a mapping. Reusable.
- **Timer.** Waits for a specified period of time to run next task. Non reusable.

### Creating a Task

You can create tasks in the Task Developer, or you can create them in the Workflow Designer or the Worklet Designer as you develop the workflow or worklet. Tasks you create in the Task Developer are reusable. Tasks you create in the Workflow Designer and Worklet Designer are non-reusable by default.

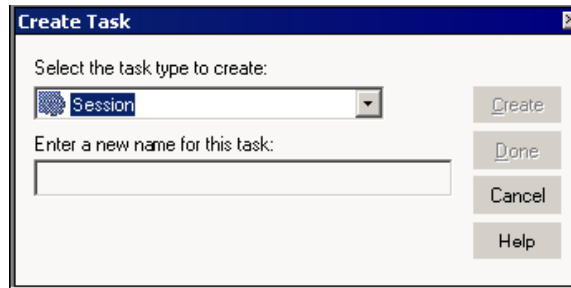
*To create a task in Task Developer:*

1. In the Task Developer, click Tasks > Create. The Create Task dialog box appears.
2. Select the task type you want to create, Command, Session, or Email. Other type or tasks can't be created here.
3. Enter a name for the task.
4. For session tasks, select the mapping you want to associate with the session.
5. Click Create. The Task Developer creates the workflow task.
6. Click Done to close the Create Task dialog box.

*To create tasks in the Workflow Designer or Worklet Designer:*

1. In the Workflow Designer or Worklet Designer, open a workflow or worklet.

- Click Tasks > Create. Select the type of task you want to create.



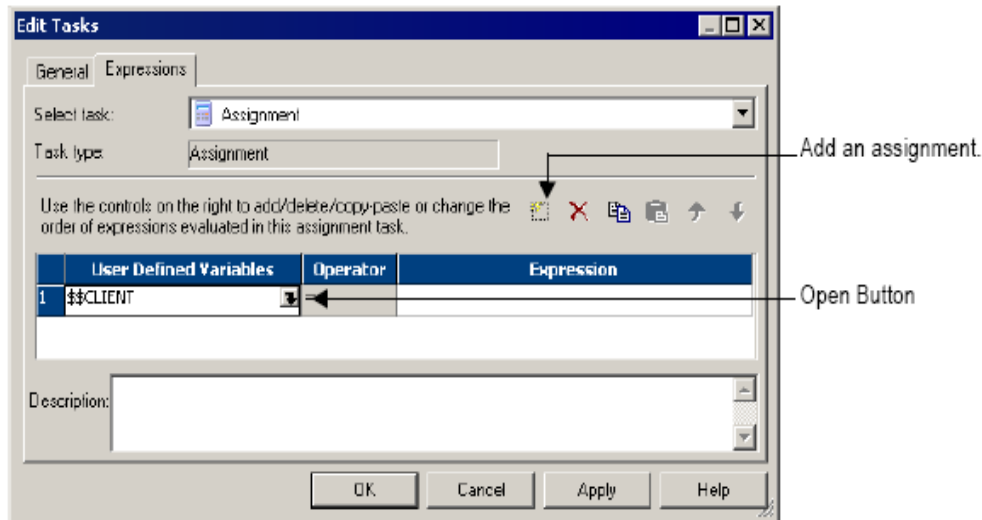
- Enter a name for the task. Click Create. The Workflow Designer or Worklet Designer creates the task and adds it to the workspace.
- Click Done.

### Assignment Task

You can assign a value to a user-defined workflow variable with the Assignment task. To use an Assignment task in the workflow, first create and add the Assignment task to the workflow. Then configure the Assignment task to assign values or expressions to user-defined variables. After you assign a value to a variable using the Assignment task, the Integration Service uses the assigned value for the variable during the remainder of the workflow. You must create a variable before you can assign values to it. You cannot assign values to predefined workflow variables.

*To create an Assignment task:*

- Click Tasks > Create. Select Assignment Task for the task type.



- Enter a name for the Assignment task. Click Create. Then Click Done. The Workflow Designer creates and adds the Assignment task to the workflow.
- Double-click the Assignment task to open the Edit Task dialog box.
- On the Expressions tab, click Add to add an assignment.
- Click the Open button in the User Defined Variables field.
- Select the variable for which you want to assign a value. Click OK.
- Click the Edit button in the Expression field to open the Expression Editor. The Expression Editor shows predefined workflow variables, user-defined workflow variables, variable functions, and Boolean and arithmetic operators.
- Enter the value or expression you want to assign. Click Validate. Then Click OK.



## Command Task

You can specify one or more shell commands to run during the workflow with the Command task. For example, you can specify shell commands in the Command task to delete reject files, copy a file, or archive target files.

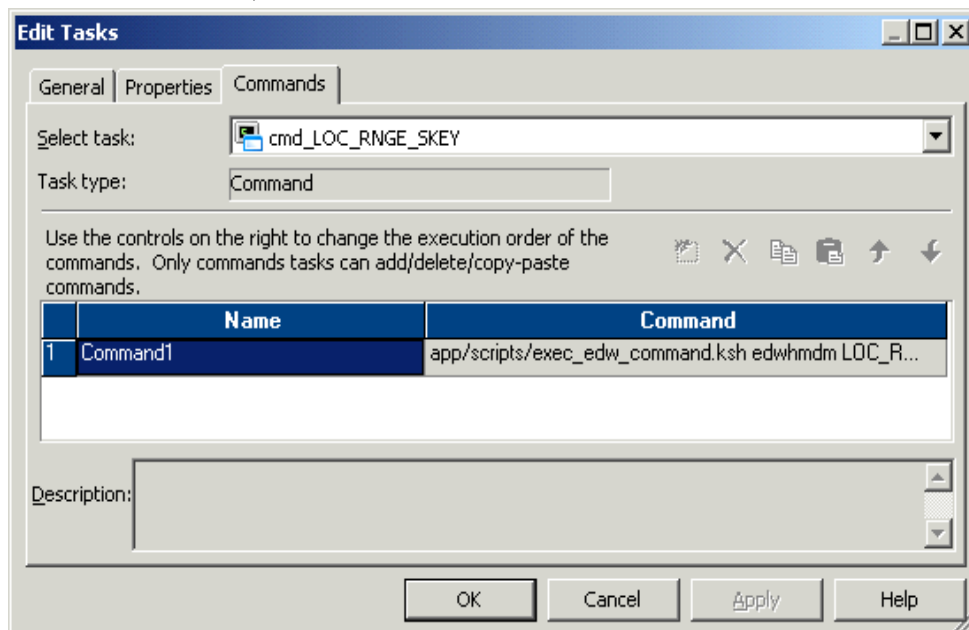
Use a Command task in the following ways:

- **Standalone Command task.** Use a Command task anywhere in the workflow or worklet to run shell commands. Use any valid UNIX command or shell script for UNIX servers, or any valid DOS or batch file for Windows servers.
- **Pre- and post-session shell command.** You can call a Command task as the pre- or post-session shell command for a Session task. You can use process variables or session parameters in pre- and post-session shell commands.

Each shell command runs in the same environment (UNIX or Windows) as the Integration Service. Environment settings in one shell command script do not carry over to other scripts. To run all shell commands in the same environment, call a single shell script that invokes other scripts.

To create a Command task:

1. Click Task > Create. Select Command Task for the task type.
2. Enter a name for the Command task. Click Create. Then click Done.
3. Double-click the Command task in the workspace to open the Edit Tasks dialog box.
4. In the Commands tab, click the Add button to add a command.



5. In the Name field, enter a name for the new command.
6. In the Command field, click the Edit button to open the Command Editor.
7. Enter the command you want to perform. Enter one command in the Command Editor. Click OK.

If you specify non-reusable shell commands for a session, you can promote the non-reusable shell commands to a reusable Command task.

## Control Task

Use the Control task to stop, abort, or fail the top-level workflow or the parent workflow based on an input link condition. A parent workflow or worklet is the workflow or worklet that contains the Control task.

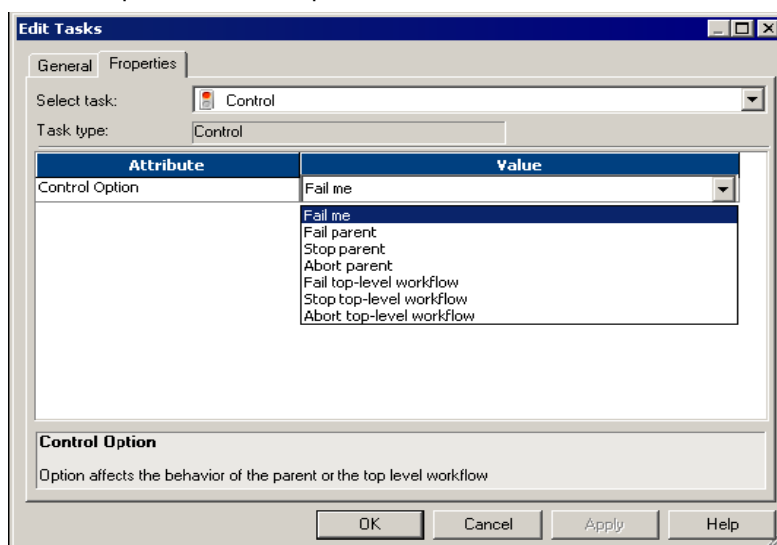


The following are the options in a Control Task:

- **Fail Me.** Marks the Control task as “Failed.” The Integration Service fails the Control task if you choose this option. If you choose Fail Me in the Properties tab and choose Fail Parent If This Task Fails in the General tab, the Integration Service fails the parent workflow.
- **Fail Parent.** Marks the status of the workflow or worklet that contains the Control task as failed after the workflow or worklet completes.
- **Stop Parent.** Stops the workflow or worklet that contains the Control task.
- **Abort Parent.** Aborts the workflow or worklet that contains the Control task.
- **Fail Top-Level Workflow.** Fails the workflow that is running.
- **Stop Top-Level Workflow.** Stops the workflow that is running.
- **Abort Top-Level Workflow.** Aborts the workflow that is running.

To create a Control Task:

1. Click Tasks > Create. Select Control Task for the task type.
2. Enter a name for the Control task. Click Create. Then Click Done. The Workflow Manager creates and adds the Control task to the workflow.
3. Double-click the Control task in the workspace to open it.
4. Configure control options on the Properties tab.



## Decision Task

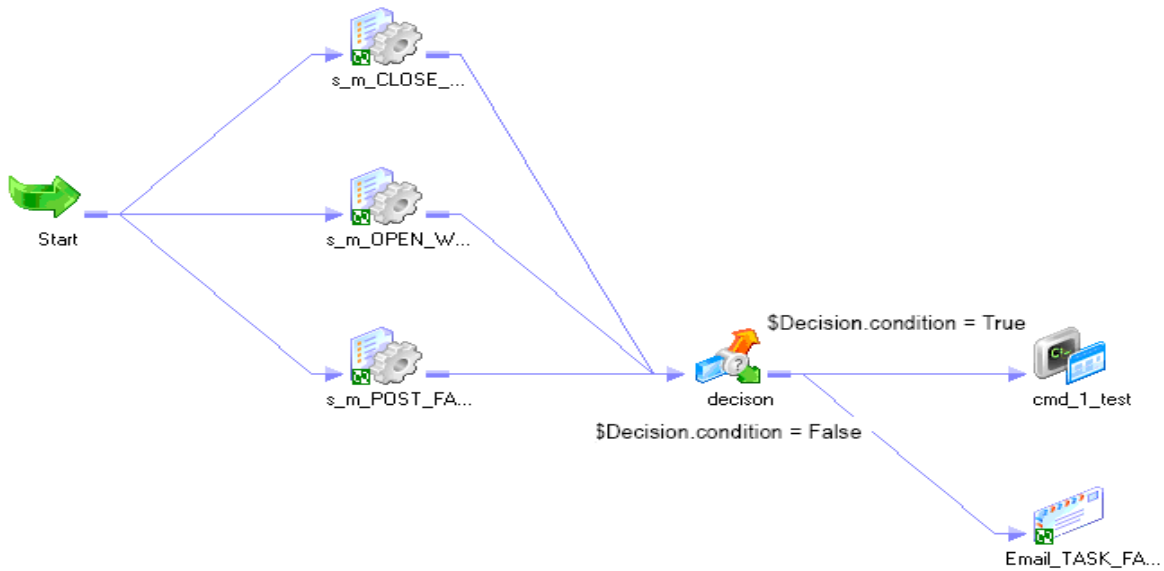
You can enter a condition that determines the execution of the workflow, similar to a link condition with the Decision task. The Decision task has a predefined variable called `$Decision_task_name.condition` that represents the result of the decision condition. The Integration Service evaluates the condition in the Decision task and sets the predefined condition variable to True (1) or False (0). You can specify one decision condition per Decision task. Use the Decision task instead of multiple link conditions in a workflow.

For example, if you want the Integration Service to run the Command task when any of the three sessions fails, use a Decision task with the following decision condition:

```
$Q1_session.status = FAILED OR $Q2_session.status = FAILED OR $Q3_session.status = FAILED
```

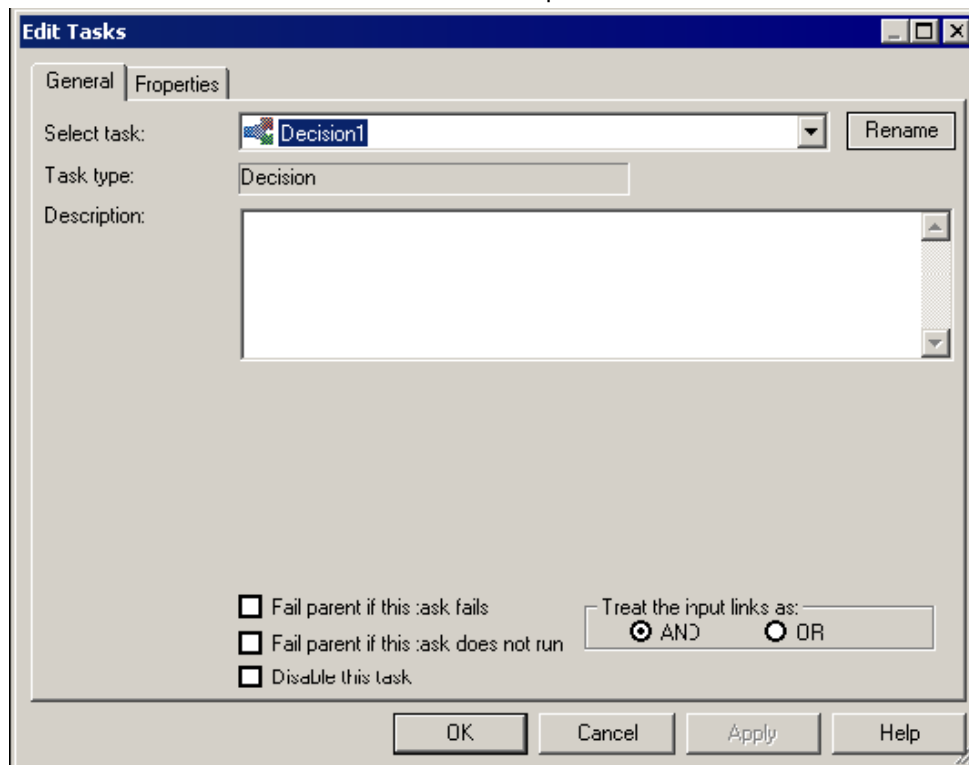
You can then use the predefined condition variable in the input link condition of the Command task. Configure the input link with the following link condition:

```
$Decision.condition = True
```



To create a Decision Task:

1. Click Tasks > Create. Select Decision Task for the task type.
2. Enter a name for the Decision task. Click Create. Then click Done. The Workflow Designer creates and adds the Decision task to the workspace.



3. Double-click the Decision task to open it.
4. Click the Open button in the Value field to open the Expression Editor.
5. In the Expression Editor, enter the condition you want the Integration Service to evaluate. Validate the expression before you close the Expression Editor. Click OK.

## Event Task

You can define events in the workflow to specify the sequence of task execution. The event is triggered based on the completion of the sequence of tasks. Use the following tasks to help you use events in the workflow:

- **Event-Raise task.** Event-Raise task represents a user-defined event. When the Integration Service runs the Event-Raise task, it triggers the event. Use the Event-Raise task with the Event-Wait task to define events.
- **Event-Wait task.** The Event-Wait task waits for an event to occur. Once the event triggers, the Integration Service continues executing the rest of the workflow.

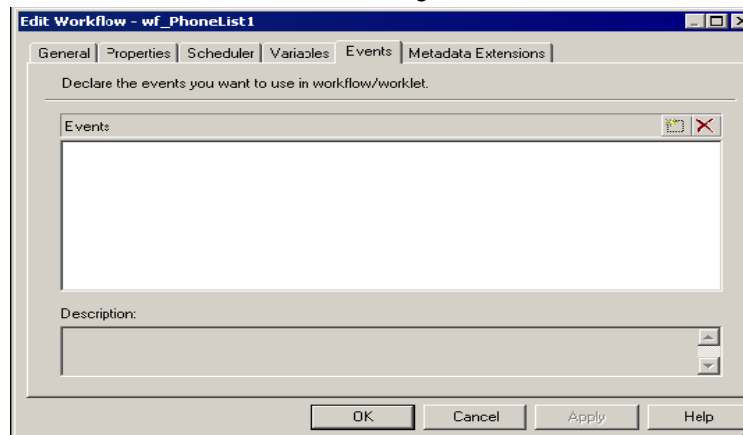
To coordinate the execution of the workflow, you may specify the following types of events for the Event-Wait and Event-Raise tasks:

- **Predefined event.** A predefined event is a file-watch event. For predefined events, use an Event-Wait task to instruct the Integration Service to wait for the specified indicator file to appear before continuing with the rest of the workflow. When the Integration Service locates the indicator file, it starts the next task in the workflow.
- **User-defined event.** A user-defined event is a sequence of tasks in the workflow. Use an Event-Raise task to specify the location of the user-defined event in the workflow. A user-defined event is sequence of tasks in the branch from the Start task leading to the Event-Raise task.

When all the tasks in the branch from the Start task to the Event-Raise task complete, the Event-Raise task triggers the event. The Event-Wait task waits for the Event-Raise task to trigger the event before continuing with the rest of the tasks in its branch.

*To declare a User Defined Event:*

1. In the Workflow Designer, click Workflow > Edit to open the workflow properties.
2. Select the Events tab in the Edit Workflow dialog box.



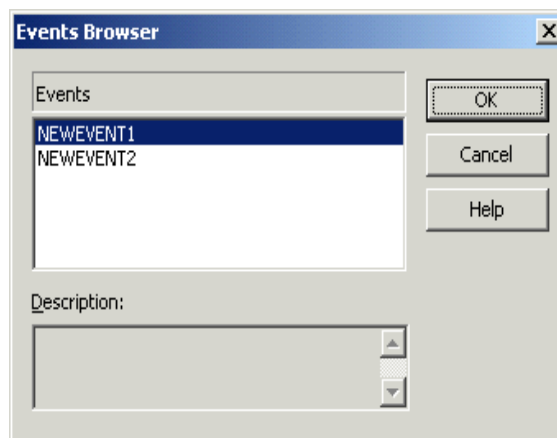
3. Click Add to add an event name. Click OK.

## Using an Event-Raise Task

The Event-Raise task represents the location of a user-defined event. When the Integration Service runs the Event-Raise task, the Event-Raise task triggers the user-defined event.

*To create an Event-Raise Task:*

1. In the Workflow Designer workspace, create an Event-Raise task and place it in the workflow to represent the user-defined event you want to trigger.
2. Double-click the Event-Raise task.
3. Click the Open button in the Value



- field on the Properties tab to open the Events Browser for user-defined events.
4. Choose an event in the Events Browser.
5. Click OK twice to return to the workspace.

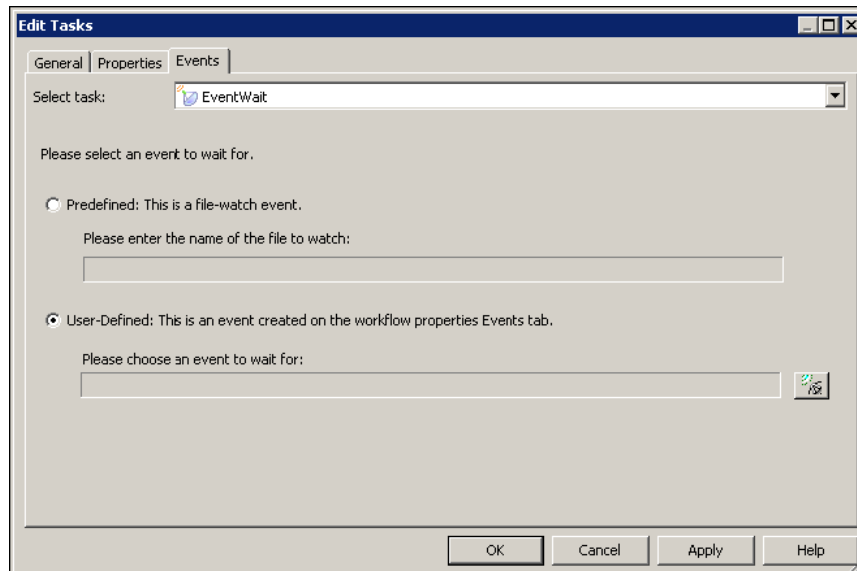
### Using an Event-Wait Task

The Event-Wait task waits for a predefined event or a user-defined event. When you use the Event-Wait task to wait for a predefined event, you specify an indicator file for the Integration Service to watch. The Integration Service waits for the indicator file to appear. Once the indicator file appears, the Integration Service continues running tasks after the Event-Wait task.

You can also use the Event-Wait task to wait for a user-defined event. To use the Event-Wait task for a user-defined event, specify the name of the user-defined event in the Event-Wait task properties. The Integration Service waits for the Event-Raise task to trigger the user-defined event. Once the user-defined event is triggered, the Integration Service continues running tasks after the Event-Wait task.

*To create an Event-Wait Task:*

1. In the workflow, create an Event-Wait task and double-click the Event-Wait task to open it.
2. In the Events tab of the task, select User-Defined or Predefined.



3. If User-Defined is chosen, Click the Event button to open the Events Browser dialog box and Select a user-defined event already declared in the workflow properties.
4. If Pre-defined is chosen, enter the path of the indicator file.
5. If you want the Integration Service to delete the indicator file after it detects the file, select the Delete Filewatch File option in the Properties tab. Click OK

By default, the Event-Wait task waits for the Event-Raise task to trigger the event. By default, the Event-Wait task does not check if the event already occurred. You can select the Enable Past Events option in Properties tab, so that the Integration Service verifies that the event has already occurred and the Integration Service continues executing the next tasks if the event already occurred.

### Timer Task

You can specify the period of time to wait before the Integration Service runs the next task in the workflow with the Timer task. You can choose to start the next task in the workflow at a specified time and date. You can also choose to wait a period of time after the start time of another task, workflow, or worklet before starting the next task.

The Timer task has two types of settings:

- **Absolute time.** You specify the time that the Integration Service starts running the next task in the workflow. You may specify the date and time, or you can choose a user-defined workflow variable to specify the time.
- **Relative time.** You instruct the Integration Service to wait for a specified period of time after the Timer task, the parent workflow, or the top-level workflow starts.

*To Create a Timer Task:*

1. Click Tasks > Create. Select Timer Task for the task type. Double-click the Timer task to open it.
2. On the General tab, enter a name for the Timer task.
3. Click the Timer tab to specify when the Integration Service starts the next task in the workflow.
4. Specify one of the following attributes for Absolute Time or Relative Time:
  - a. Absolute Time: Specify the exact time to start.
  - b. Absolute Time: Use this workflow date-time variable to calculate the wait.
  - c. Relative time: Start after.
  - d. Relative time: from the start time of this task.
  - e. Relative time: from the start time of the parent workflow/worklet.
  - f. Relative time: from the start time of the top-level workflow.

## Email Task

You can send email during a workflow using the Email task on the Workflow Manager. You can create reusable Email tasks in the Task Developer for any type of email.

Use Email tasks in any of the following locations:

- **Session properties.** You can configure a session to send email it completes or fails.
- **Workflow properties.** You can configure a workflow to send email when it is interrupted.
- **Workflows or worklets.** You can include an Email task anywhere in a workflow or worklet to send email based on a condition you define.

*To Create an Email Task:*

1. In the Task Developer, click Tasks > Create. Select an Email task and enter a name for the task. Click Create.
2. Double-click the Email task in the workspace. Click the Properties tab.

3. Enter the fully qualified email address of the mail recipient, subject of the email, and the Email Text. Click OK twice.

Attribute	Value
Email User Name	\$PNSuccessEmailUser
Email Subject	sessions successful
Email Text	Session completed

**Email Text**

Email text

## Sessions

A session is a set of instructions that tells the Integration Service how and when to move data from sources to targets. A session is a type of task, similar to other tasks available in the Workflow Manager.

You can also select options to run pre-session shell commands, send On-Success or On-Failure email, and use FTP to transfer source and target files. You can configure the session to override parameters established in the mapping, such as source and target location, source and target type, error tracing levels, and transformation attributes. You can also configure the session to collect performance details for the session and store them in the PowerCenter repository. You might view performance details for a session to tune the session.

To run a session, you must first create a workflow to contain the Session task. You can run as many sessions in a workflow as you need. You can run the Session tasks sequentially or concurrently, depending on the requirement.

### Creating a Session Task

You create a Session task for each mapping you want the Integration Service to run. You can create a reusable Session task in the Task Developer. You can also create non-reusable Session tasks in the Workflow Designer as you develop the workflow.

*To Create a Session Task:*

1. In the Workflow Designer, click the Session Task icon on the Tasks toolbar.
2. Enter a name for the Session task. Click Create.
3. Select the mapping you want to use in the session task. Click Ok. Click Done.

### Editing a Session Task

You can edit session properties at any time. The repository updates the session properties immediately. If the session is running when you edit the session, the repository updates the session when the session completes. If the mapping changes, the Workflow Manager might issue a warning that the session is invalid. The Workflow Manager then lets you continue editing the session properties. After you edit the session properties, the Integration Service validates the session and reschedules the session.

Double-click the Session task to open the session properties. The session has the following tabs, and each of those tabs has multiple settings:

- **General tab.** Enter session name, mapping name, and description for the Session task, assign resources, and configure additional task options.

- **Properties tab.** Enter session log information, test load settings, and performance configuration.
- **Config Object tab.** Enter advanced settings, log options, and error handling configuration.
- **Mapping tab.** Enter source and target information, override transformation properties, and configure the session for partitioning.
- **Components tab.** Configure pre- or post-session shell commands and emails.
- **Metadata Extension tab.** Configure metadata extension options.

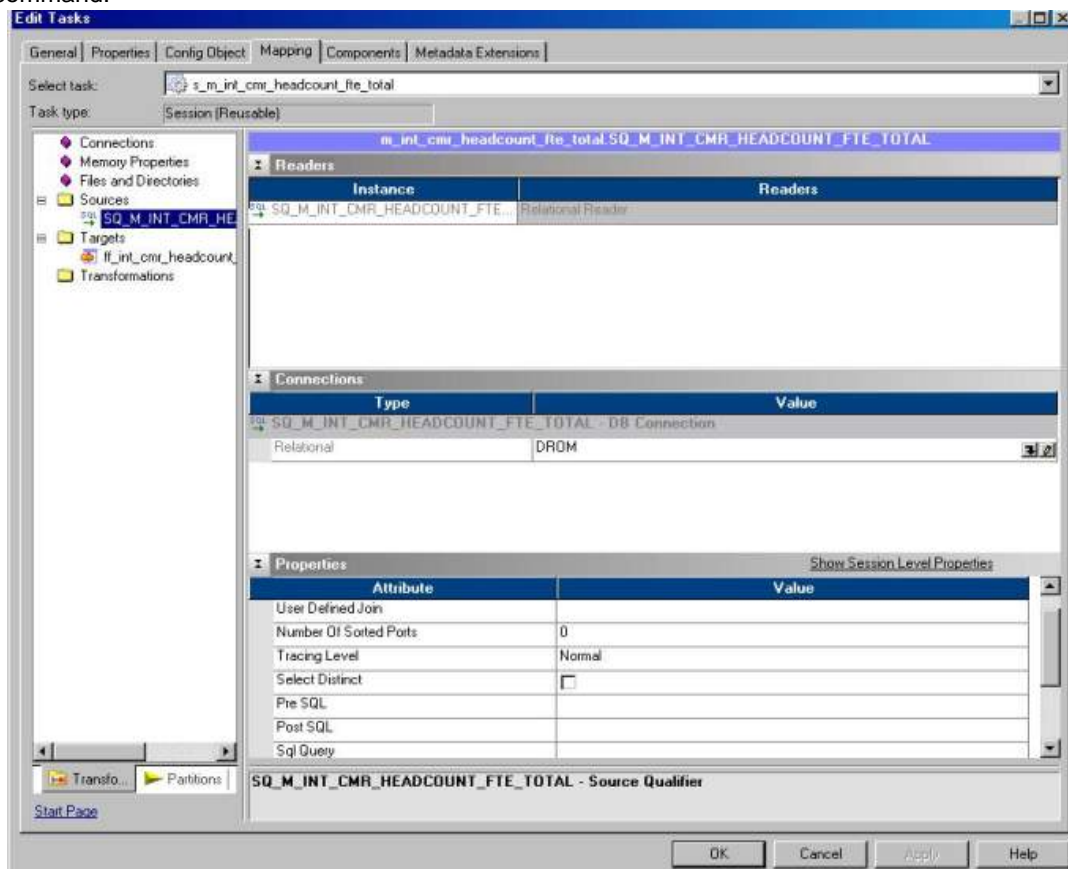
### Using Pre- and Post-Session SQL Commands

You can specify pre- and post-session SQL in the Source Qualifier transformation and the target instance when you create a mapping. When you create a Session task in the Workflow Manager you can override the SQL commands on the Mapping tab. The Integration Service runs pre-session SQL commands before it reads the source. It runs post-session SQL commands after it writes to the target.

Remember the following guidelines when creating the SQL statements:

- Use any command that is valid for the database type. However, the Integration Service does not allow nested comments, even though the database might.
- Use mapping parameters and variables in SQL executed against the source, but not the target.
- Use a semicolon (;) to separate multiple statements.
- The Integration Service ignores semicolons within /\* ... \*/.
- If you need to use a semicolon outside of comments, you can escape it with a backslash (\).
- The Workflow Manager does not validate the SQL.

You can configure error handling on the Config Object tab. You can choose to stop or continue the session if the Integration Service encounters an error issuing the pre- or post- session SQL command.





## Using Pre- and Post- Session Shell Commands

The Integration Service can perform shell commands at the beginning of the session or at the end of the session. Shell commands are operating system commands.

The Workflow Manager provides the following types of shell commands for each Session task:

- **Pre-session command.** The Integration Service performs pre-session shell commands at the beginning of a session. You can configure a session to stop or continue if a pre-session shell command fails.
- **Post-session success command.** The Integration Service performs post-session success commands only if the session completed successfully.
- **Post-session failure command.** The Integration Service performs post-session failure commands only if the session failed to complete.

Use the following guidelines to call a shell command:

- Use any valid UNIX command or shell script for UNIX nodes, or any valid DOS or batch file for Windows nodes.
- Configure the session to run the pre- or post-session shell commands.

The Workflow Manager lets you choose from the following options when you configure shell commands:

- **Create non-reusable shell commands.** Create a non-reusable set of shell commands for the session. Other sessions in the folder cannot use this set of shell commands.
- **Use an existing reusable Command task.** Select an existing Command task to run as the pre- or post-session shell command.

You can include any service process variable, such as \$PMTARGETFILEDIR, or session parameters in commands in pre-session and post-session commands. When you use a service process variable instead of entering a specific directory, you can run the same workflow on different Integration Services without changing session properties. You cannot use service process variables or session parameters in standalone Command tasks in the workflow. The Integration Service does not expand service process variables or session parameters used in standalone Command tasks.

You can configure the session to stop or continue if a pre-session shell command fails. If you select stop, the Integration Service stops the session, but continues with the rest of the workflow. If

Task	Type	Value
Pre-Session Command	None	
Post-Session Success Command	None	
Post-Session Failure Command	None	
On Success E-Mail	None	
On Failure E-Mail	None	



you select Continue, the Integration Service ignores the errors and continues the session. By default the Integration Service stops the session upon shell command errors. Configure the session to stop or continue if a pre-session shell command fails in the Error Handling settings on the Config Object tab.

### Using Post-Session Email

The Integration Service can send emails after the session completes. You can send an email when the session completes successfully. Or, you can send an email when the session fails. The Integration Service can send the following types of emails for each Session task:

- **On-Success Email.** The Integration Service sends the email when the session completes successfully.
- **On-Failure Email.** The Integration Service sends the email when the session fails.

If you have already created a reusable Email task, you can select it as the On-Success or On-Failure email for the session. Or, you can create non-reusable emails that exist only within the Session task.

### Validating a Session

The Workflow Manager validates a Session task when you save it. You can also manually validate Session tasks and session instances. Validate reusable Session tasks in the Task Developer. Validate non-reusable sessions and reusable session instances in the Workflow Designer.

The Workflow Manager marks a reusable session or session instance invalid if you perform one of the following tasks:

- Edit the mapping in a way that might invalidate the session. You can edit the mapping used by a session at any time. When you edit and save a mapping, the repository might invalidate sessions that already use the mapping. You must reconnect to the folder to see the effect of mapping changes on Session tasks. When you edit a session based on an invalid mapping, the Workflow Manager displays a warning message:  
*The mapping [mapping\_name] associated with the session [session\_name] is invalid.*
- Delete a database, FTP, or external loader connection used by the session.
- Leave session attributes blank. For example, the session is invalid if you do not specify the source file name.
- Change the code page of a session database connection to an incompatible code page.
- If you delete objects associated with a Session task such as session configuration object, Email, or Command task, the Workflow Manager marks a reusable session invalid. However, the Workflow Manager does not mark a non-reusable session invalid if you delete an object associated with the session.
- If you delete a shortcut to a source or target from the mapping, the Workflow Manager does not mark the session invalid.
- The Workflow Manager does not validate SQL overrides or filter conditions entered in the session properties when you validate a session. You must validate SQL override and filter conditions in the SQL Editor.
- If a reusable session task is invalid, the Workflow Manager displays an invalid icon over the session task in the Navigator and in the Task Developer workspace. This does not affect the validity of the session instance and the workflows using the session instance.
- If a reusable or non-reusable session instance is invalid, the Workflow Manager marks it invalid in the Navigator and in the Workflow Designer workspace. Workflows using the session instance remain valid.

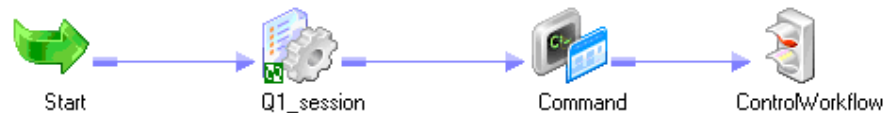
*To validate a session:*

1. Select the session in the workspace and Click Tasks > Validate.
2. Or, Right Click the session in the workspace and Choose Validate.

## Workflows

A workflow is a set of instructions that tells the Integration Service how to run tasks such as sessions, email notifications, and shell commands. After you create tasks in the Task Developer and Workflow Designer, you connect the tasks with links to create a workflow.

In the Workflow Designer, you can specify conditional links and use workflow variables to create branches in the workflow. The Workflow Manager also provides Event-Wait and Event-Raise tasks to control the sequence of task execution in the workflow. You can also create worklets and nest them inside the workflow. Every workflow contains a Start task, which represents the beginning of the workflow.



Use the following guidelines when you develop a workflow:

- **Create a workflow.** Create a workflow in the Workflow Designer.
- **Add tasks to the workflow.** You might have already created tasks in the Task Developer. Or, you can add tasks to the workflow as you develop the workflow in the Workflow Designer.
- **Connect tasks with links.** After you add tasks to the workflow, connect them with links to specify the order of execution in the workflow.
- **Specify conditions for each link.** You can specify conditions on the links to create branches and dependencies.
- **Validate workflow.** Validate the workflow in the Workflow Designer to identify errors.
- **Save workflow.** When you save the workflow, the Workflow Manager validates the workflow and updates the repository.
- **Run workflow.** In the workflow properties, select an Integration Service to run the workflow. Run the workflow from the Workflow Manager, Workflow Monitor, or pmcmd. You can monitor the workflow in the Workflow Monitor.

### Assigning Integration Service

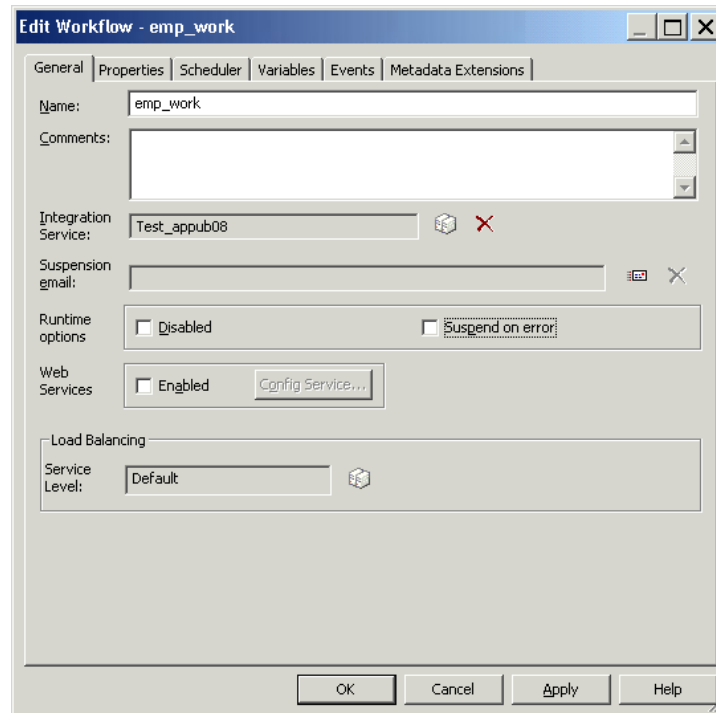
Before you can run a workflow, you must assign an Integration Service to run it. You can choose an Integration Service to run a workflow by editing the workflow properties. You can also assign an Integration Service from the menu. When you assign a service from the menu, you can assign multiple workflows without editing each workflow.

*To select an Integration Service to run a workflow:*

1. In the Workflow Designer, open the Workflow.
2. Click Workflows > Edit. The Edit Workflow dialog box appears.
3. On the General tab, click the Browse Integration Services button. A list of Integration Services appears.
4. Select the Integration Service that you want to run the workflow.
5. Click OK twice to select the Integration Service for the workflow.

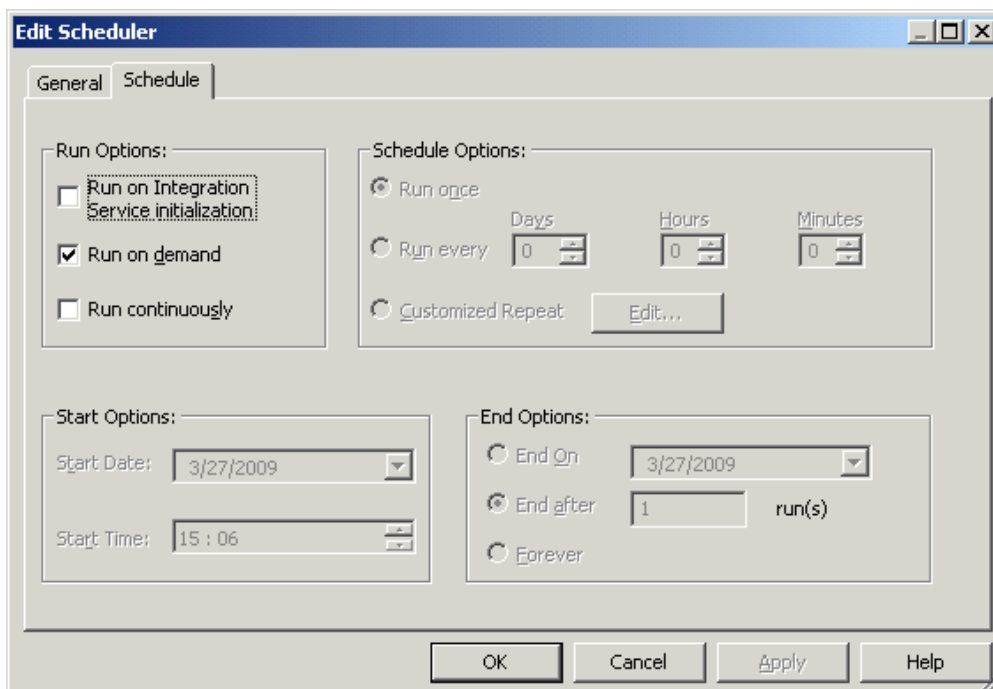
The Integration Service does not run the workflow if:

- The prior workflow run fails.
- You remove the workflow from the scheduler.
- Integration Service is running in safe mode.



### Scheduling a Workflow

You can schedule a workflow to run continuously, repeat at a given time or interval, or you can manually start a workflow. The Integration Service runs a scheduled workflow as configured. By default, the workflow runs on demand. You can change the schedule settings by editing the scheduler. If you change, the Integration Service reschedules the workflow according to the new settings. Each workflow has an associated scheduler. You can create a non-reusable scheduler for the workflow. Or, you can create a reusable scheduler to use the same set of schedule settings for workflows in the folder. The Workflow Manager marks a workflow invalid if you delete the scheduler associated with the workflow. Configure the Schedule tab of the scheduler to set run options, schedule options, start options, and end options for the schedule.



### Try It Out

---

5. Create a Relational Database Connection and assign execute permission to users on the connection object.
6. Create a Workflow, include non-reusable sessions and use decision task.
7. Create Reusable Email Tasks and include them in the sessions as post session failure emails.
8. Schedule the workflow to run on 5 weekdays at a specified time.

### Summary

---

- ❖ The tools in Workflow Manager are Task Developer, Workflow Designer, Worklet Designer.
- ❖ Different types of tasks are Assignment, Command, Control, Decision, Email, Event Raise, Event Wait, Session and Timer.
- ❖ A session is a set of instructions that tells the PowerCenter Server how and when to move data from sources to targets.
- ❖ A workflow is a set of instructions that describes how and when to run tasks related to extracting, transforming, and loading data.

### Test your Understanding

---

1. What is a workflow?
2. What are the components of Workflow Manager?
3. How to create tasks, sessions, workflows, and worklets?

## Session 7: Workflow Monitor

### Learning Objectives

---

- Monitor Overview
- Workflow Monitor Views
- Running and Monitoring Workflows
- Scheduling
- Workflow and Session Logs
- Monitor Configuration

### Monitor Overview

---

The Workflow Monitor is a tool that allows you to monitor workflows and tasks. You can view details about a workflow or task in either Gantt Chart view or Task view. You can run, stop, abort, and resume workflows from the Workflow Monitor. The monitor continuously receives information from the PowerCenter Server and Repository Server. It also fetches information from the repository to display historic information. You can view dynamic information about workflow runs by connecting to an Integration Service of the Repository. You can monitor multiple repositories, Integration Services, and workflows at the same time. The Workflow Monitor displays workflows that have run at least once.

*You can open the Workflow Monitor in the following ways:*

- From the Windows Start menu.
- From the Workflow Manager Navigator.
- Configure the Workflow Manager to open the Workflow Monitor when you run a workflow from the Workflow Manager.
- Click Tools > Workflow Monitor from the Designer, Workflow Manager, or Repository Manager. Or, click the Workflow Monitor icon on the Tools toolbar. When you use a Tools button to open the Workflow Monitor, PowerCenter uses the same repository connection to connect to the repository and opens the same folders.

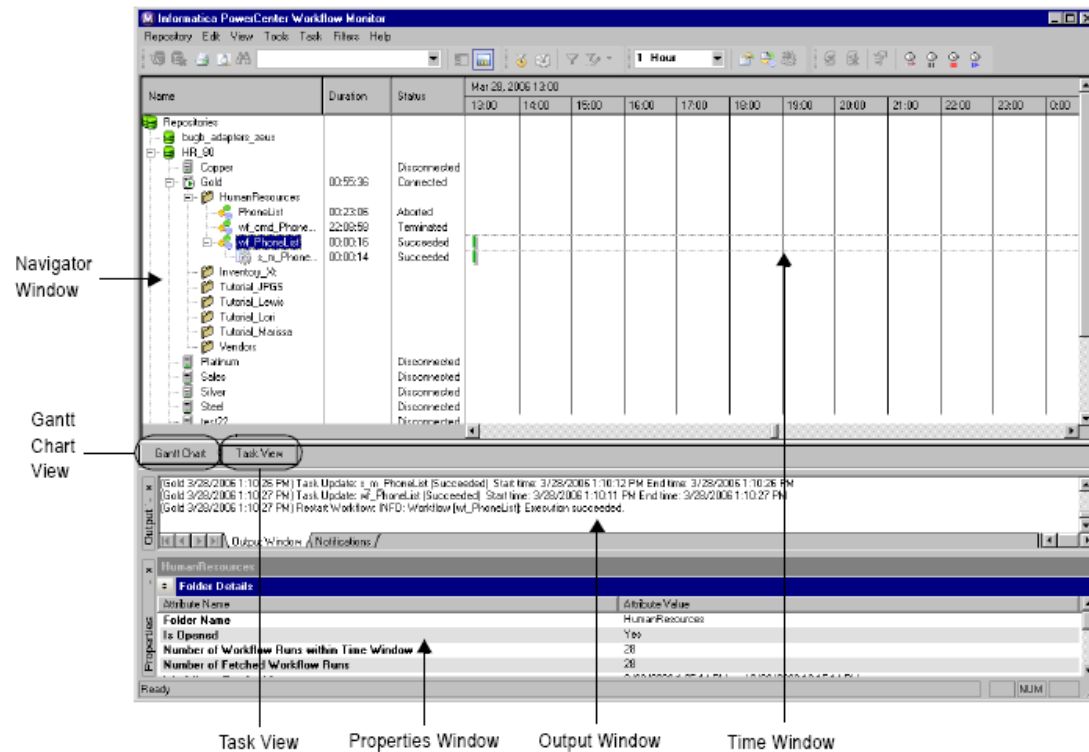
### Connecting to Repositories

When you open the Workflow Monitor, you must connect to a repository to monitor the objects in it. Connect to repositories by clicking Repository > Connect. Enter the repository name and connection information. On connecting to a repository, the Workflow Monitor displays a list of Integration Services available for the repository.

### Connecting to Integration Services

To monitor tasks and workflows that run on an Integration Service, you must connect to the Integration Service. To connect to an Integration Service, right-click it and choose Connect. When you connect to an Integration Service, you can view all folders that you have read permission on. You can disconnect from an Integration Service by right-clicking it and selecting Disconnect. When you disconnect from an Integration Service, or when the Workflow Monitor cannot connect to an Integration Service, the Workflow Monitor displays disconnected for the Integration Service status. You can also open an Integration Service node in the Navigator without connecting to it. When you open an Integration Service, the Workflow Monitor gets workflow run information stored in the repository. It does not get dynamic workflow run information from currently running workflows.

The Workflow Monitor consists of the following windows:



- **Navigator window.** Displays monitored repositories, Integration Services, and repository objects.
- **Output window.** Displays messages from the Integration Service and the Repository Service.
- **Properties window.** Displays details about services, workflows, worklets, and tasks.
- **Time window.** Displays progress of workflow runs.

## Workflow Monitor Views

### Gantt Chart View

You can view chronological details of workflow runs with the Gantt Chart view.

The Gantt Chart view displays the following information:

- **Task name.** Name of the task in the workflow.
- **Duration.** The length of time the Integration Service spends running the most recent task or workflow.
- **Status.** The status of the most recent task or workflow.
- **Connection between objects.** The Workflow Monitor shows links between objects in the Time window.

### Task View

The Task view displays information about workflow runs in a report format. The Task view provides a convenient way to compare and filter details of workflow runs.

Task view displays the following information:

- **Workflow run list.** The list of workflow runs. The workflow run list contains folder, workflow, worklet, and task names. The Workflow Monitor displays workflow runs chronologically with the most recent run at the top. It displays folders and Integration Services alphabetically.

- **Status message.** Message from the Integration Service regarding the status of the task or workflow.
- **Run type.** The method you used to start the workflow. You might manually start the workflow or schedule the workflow to start.
- **Node.** Node of the Integration Service that ran the task.
- **Start time.** The time that the Integration Service starts executing the task or workflow.
- **Completion time.** The time that the Integration Service finishes executing the task or workflow.
- **Status.** The status of the task or workflow.

## Running and Monitoring Workflows

The Workflow Monitor displays workflows that have run at least once. In the Workflow Monitor, you can run a workflow or any task or worklet in the workflow.

### Running a Task, Workflow, or Worklet

*To run a Workflow:*

1. In the Navigator, select the workflow you want to run.
2. Right-click the workflow in the Navigator and choose Restart.

*To run a particular Task alone:*

1. In the Navigator, select the workflow you want to run.
2. Right-click the workflow and choose Restart Task.

*To run a part of a workflow:*

1. In the Navigator, select the workflow you want to run.
2. Right-click the workflow and choose Restart Workflow from Task.

Restart Workflow from Task  
Restart Task  
Recover Workflow from Task  
Recover Task  
Stop  
Abort

Go to Previous Run  
Go to Next Run  
Go to Task View

Get Session Log  
Show History Names...  
Get Run Properties

**Resume a suspended workflow.** In the workflow properties, you can choose to suspend the workflow or worklet if a session fails. After you fix the errors that caused the session to fail, recover the workflow in the Workflow Monitor.

*To recover a workflow or worklet:*

1. In the Navigator, select the workflow you want to run.
2. Right-click the workflow or worklet and choose Recover.

**Stop or abort a task or workflow.** You can stop or abort a task, worklet, or workflow in the Workflow Monitor at any time. When you stop a task in the workflow, the Integration Service stops processing that task and all other tasks in its path. However, the Integration Service continues running concurrent tasks. If the Integration Service cannot stop processing the task, you need to abort the task. When the Integration Service aborts a task, it kills the DTM process and terminates the task.

*To stop or abort tasks, worklets, or workflows:*

1. In the Navigator, select the task, workflow, or worklet you want to stop or abort.
2. Right-click the task, workflow, or worklet in the Navigator and choose Stop or Abort.

When a Task is stopped, the Integration Service stops reading data but continues processing, writing and committing data to target. If the Integration Service does not complete processing the data for a long time, you have to abort the task.

When a Task is aborted, the Integration Service stops reading data and continues processing, writing and committing data to target. If the processing is not completed within 60 seconds, it kills the DTM process and terminates the task.



## Scheduling Workflows

You can schedule and unschedule workflows in the Workflow Monitor. You can schedule any workflow that is not configured to run on demand. When you try to schedule a run on demand workflow, the Workflow Monitor displays an error message in the Output window. When you schedule an unscheduled workflow, the workflow uses its original schedule specified in the workflow properties.

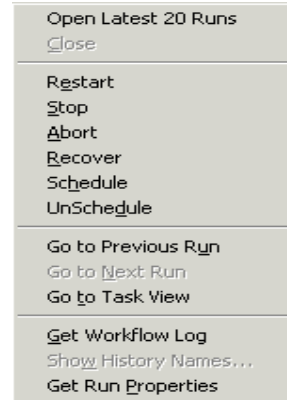
*To Schedule an unscheduled workflow:*

1. In the Navigator, select the workflow to be scheduled.
2. Right-click the workflow and choose Schedule.

*To Unschedule a scheduled workflow:*

1. In the Navigator, select the workflow to be unscheduled.
3. Right-click the workflow and choose Unschedule.

Workflow Monitor displays the workflow status as Scheduled/Unscheduled and displays a message in the Output window. If you want to specify a different schedule for the workflow, you must edit the scheduler in the Workflow Manager.



## Workflow and Session Logs

You can view session and workflow logs from the Workflow monitor. By default the most recent session or workflow log file opens in the Log Viewer.

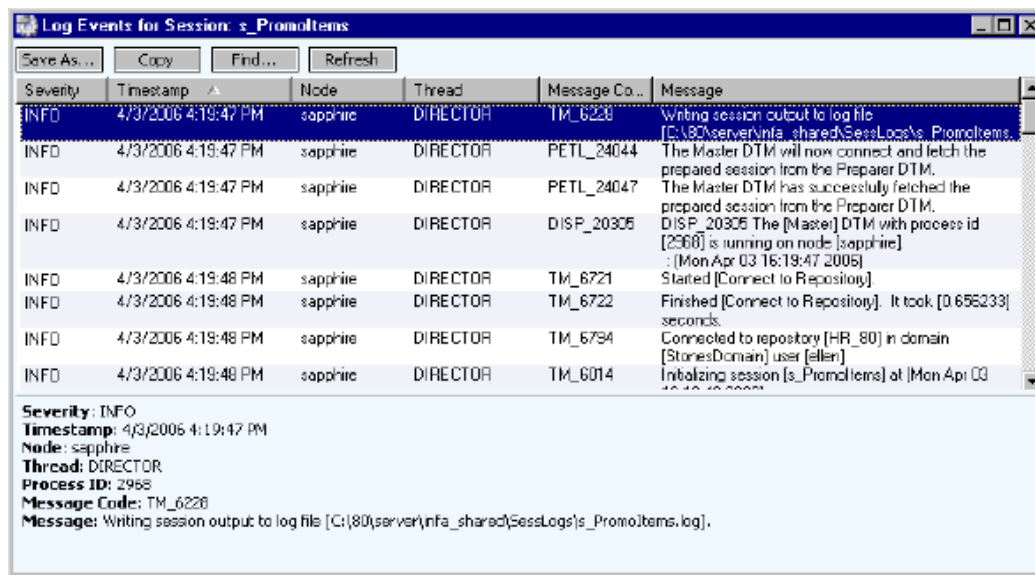
*To view a session or workflow log file:*

1. Right-click a session or workflow in the Navigator or Time window.
2. Choose Get Session Log or Get Workflow Log.

If the workflow has been configured to save the logs, you can view the logs of previous runs.

*To view the logs of a session or workflow's previous runs:*

1. Right-click the chart of the previous run in the Navigator window of Gantt Chart View or the workflow/session name in the Task View
2. Choose Get Session Log or Get Workflow Log



A Sample Session Log



Log Events for Workflow: wf_EVENT_HCDB_01					
<div> <div>Save As...</div> <div>Copy</div> <div>End...</div> <div>Refresh</div> </div>					
Severity	Timestamp	Node	Thread	Message Co...	Message
INFO	2/24/2009 9:34:13 AM	EDC1_SIT_NC	7454	LM_36435	Starting execution of workflow [wf_EVENT_HCDB_01] in folder [EDW_EVENT] last saved by user [u493586].
INFO	2/24/2009 9:34:13 AM	EDC1_SIT_NC	7454	LM_44195	Workflow [wf_EVENT_HCDB_01] service level [SLPriority:5.SLDispatchWaitTime:1800].
INFO	2/24/2009 9:34:13 AM	EDC1_SIT_NC	7454	LM_36330	Start task instance [Start]. Execution started.
INFO	2/24/2009 9:34:13 AM	EDC1_SIT_NC	7454	LM_36318	Start task instance [Start]. Execution succeeded.
INFO	2/24/2009 9:34:13 AM	EDC1_SIT_NC	7454	LM_36505	Link [Start -> s_m_OPEN_wf_EVENT_HCDB_01] empty expression string, evaluated to TRUE.
INFO	2/24/2009 9:34:13 AM	EDC1_SIT_NC	7454	LM_36388	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01] is waiting to be started.
INFO	2/24/2009 9:34:13 AM	EDC1_SIT_NC	7454	LM_36682	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: started a process with pid [692274] on node [EDC1_SIT_N01].
INFO	2/24/2009 9:34:13 AM	EDC1_SIT_NC	7454	LM_36330	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: Execution started.
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	7968	VAR_27085	Parameter file [/wamu/DAI/Informatica/SrcFiles/ParFiles/wf_EDW_HCDB_SCHED_02.par] is opened for [session [wf_EVENT_HCDB_01.s_m_OPEN_wf_EVENT_HCDB_01]].
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	7968	VAR_27062	Warning! Cannot find section for workflow [wf_EVENT_HCDB_01] and folder [EDW_EVENT] in parameter file [/wamu/DAI/Informatica/SrcFiles/ParFiles/wf_EDW_HCDB_SCHED_02.par].
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [TM_6793 Fetching initialization properties from the Integration Service : [Mon Feb 23 22:04:14 2009]].
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [DISP_20305 The [Preparer] DTM with process id [692274] is running on node [EDC1_SIT_N01]. : [Mon Feb 23 22:04:14 2009]].
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [PETL_24037 Finished the prepare phase for the session.]
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [TM_6722 Finished [Fetch Session from Repository]. It took [0.31069] seconds.]
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [TM_6721 Started [Fetch Session from Repository].]
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [TM_6721 Started [Connect to Repository].]
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [VAR_27028 Use override value [] for mapping parameter [\$\$\$TRCT_NM]].
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [VAR_27028 Use override value [wf_EVENT_HCDB_01] for mapping parameter [\$\$\$WKFL_TASK_NM]].
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [TM_6722 Finished [Connect to Repository]. It took [0.38065] seconds.]
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [VAR_27028 Use override value [7013] for mapping parameter [\$\$\$WKFL_TASK_ID]].
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [TM_6722 Finished [Partition Group Formation]. It took [0.02265] seconds.]
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [TM_6792 Notifying the Integration Service that the prepare phase has been completed. : [Mon Feb 23 22:04:15 2009]].
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [TM_6228 Writing session output to log file [/wamu/DAI/Informatica/SessionLogs/s_m_OPEN_wf_EVENT_HCDB_01.log]].
INFO	2/24/2009 9:34:15 AM	EDC1_SIT_NC	8739	LM_36488	Session task instance [s_m_OPEN_wf_EVENT_HCDB_01]: [TM_6793 Fetching initialization properties from the
<b>Severity:</b> INFO <b>Timestamp:</b> 2/24/2009 9:34:13 AM <b>Node:</b> EDC1_SIT_N01 <b>Thread:</b> 7454 <b>Process ID:</b> 561390 <b>Message Code:</b> LM_36435 <b>Message:</b> Starting execution of workflow [wf_EVENT_HCDB_01] in folder [EDW_EVENT] last saved by user [u493586].					

## A Sample Workflow Log

## Monitor Configuration

You can configure how the Workflow Monitor displays general information, workflows, and tasks. You can configure general tasks such as the maximum number of days or runs that the Workflow Monitor appears. You can also configure options specific to Gantt Chart and Task view.

Click Tools > Options to configure Workflow Monitor options.

You can configure the following options in the Workflow Monitor:

- **General.** Customize general options such as the maximum number of workflow runs to display, whether to receive messages from the Workflow Manager and Repository Services.
- **Gantt Chart view.** Configure Gantt Chart view options such as workspace color, status colors, and time format.
- **Task view.** Configure which columns to display in Task view.
- **Advanced.** Configure advanced options such as the number of workflow runs the Workflow Monitor holds in memory for each Integration Service, whether to expand running workflows automatically, whether to expand the workflow runs when opening the latest runs.

### Try It Out

---

4. Run a workflow from the Workflow Monitor manually.
5. Check the session log of a session.
6. Configure the options in General Tab and display the workflow runs in Gantt Chart and Task Views.

### Summary

---

- ❖ The Workflow Monitor is a tool that allows you to monitor workflows and tasks.
- ❖ The monitor continuously receives information from the PowerCenter Server and Repository Server. It can be used to monitor multiple repositories, Integration Services, and workflows at the same time.
- ❖ Workflow Monitor consists of two views to monitor the tasks – Gantt Chart View and Task View.
- ❖ You can run, stop, abort, and resume workflows from the Workflow Monitor.
- ❖ The workflows that have been configured with Schedulers in Workflow Manager can be scheduled or unscheduled in Workflow Monitor.
- ❖ The session and workflow logs can be viewed for the session and workflow runs.

### Test your Understanding

---

1. What is Workflow Monitor?
2. What is Gantt Chart View?
3. What is Task View?
4. How is a Workflow scheduled/unscheduled?
5. How is a Task run, stopped, aborted and recovered?
6. How is a session log obtained?

## Session 8: Reusable Objects

### Learning Objectives

- Reusable Transformations
- Maplets
- Worklets
- Reusable Workflow Tasks

### Reusable Transformations

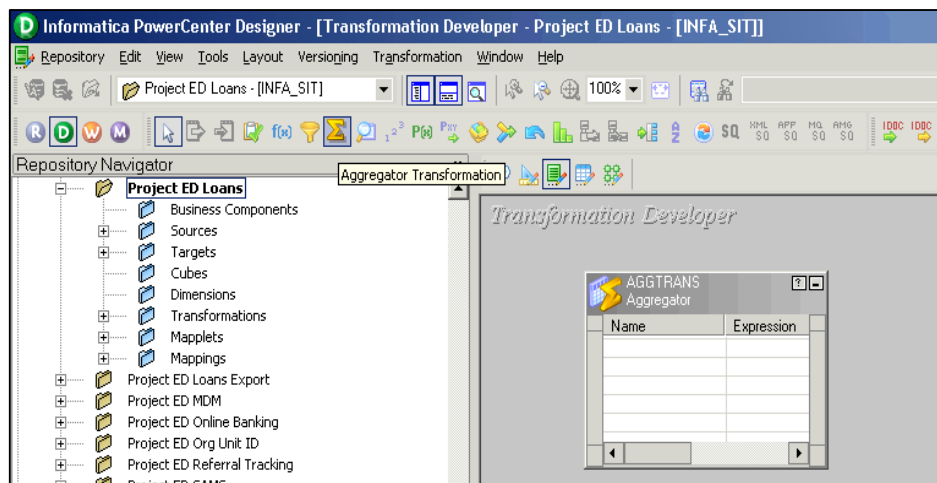
Mappings can contain reusable and non-reusable transformations. Non-reusable transformations exist within a single mapping. Reusable transformations can be used in multiple mappings. For example, you can create a reusable Aggregator transformation to perform the same aggregate calculations in multiple mappings. You can create most transformations as a non-reusable or reusable. However, you can create the External Procedure transformation as a reusable transformation only.

### Instances and Inherited Changes

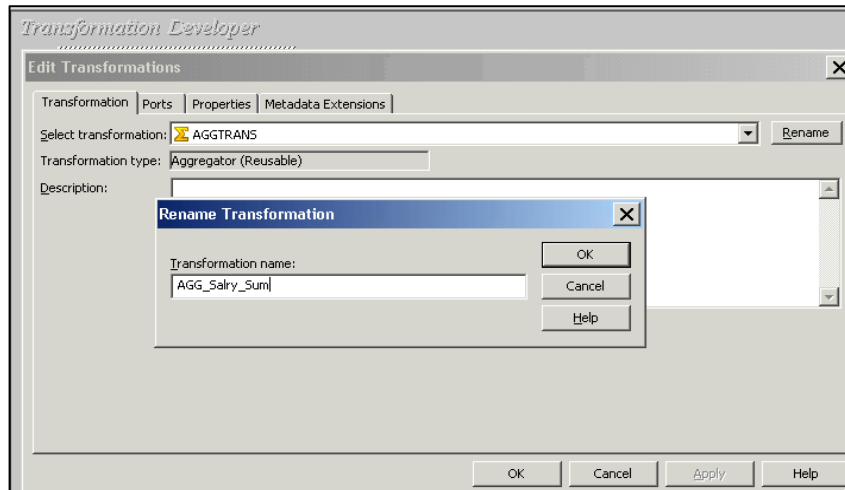
When you add a reusable transformation to a mapping, the definition of the transformation exists outside the mapping, while a copy (or instance) appears within the mapping. When you change the transformation in the Transformation Developer, its instances reflect these changes. Instead of updating the transformation instances in every mapping, you can update the reusable transformation once, and all instances of the transformation inherit the change. However you can change the properties for a particular instance in a mapping without affecting the reusable transformations and other instances in other mappings.

### Designing a Reusable Transformation in the Transformation Developer.

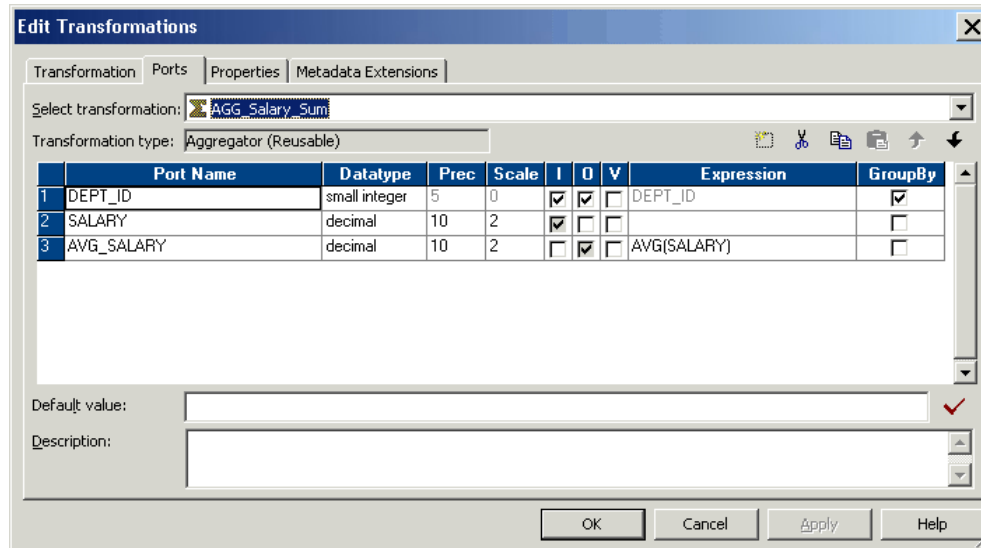
1. In the Designer, switch to the Transformation Developer.
2. Click the button on the Transformation toolbar corresponding to the type of transformation you want to create.
3. Drag within the workbook to create the transformation.



4. Double-click the transformation title bar to open the dialog displaying its properties.
5. Click the Rename button and enter a descriptive name for the transformation
6. Click OK



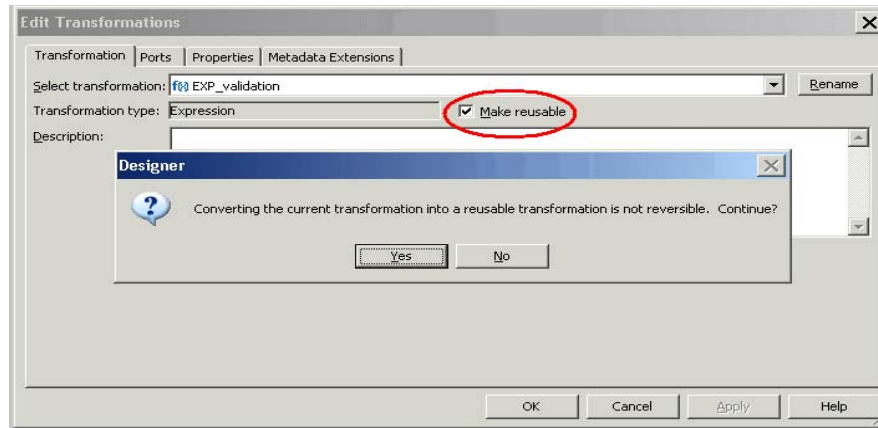
7. Click the Ports tab, then add any input and output ports you need for this transformation.



6. Set the other properties of the transformation, and click OK.
8. Click Repository > Save.

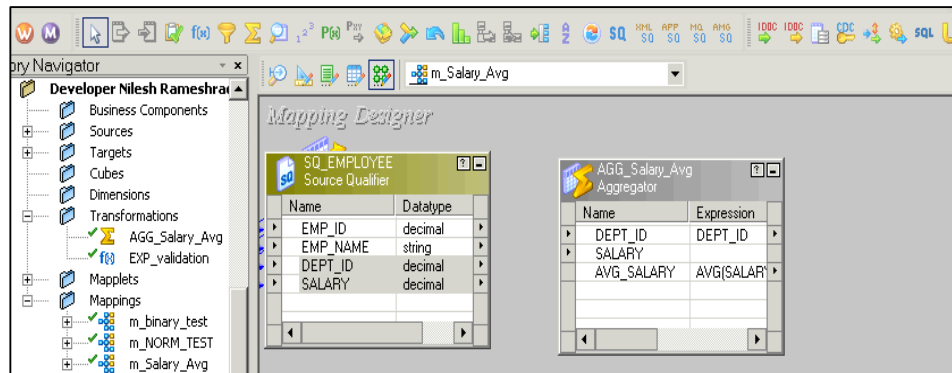
#### Promoting a non-reusable transformation from the Mapping Designer

1. In the Designer, open a mapping and double-click the title bar of the transformation you want to promote.
2. Select the Make Reusable Check Box.
3. When prompted whether you are sure you want to promote the transformation, click Yes.
4. Click OK to return to the mapping.
5. Click Repository > Save.



### Adding Reusable Transformations to Mappings

1. In the Designer, switch to the Mapping Designer.
2. Open or create a mapping.
3. In the list of repository objects, drill down until you find the reusable transformation you want in the Transformations section of a folder.
4. Drag the transformation from the Navigator into the mapping. A copy (or instance) of the reusable transformation appears.



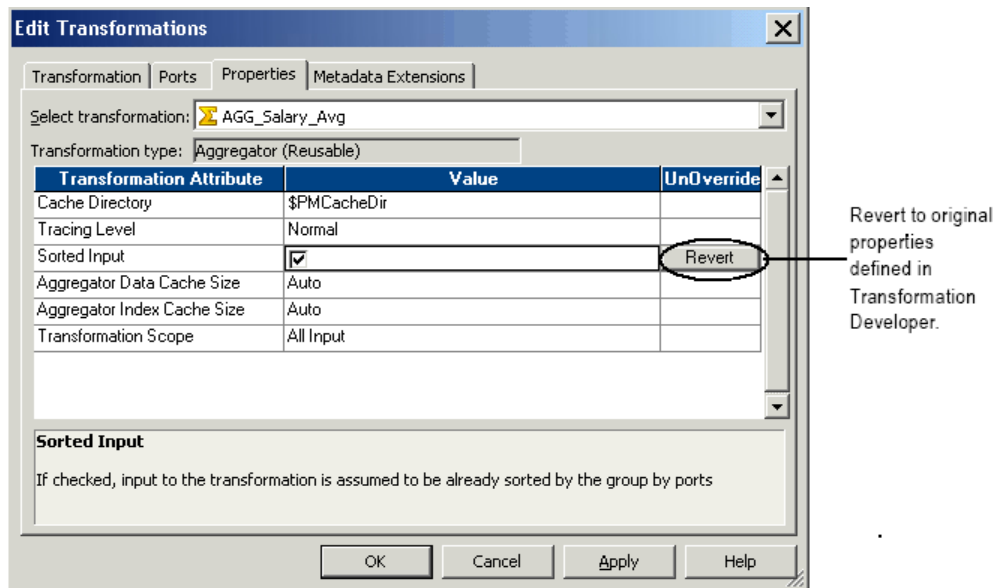
5. Link the new transformation to other transformations or target definitions.
6. Click Repository > Save.

### Modifying a Reusable Transformation

Changes to a reusable transformation that you enter through the Transformation Developer are immediately reflected in all instances of that transformation. You risk invalidating mappings when you modify a reusable transformation. Any of the following changes to the reusable transformation, mappings that use instances of it may be invalidated:

- When you delete a port or multiple ports in a transformation, you disconnect the instance from part or all of the data flow through the mapping.
- When you change a port data type, you make it impossible to map data from that port to another port using an incompatible data type.
- When you change a port name, expressions that refer to the port are no longer valid.
- When you enter an invalid expression in the reusable transformation, mappings that use the transformation are no longer valid. The Integration Service cannot run sessions based on invalid mappings.

## Reverting to Original Reusable Transformation



The changes to a reusable transformation in mapping can be reverted to the original reusable transformation properties by clicking the Revert button.

## Mapplets

A mapplet is a reusable object created in the Mapplet Designer. It contains a set of transformations and transformation logic that can be reused in multiple mappings. For example, if you have several fact tables that require a series of dimension keys, create a mapplet containing a series of Lookup transformations to find each dimension key. You can then use the mapplet in each fact table mapping, rather than recreate the same lookup logic in each mapping. When you use a mapplet in a mapping, you use an instance of the mapplet. Like a reusable transformation, any change made to the mapplet is inherited by all instances of the mapplet.

Mapplets help simplify the mappings in the following ways:

- **Include source definition.** Use multiple source definitions and source qualifiers to provide source data for a mapping.
- **Accept data from sources in a mapping.** Use an input transformation to receive source data from the mapping.
- **Include multiple transformations.** A mapplet can contain as many transformations as you need.
- **Pass data to multiple transformations.** Each output transformation in a mapplet represents one output group in a mapplet, to feed data to multiple transformations in the mapping.
- **Contain unused ports.** Do not have to connect all mapplet input and output ports in a mapping.

In addition to transformation logic, a mapplet has the following components:

### Mapplet Input

Mapplet input can originate from a source definition and/or from an Input transformation in the mapplet.

#### *Using Source Definitions for Mapplet Input*

Use one or more source definitions in a mapplet to provide source data. When you use such a mapplet in a mapping, it is the first object in the mapping pipeline and contains no input ports.

### Using Input Transformations for Maplet Input

Use an Input transformation in a maplet when you want the maplet to receive input from a source in a mapping. When you use the maplet in a mapping, the input transformation provides input ports so you can pass data through the maplet. Each port in the Input transformation connected to another transformation in the maplet becomes an input port of the maplet. Input transformations can receive data from a single active source. Unconnected ports do not display in the Mapping Designer. You can connect an Input transformation to multiple transformations in a maplet. However, you cannot connect a single port in the Input transformation to multiple transformations in the maplet.

### Maplet Output

Use an output transformation in a maplet to pass data through the maplet into a mapping. A maplet must contain at least one output transformation with at least one connected port in the maplet. Each connected port in an output transformation displays as a maplet output port in a

#### Maplet Designer

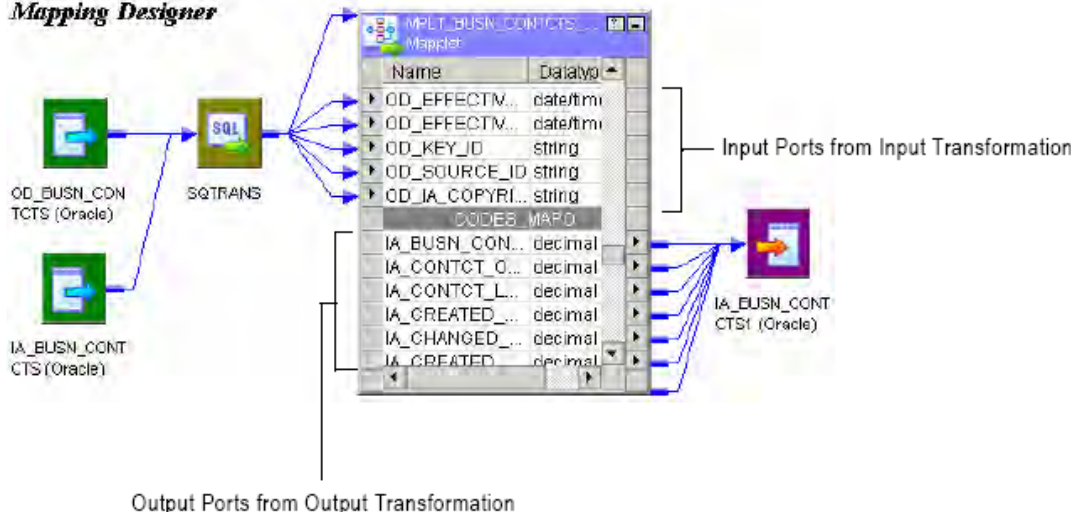


mapping. Each output transformation in a maplet displays as an output group in a mapping. An output group can pass data to multiple pipelines in a mapping.

### Maplet Port

Maplet ports display only in the Mapping Designer. Maplet ports consist of input ports from input transformations and output ports from output transformations. If a maplet uses source definitions rather than input transformations for input, it does not contain any input ports in the mapping.

#### Mapping Designer





## Creating a Mapplet

A mapplet can be active or passive depending on the transformations in the mapplet. Active Mapplets contain one or more active transformations. Passive Mapplets contain only passive transformations. When you use a mapplet in a mapping, all transformation rules apply to the mapplet depending on the mapplet type. For example, as with an active transformation, you cannot concatenate data from an active mapplet with a different pipeline.

*To create and configure a mapplet in the Mapplet Designer:*

1. **Create a mapplet.** Click Mapplets > Create from the menu in the Mapplet Designer. The recommended naming convention for mapplets is `mpltMappletName`.
2. **Create mapplet transformation logic.** Create and link transformations in the same manner as in a mapping.
3. **Create mapplet ports.**

## Validating Mapplets

The Designer automatically validates a mapplet when you save it. You can also validate a mapplet using the Mapplets > Validate menu command. The Designer validates the mapplet pipeline in the same way it validates a mapping. In addition, the Designer performs the following checks specific to mapplets:

- The mapplet should contain Input transformations and/or source definitions with at least one port connected to a transformation in the mapplet.
- The mapplet should contain at least one Output transformation with at least one port connected to a transformation in the mapplet.

## Editing Mapplets

You can edit a mapplet in the Mapplet Designer. When you save changes to a mapplet, all instances of the mapplet and all shortcuts to the mapplet inherit the changes. These changes may invalidate mappings that use the mapplet. You can make the following changes to a mapplet *without* affecting the validity of existing mappings and sessions:

- Add input or output ports.
- Change port names or comments.
- Change Input or Output transformation names or comments.
- Change transformation names, comments, or properties.
- Change port default values for transformations in the mapplet.
- Add or remove transformations in the mapplet, providing you do not change the mapplet type from active to passive or from passive to active.

To keep existing mappings valid when you edit a mapplet used in the mapping:

- *Do not delete a port from the mapplet.* The Designer deletes mapplet ports in the mapping when you delete links to an Input or Output transformation or when you delete ports connected to an Input or Output transformation.
- *Do not change the datatype, precision, or scale of a mapplet port.* The datatype, precision, and scale of a mapplet port is defined by the transformation port to which it is connected in the mapplet. Therefore, if you edit a mapplet to change the datatype, precision, or scale of a port *connected* to a port in an Input or Output transformation, you change the mapplet port
- *Do not change the mapplet type.* If you remove all active transformations from an active mapplet, the mapplet becomes passive. If you add an active transformation to a passive mapplet, the mapplet becomes active.

## Using Mapplets in Mappings

In a mapping, a mapplet has input and output ports that you can connect to other transformations in the mapping. You do not have to connect all mapplet ports in a mapping. However, if the mapplet contains an SQL override, you must connect all mapplet output ports in the mapping.

*To use a mapplet in a mapping:*

1. Open the mapping in Mapping Designer.



2. Drag the reusable maplet into the mapping designer.
3. If the maplet contains input ports, connect at least one maplet input port to a transformation in the mapping.
4. Connect at least one maplet output port to a transformation in the mapping.

### Creating and Configuring Maplet Ports

After creating transformation logic for a maplet, you can create maplet ports. Use an Input transformation to define maplet input ports if the maplet contains no source definitions. Use an Output transformation to create a group of output ports. Only connected ports in an Input or Output transformation become maplet input or output ports in a mapping. Unconnected ports do not display when you use the maplet in a mapping. You can view the datatype, precision, and scale of available maplet ports when you use the maplet in a mapping.

We can create a maplet port in the following ways:

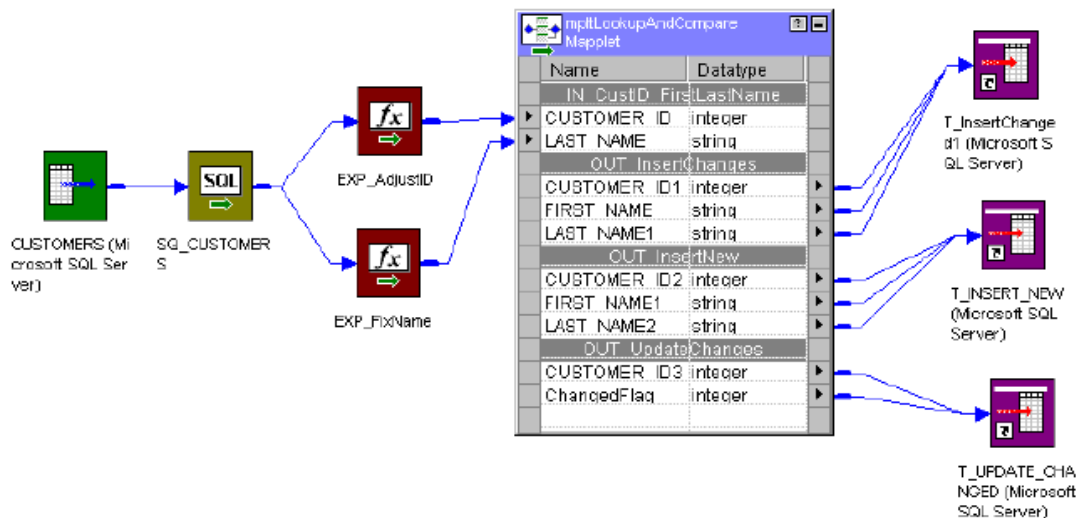
- **Manually create ports in the Input/output transformation.** You can create port names in Input and Output transformations. You can also enter a description for each port name. The port has no defined data type, precision, or scale until you connect it to a transformation in the maplet.
- **Drag a port from another transformation.** you can create an input or output port by dragging a port from another transformation into the Input or Output transformation. The new port inherits the port name, description, data type, and scale of the original port. you can edit the new port name and description in the transformation. If you change a port connection, the Designer updates the Input or Output transformation port to match the attributes of the new connection.

### Connecting to Maplet Input Ports

When using a maplet with input ports in a mapping, you connect the maplet input ports to the mapping pipeline. you can pass data into a maplet only when it originates from a single active transformation.

For example, in the figure shown below, the maplet *mpItLookupAndCompare* accepts data from two Expression transformations because data from both transformations originate from a single source qualifier. The Source Qualifier *SQ\_CUSTOMERS* is the active transformation providing maplet source data:

#### *Mapping Designer*



## Connecting to Mapplet Output Groups

Each Output transformation displays as an output group when you use a mapplet in a mapping. Connect the mapplet output ports to the mapping pipeline. Use Autolink to connect the ports.

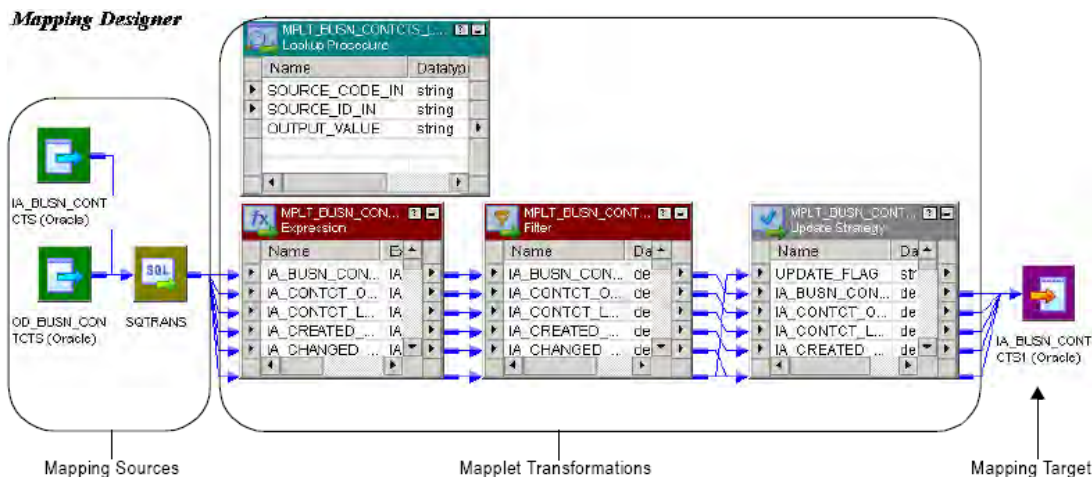
Use the following rules and guidelines when you connect mapplet output ports in the mapping:

- When a mapplet contains a source qualifier that has an override for the default SQL query, you must connect all of the source qualifier output ports to the next transformation within the mapplet.
- If the mapplet contains more than one source qualifier, use a Joiner transformation to join the output into one pipeline.
- If the mapplet contains only one source qualifier, you must connect the mapplet output ports to separate pipelines. You cannot use a Joiner transformation to join the output.
- If you need to join the pipelines, you can create two mappings to perform this task.
- Use the mapplet in the first mapping and write data in each pipeline to separate targets.
- Use the targets as sources in the second mapping to join data, then perform any additional transformation necessary.

## Viewing the Mapplet

When you use a mapplet in a mapping, the Designer displays the mapplet object, which contains only the input and output ports of the mapplet. However, you can expand the mapplet by clicking Mappings > Expand from the menu.

When the Designer expands the mapplet, it displays the entire mapping with the mapplet transformations. It does not display the Input and Output transformations. you can view the mapping in this expanded form, but you cannot edit any of the properties, navigate to other folders, or save the repository while the mapplet is expanded.. To continue designing the mapping, click Mappings > Unexpand.



## Rules and Guidelines for Usage of Mapplets

- You can connect an Input transformation to multiple transformations in a mapplet. However, you cannot connect a single port in the Input transformation to multiple transformations in the mapplet.
- An Input transformation must receive data from a single active source.
- A mapplet must contain at least one Input transformation or source definition with at least one port connected to a transformation in the mapplet.
- A mapplet must contain at least one Output transformation with at least one port connected to another transformation in the mapping.
- When a mapplet contains a source qualifier that has an override for the default SQL query, you must connect all of the source qualifier output ports to the next transformation within the mapplet.

- If the mapplet contains more than one source qualifier, use a Joiner transformation to join the output into one pipeline. If the mapplet contains only one source qualifier, you must connect the mapplet output ports to separate pipelines. You cannot use a Joiner transformation to join the output.
- When you change the mapplet type from passive to active, you might invalidate the mapping.
- When you delete ports in the mapplet, you might invalidate the mapping using it.
- When you change the datatype, precision, or scale of a mapplet port, you might invalidate the mapping.
- If you use a Sequence Generator transformation, you must use a reusable Sequence Generator transformation.
- If you use a Stored Procedure transformation, you must configure the Stored Procedure Type to be *Normal*.
- You cannot include the following objects in a mapplet:
  - » Normalizer transformations
  - » Cobol sources
  - » XML Source Qualifier transformations
  - » XML sources
  - » Target definitions
  - » Pre- and post- session stored procedures
  - » Other Mapplets

## Worklets

---

A worklet is an object that represents a set of tasks that you create in the Worklet Designer. Create a worklet when you want to reuse a set of workflow logic in more than one workflow.

To run a worklet, include the worklet in a workflow. The workflow that contains the worklet is called the parent workflow. When the Integration Service runs a worklet, it expands the worklet to run tasks and evaluate links within the worklet. It writes information about worklet execution in the workflow log.

### Creating a Reusable Worklet

You can create reusable worklets in the Worklet Designer.

1. In the Worklet Designer, click Worklet > Create. The Create Worklet dialog box appears.
2. Enter a name for the worklet.
3. Click OK. The Worklet Designer creates a worklet with a Start Task.

You can create a non reusable worklet in the Workflow Designer.

### Configuring Worklet Properties

When you use a worklet in a workflow, you can configure the same set of general task settings on the General tab as any other task. For example, you can make a worklet reusable, disable a worklet, configure the input link to the worklet, or fail the parent workflow based on the worklet.

In addition to general task settings, you can configure the following worklet properties:

- *Worklet variables.* Use worklet variables to reference values and record information. you use worklet variables the same way you use workflow variables. you can assign a workflow variable to a worklet variable to override its initial value.
- *Events.* To use the Event-Wait and Event-Raise tasks in the worklet, you must first declare an event in the worklet properties.
- *Metadata extension.* Extend the metadata stored in the repository by associating information with repository objects.

### Adding Tasks in Worklets

After creating a worklet, add tasks by opening the worklet in the Worklet Designer. A worklet must contain a Start task. The Start task represents the beginning of a worklet. When you create a

worklet, the Worklet Designer creates a Start task. Add tasks in the worklet by using the Tasks toolbar or click Tasks > Create in the Worklet Designer and connect tasks with links.

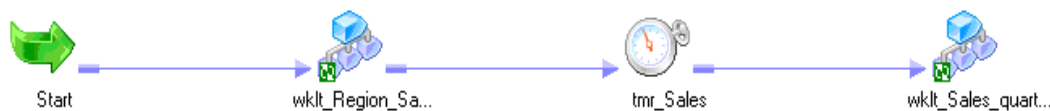
### Nesting Worklets

You can nest a worklet within another worklet. When you run a workflow containing nested worklets, the Integration Service runs the nested worklet from within the parent worklet. you can group several worklets together by function or simplify the design of a complex workflow when you nest worklets.

You might choose to nest worklets to simplify the design of a complex workflow. In the workflow shown below, two worklets relate to regional sales and two worklets relate to quarterly sales.



The workflow shown below has the worklets grouped and nested in parent worklets.



### Creating Nested Worklets

1. From the Worklet Designer, open the parent worklet.
2. To nest an existing reusable worklet, click Tasks > Insert Worklet or Drag and drop the reusable worklet from the repository folder.
3. To create a non-reusable nested worklet, click Tasks > Create, and select worklet.

### Using Worklet Variables

A worklet has the same set of predefined variables as any other task. You can also create user-defined worklet variables which can be persistent or non-persistent. The persistent worklet variable retains its value the next time the Integration Service runs the worklet in the parent workflow. To create a persistent worklet variable, select Persistent when you create the variable.

For example, Use two instances of a reusable worklet with a persistent variable in a workflow to run the worklet twice. Name the first instance of the worklet Worklet1 and the second instance Worklet2. When you run the workflow, the persistent worklet variable retains its value from Worklet1 and becomes the initial value in Worklet2. After the Integration Service runs Worklet2, it retains the value of the persistent variable in the repository and uses the value the next time you run the workflow.

Worklet variables persist for each workflow. A worklet variable does not retain its value when you use instances of the worklet in different workflows.

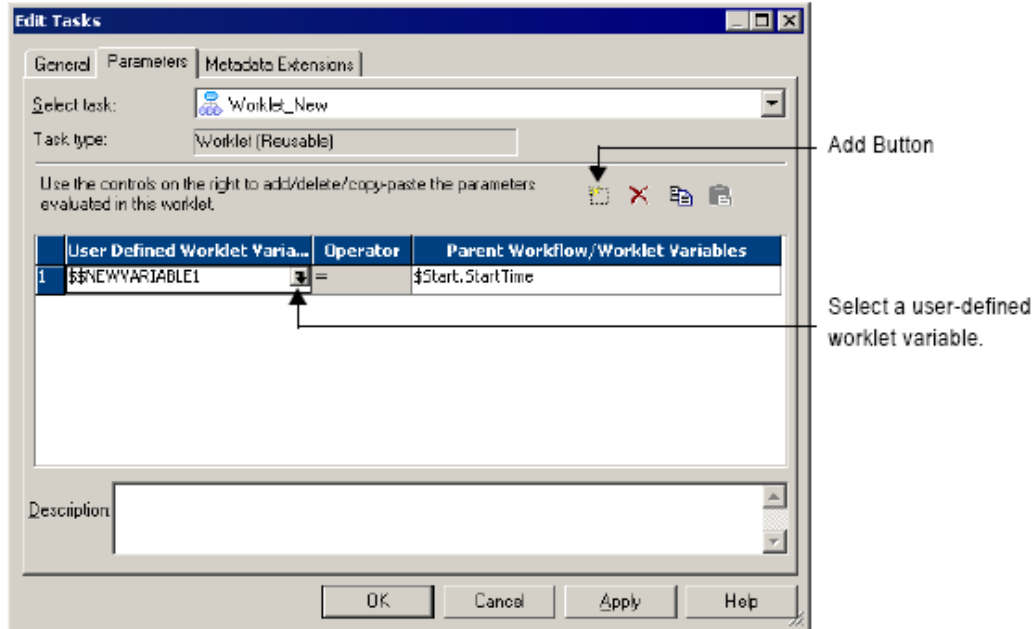
### Overriding the Initial Value of Worklet Variables

For each worklet instance, you can override the initial value of the worklet variable by assigning a workflow variable to it.

*To override the initial value of a worklet variable:*

1. Double-click the worklet instance in the Workflow Designer workspace.
2. On the Parameters tab, click the Add button.
3. Click the open button in the User-Defined Worklet Variables field to select a worklet variable.

4. Click Apply. The worklet variable in this worklet instance now has the selected workflow variable as its initial value.



### Rules and Guidelines

Use the following rules and guidelines when you work with worklet variables:

- You cannot use variables from the parent workflow in the worklet.
- You cannot use user-defined worklet variables in the parent workflow.
- You can use predefined worklet variables in the parent workflow, just as you use predefined variables for other tasks in the workflow.

### Validating Worklets

The Workflow Manager validates worklets when you save the worklet in the Worklet Designer. In addition, when you use worklets in a workflow, the Integration Service validates the workflow according to the following validation rules at run time:

- You cannot run two instances of the same worklet concurrently in the same workflow.
- You cannot run two instances of the same worklet concurrently across two different workflows.
- Each worklet instance in the workflow can run once.

The Workflow Manager displays a red invalid icon if the worklet object is invalid. The Workflow Manager validates the worklet object using the same validation rules used for workflows.

The Workflow Manager displays a blue invalid icon if the worklet instance in the workflow is invalid. When a worklet instance is invalid, the workflow using the worklet instance remains valid.

The worklet instance may be invalid when any of the following conditions occurs:

- The parent workflow or worklet variable you assign to the user-defined worklet variable does not have a matching datatype.
- The user-defined worklet variable you used in the worklet properties does not exist.
- You do not specify the parent workflow or worklet variable you want to assign.

For non-reusable worklets, you may see both red and blue invalid icons displayed over the worklet icon in the Navigator.

## Reusable Workflow Tasks

Workflows can contain reusable task instances and non-reusable tasks. Non-reusable tasks exist within a single workflow. Reusable tasks can be used in multiple workflows in the same folder.

You can create any task as non-reusable or reusable. Tasks you create in the Task Developer are reusable. Tasks you create in the Workflow Designer are non-reusable by default. However, you can edit the general properties of a task to promote it to a reusable task.

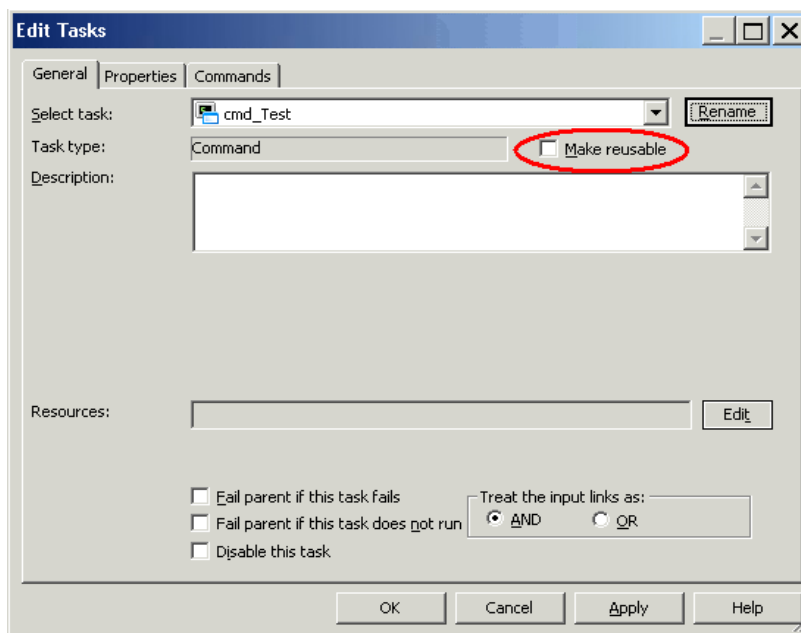
The Workflow Manager stores each reusable task separate from the workflows that use the task. you can view a list of reusable tasks in the Tasks node in the Navigator window. You can see a list of all reusable Session tasks in the Sessions node in the Navigator window.

### Promoting a Non-Reusable Workflow Task

You can promote a non-reusable workflow task to a reusable task. Reusable tasks must have unique names within the repository because when you promote a non-reusable task, the repository service checks for any naming conflicts. If a reusable task with the same name already exists, the repository appends a number to the reusable task name to make it unique. The repository applies the appended name to the checked-out version and to the latest checked-in version of the reusable task.

*To promote a non-reusable workflow task:*

1. In the Workflow Designer, double-click the task you want to make reusable.
2. In the General tab of the Edit Task dialog box, select the Make Reusable option.



3. When prompted whether you are sure you want to promote the task, click Yes.
4. Click OK to return to the workflow.
5. Click Repository > Save.

The newly promoted task appears in the list of reusable tasks in the Tasks node in the Navigator window.

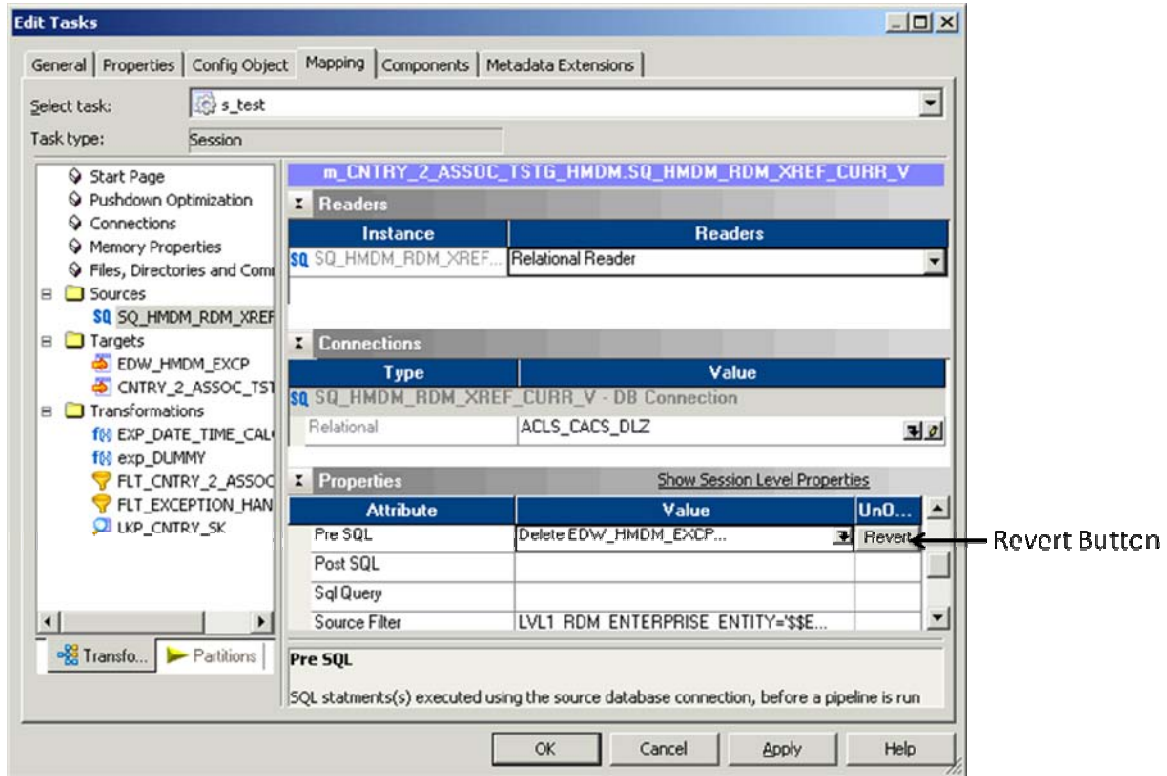
### Instances and Inherited Changes

When you add a reusable task to a workflow, you add an instance of the task. The definition of the task exists outside the workflow, while an instance of the task exists in the workflow. You can edit the task instance in the Workflow Designer. Changes you make in the task instance exist only in the workflow. The task definition remains unchanged in the Task Developer. When you make changes to a reusable task definition in the Task Developer, the changes reflect in the instance of the task in the workflow if you have not edited the instance.



## Reverting Changes in Reusable Tasks Instances

When you edit an instance of a reusable task in the workflow, you can revert back to the settings in the task definition. When you change settings in the task instance, the Revert button appears. The Revert button appears after you override task properties. You cannot use the Revert button for settings that are read-only or locked by another user.



## Try It Out

---

1. Promote an unconnected look up transformation used in a mapping.
2. Create maplet with two sources and include it in a mapping.
3. Create a reusable session, add it in a worklet and override the session properties.
4. Create reusable worklets and add them in workflow so that they are executed serially.

## Summary

---

- ❖ In Informatica Power Center, certain objects can be re-used, if the operating logic or operations is same. Those objects are known as Reusable components
- ❖ Reusable transformations can be used in multiple mappings. An instance of the Reusable Transformations appears within the mapping.
- ❖ A Maplet is a reusable object that you create in the Maplet Designer. It contains a set of transformations and allows you to reuse that transformation logic in multiple mappings.
- ❖ A worklet is an object that represents a set of tasks that you create in the Worklet Designer. Create a worklet when you want to reuse a set of workflow logic in more than one workflow.
- ❖ Reusable tasks can be used in multiple workflows. An instance of the Reusable tasks appears within the mapping. The non-reusable task in workflow can be promoted to reusable task.

## Test your Understanding

---

1. How do you create a reusable transformation?
2. Can you override properties of reusable Transformation instance in a mapping?
3. What are the different types of Maplet Ports?
4. What are worklet variables?



## Session 9: Advanced Topics

### Learning Objectives

- Versioning
- Deployment Groups
- Exporting and Importing Objects
- Copying Objects
- Debugger
- Data Recovery
- Partitioning
- Performance Data Collection
- Bottlenecks

### Versioning

Versioning is a concept where you can configure the repository to store multiple versions of objects. You can configure a repository for versioning when you create it, or you can upgrade an existing repository to support versioned objects. With object versioning, you can store copies of previous versions of objects in development, track changes to those objects, and prepare them for deployment to a production environment.

A versioned repository assigns multiple version numbers to versions of the same object. Each time you check in an object, the repository increments the version number by one and stores a new version of the object in the repository database.

A repository enabled for versioning can store multiple versions of the following objects:

- Sources
- Targets
- Transformations
- Mappings
- Maplets
- Sessions
- Tasks
- Workflows
- Worklets
- Session configurations
- Schedulers
- Cubes
- Dimensions

### Tasks on Versioned Object

You can complete the following tasks when you work with a versioned object in the Repository Manager, Designer, and Workflow Manager.

- **View object version properties.** Each versioned object has a set of version properties and a status. You can also configure the status of a folder to freeze all objects it contains or make them active for editing.
- **Track changes to an object.** You can view a history that includes all versions of a given object, and compare any version of the object in the history to any other version. With the history, you can determine changes made to an object over time.

- **Check out or check in the versioned object.** You can check out an object to reserve it while you edit the object. When you check in an object, the repository saves a new version of the object, and you can add comments to the version. You can also find objects checked out by yourself and other users.
- **View multiple versions of an object in the workspace.** You can view multiple versions of an object in the workspace of the Designer and Workflow Manager.
- **Apply labels to objects.** You can create labels to associate with any versioned object or group of versioned objects in a repository. Use labels to track versioned objects during development, improve query results, and associate groups of objects for deployment or import and export.
- **Group objects for deployment.** You can create groups of versioned objects to deploy to another repository or folder. Use the result set from an object query to group objects for deployment. Or, you can create a static group of objects for deployment.
- **Delete or purge the object version.** You can delete an object from view and continue to store it in the repository. You can recover, or undelete, deleted objects. If you want to permanently remove an object version, you can purge it from the repository.

### Viewing Object History

The history of an object is a record of all of the versions of an object stored in the repository, going back to the initial version, version number one. You can view user changes to an object, the date and time of changes, and comments associated with and labels applied to each version. If you or another user purges a version from the repository, the object history retains a record of the version in the object history and specifies the user who purged the version. You can view object history in the Designer, Repository Manager, and Workflow Manager.

Object Name	Group	Object Type	Version	Time Stamp	Status	Version Comments	Folder Name	User Name	Host Name	Purged by
m_Income_...		Mapping	8	06/23/20...	Active	Check in for depl...	Orders	Administr...	p15389...	
m_Income_...		Mapping	7	06/16/20...	Active	Edit Agg.	Orders	brian	p15389...	
m_Income_...		Mapping	6	06/16/20...	Active	Remove shortcu...	Orders	brian	p15389...	
m_Income_...		Mapping	5	06/16/20...	Active	Remove Filter sh...	Orders	brian	p15389...	
m_Income_...		Mapping	4	06/04/20...	Active	Editing mapping.	Orders	Administr...	p15389...	
m_Income_...		Mapping	2	04/30/20...	Active	Re-enable Distin...	Orders	Administr...	p15389...	Administrator
m_Income_...		Mapping	1	04/30/20...	Active	Disabled Distinct...	Orders	Administr...	p15389...	Administrator

Label Name	User Name	Time Stamp

Ready Total: 7 Selected: 0

Use the following methods to view the object version history in the Designer or Workflow Manager:

- Right-click the object in the Navigator and click Versioning > View History.
- Right-click the object in the workspace and click Versioning > View History.
- Select the object in the workspace and click Versioning > View History from the menu.

### Checking Out Objects

To edit an object, you must check out the object. When you check out an object, the repository obtains a write-intent lock on the object. No other repository users can edit the object when you have it checked out. If you disconnect from the repository and do not save the object, it remains

checked out to you, but you lose the changes you made to it. An object is checked out by default when you create, copy, replace, or import it.

*To check out an object:*

1. Select the object you want to check out. Click Versioning > Check Out.
2. In the Check Out dialog box, enter an optional comment in the comment field.
3. Click OK to check out the object, or Apply to All to apply the checkout comment to multiple objects.

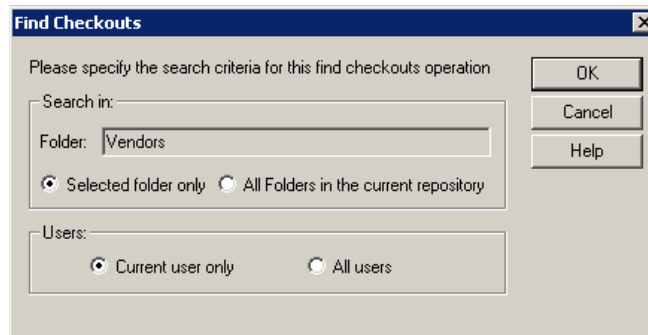
### Viewing Checked-Out Objects

In the Designer, Workflow Manager, or Repository Manager, you can view the list of object the users have checked out. You can use this feature to check in all of the objects you have checked out or to see if an object is available for you to check out. You can narrow or broaden the search for checked-out objects in the following ways:

- **By folder.** Search for checkouts in the selected folder, or search all folders in the repository.
- **By user.** Search for objects you checked out, or search for objects checked out by all users in the repository.

*To view checkouts:*

1. Click Versioning > Find Checkouts.



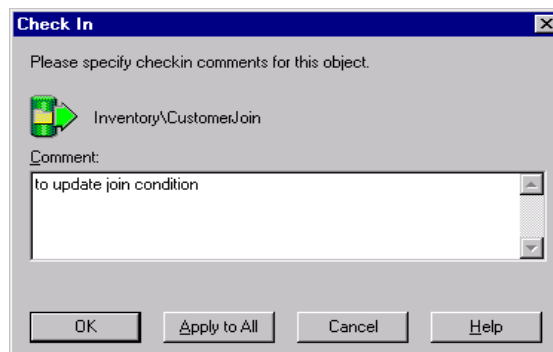
2. Select options to narrow or broaden the search, and click OK.

### Checking In Objects

You must save an object before you can check it in. When you check in an object, the repository creates a new version of the object and assigns it a version number. The repository increments the version number when you check in an object. You must check in an object to purge it. If you save an object without checking it in, the changes are committed to the repository, and the object remains checked out until you check it in. You can check in objects from the Designer, Workflow Manager, or Repository Manager. You can also check in an object from the View History, View Checkouts, View Dependencies, and Query Results windows.

*To check in an object:*

1. Select the object or objects. Click Versioning > Check in.
2. In the Check In dialog box, enter a comment in the comment field.



4. Click OK to check in the object or Apply to All to apply the comment to multiple objects.

When you check in an object, the repository creates a new version of the object and increments the version number by one.

## Deployment Groups

A deployment group is a global object that consists of versioned objects from one or more folders. You use a deployment group to copy the versioned objects to another folder or repository. You can use a deployment group when you want to copy some, but not all, of the objects in a folder. You can also use a deployment group to copy objects from multiple folders.

You can create the following types of deployment groups:

- **Static.** You populate a static deployment group by manually selecting objects. Create a static deployment group if you do not expect the set of deployment objects to change. For example, you might group objects for deployment on a certain date and deploy all objects at once.
- **Dynamic.** You use the result set from an object query to populate the deployment group. Create a dynamic deployment group if you expect the set of deployment objects to change frequently. For example, you can use a dynamic deployment group if you develop multiple objects to deploy on different schedules. You can run the dynamic deployment group query multiple times and add new objects to the group each time you run the query.

## Tasks on Deployment Groups

You can complete the following tasks when you work with deployment groups:

- **Create a deployment group.** Create a global object for deploying objects from one or more folders.
- **Edit a deployment group.** Modify a deployment group. For example, you can convert a static deployment group to a dynamic group, or you can convert a dynamic deployment group to a static group.
- **Configure permissions for a deployment group.** Control read, write, and execute permissions for a deployment group.
- **View the objects in a static or dynamic deployment group.** Preview the objects that the Repository Service will deploy.
- **Add or remove objects in a static deployment group.** Specify the objects that belong to a static deployment group.
- **Associate a query with a dynamic deployment group.** Assign a query to a deployment to dynamically update the objects that the group contains.
- **View the history of a deployment group.** View the history of a deployment group, including the source and target repositories, deployment date, and user who ran the deployment.
- **Roll back a deployment group.** Roll back a deployment group to purge deployed versions of objects from the target repository.

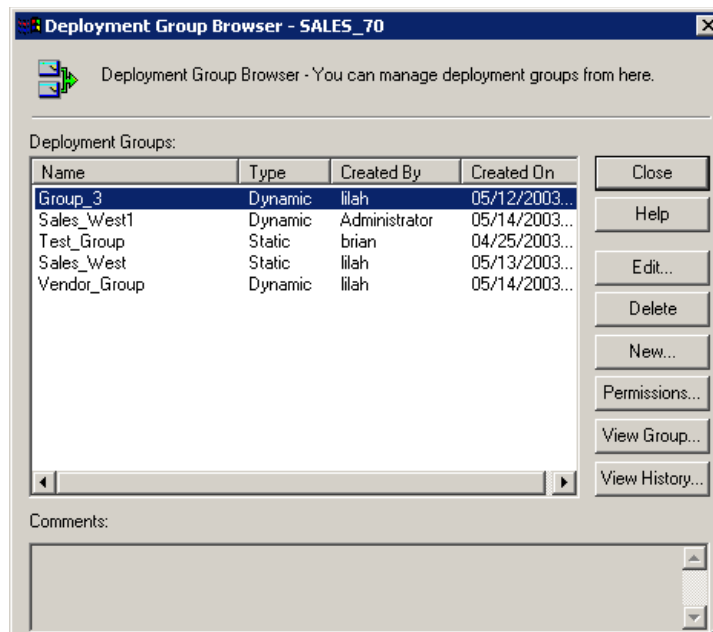
## Creating a Deployment Group

You must have the Manage Deployment Group privilege to create a deployment group. You use the Deployment Group Editor to create and edit deployment groups.

To create a deployment group:

1. In the Repository Manager, click Versioning > Deployment > Groups to view the existing deployment groups in the Deployment Group Browser.
2. Click New to configure the deployment group in the Deployment Group Editor.
3. Enter a name for the deployment group.
4. Select whether to create a static or dynamic deployment group.
5. If you are creating a dynamic deployment group, click Queries to select a query from the Query Browser, and then click Close to return to the Deployment Group Editor.

6. Optionally, enter a comment for the deployment group. Click OK.



## Exporting and Importing Objects

In the PowerCenter Client, you can export repository objects to an XML file and then import repository objects from the XML file. Use the following client applications to export and import repository objects:

- **Repository Manager.** You can export and import both Designer and Workflow Manager Objects.
- **Designer.** You can export and import Designer objects.
- **Workflow Manager.** You can export and import Workflow Manager objects.
- **PMREP.** You can export and import both Designer and Workflow Manager Objects. You might use pmrep to automate exporting objects on a daily or weekly basis.

Exporting and importing an object is similar to copying an object from one folder or repository to another. For example, when you copy an object between folders or export and import that object, you can resolve object name conflicts. However, when you copy objects between folders or repositories, you must be connected to both repositories. When you export an object from one repository and import the object into another repository, you do not need to be connected to both repositories.

You can export and import repository objects to accomplish the following tasks:

- **Deploy metadata into production.** After you test a mapping in a development repository, you can export it to an XML file and then import it from the XML file into a production repository. You might export and import objects to incrementally deploy metadata by exporting and importing part of a composite object.
- **Archive metadata.** You can export objects to an XML file that you no longer need before removing them from the repository.
- **Share metadata.** You can share metadata with a third party. For example, you want to send a mapping to someone else for testing or analysis, but you do not want to disclose repository connection information for security reasons. You can export the mapping to an XML file and edit the repository connection information before sending the XML file. The third party can import the mapping from the XML file and analyze the metadata.
- **Search and replace property names in an entire repository object.** You can search for a property name and replace all occurrences of it with a different name. For example, you

have a mapping with an unconnected Lookup transformation. You want to change the name of a port in the unconnected Lookup transformation. Several other transformations call the lookup port through an expression, so you want to make sure you change the port name in all other expressions. You can export the mapping to an XML file and open it in a text editor. Search for the old port name and replace all references to it with the new port name. Then import the mapping into the repository.

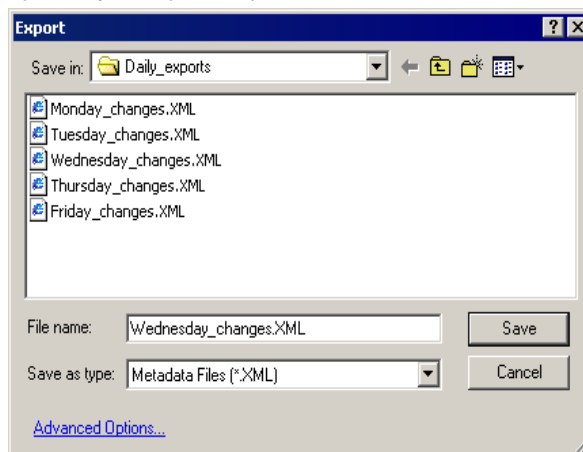
- **Copy metadata between repositories.** You can copy objects between repositories that you cannot connect to from the same client. Export the object and transfer the XML file to the target machine. Then import the object from the XML file into the target repository.
- **Create mappings.** You can export an existing mapping and use Data Stencil for Visio to turn the mapping into a mapping template. Once a mapping template is created in Data Stencil you can import multiple mappings into the repository.

## Export Objects

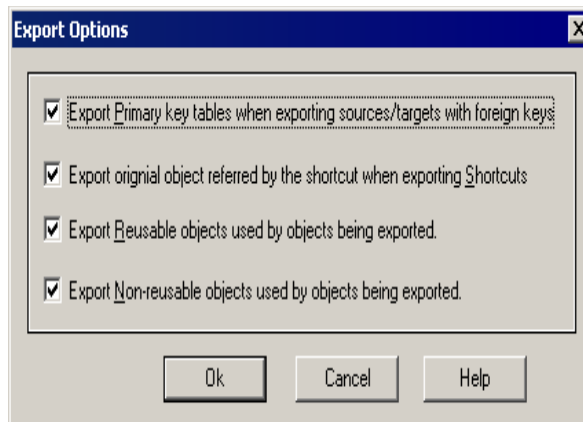
You can export objects from the repository using the Designer, Workflow Manager, Repository Manager, query result, or object history. To export objects from the query result or object history, select the objects to export and choose Tools-Export to XML File.

*To export an object from the Designer, Workflow Manager, or Repository Manager:*

1. Open the folder that contains the objects you want to export.
2. In the Navigator or workspace, select the objects to export.
3. Click Repository > Export Objects.

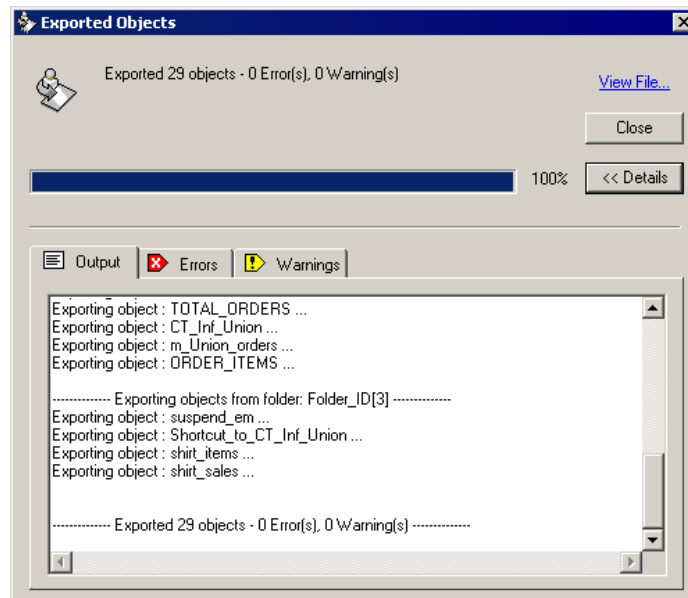


4. To choose which dependent objects to export, click Advanced Options.



5. In the Export Options dialog box, choose which dependent objects to export and click OK.
6. In the Export dialog box, navigate to the directory where you want to save the XML file. Enter a name for the XML file and click Save.

The PowerCenter Client exports the objects to an XML file, and displays export status in the Exported Objects dialog box:



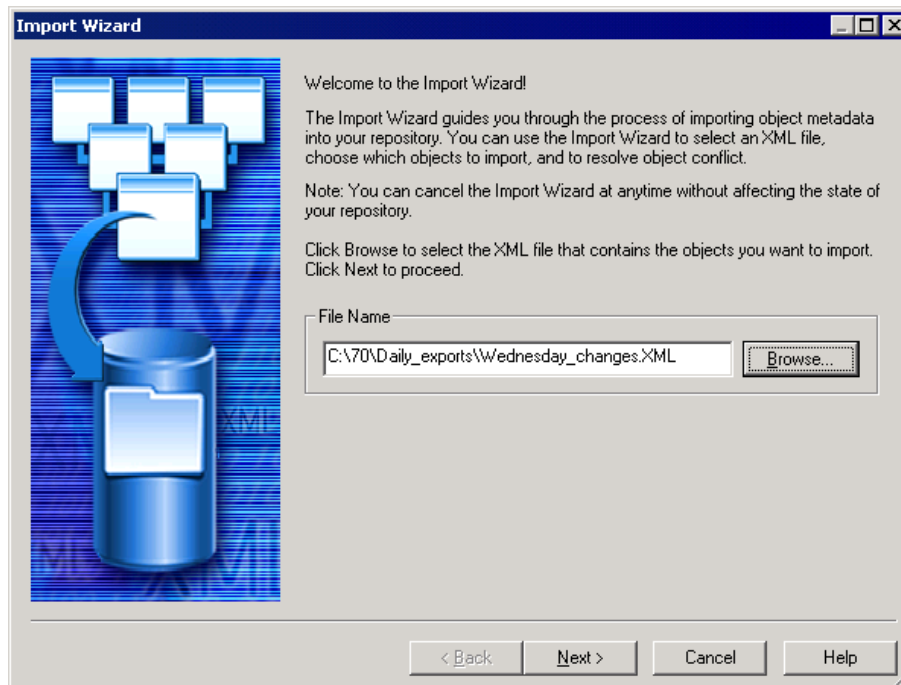
## Import Objects

You can import objects into the repository using the Designer, Workflow Manager, or Repository Manager. You can compare objects when importing objects with the Import Wizard.

*To import an object:*

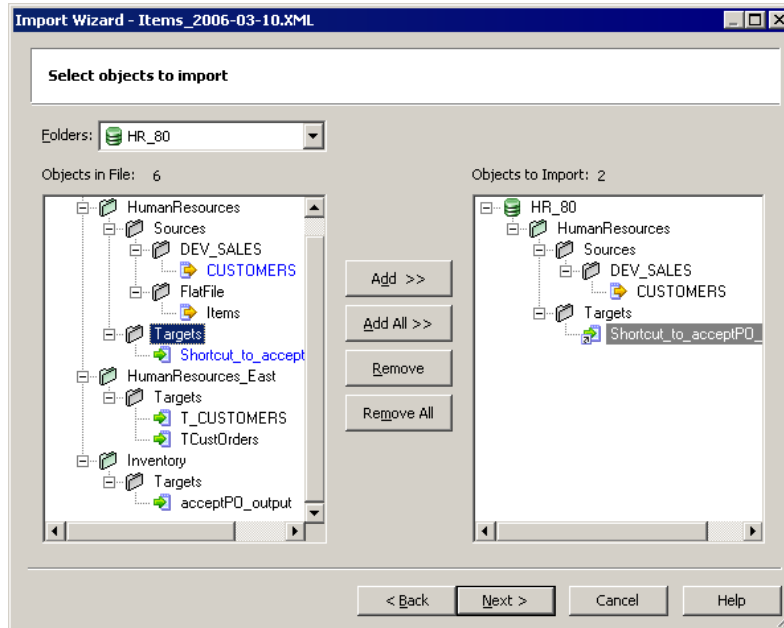
1. Open the folder into which you want to import an object.
2. Click Repository > Import Objects.

The Import Wizard opens to guide you through the process of importing the objects into the target folder.

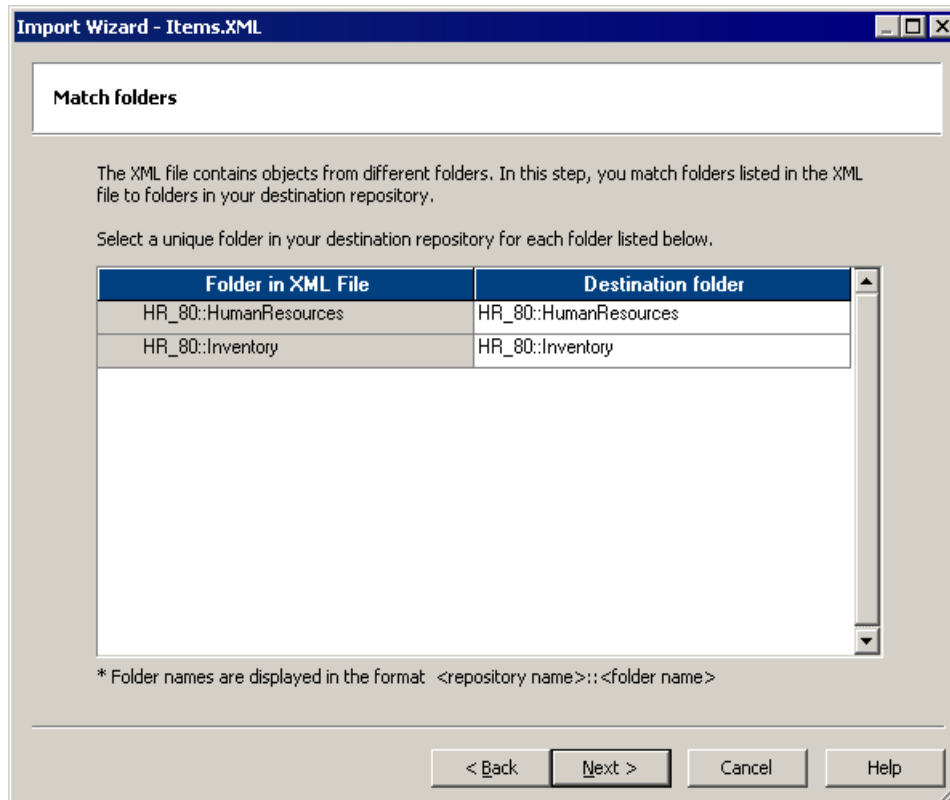




3. In the Import Wizard, click Browse to locate the XML file. Navigate to the directory where the XML file is located. Select the XML file and click OK. Click Next.
5. Select the objects to import and click Add. Click Next.

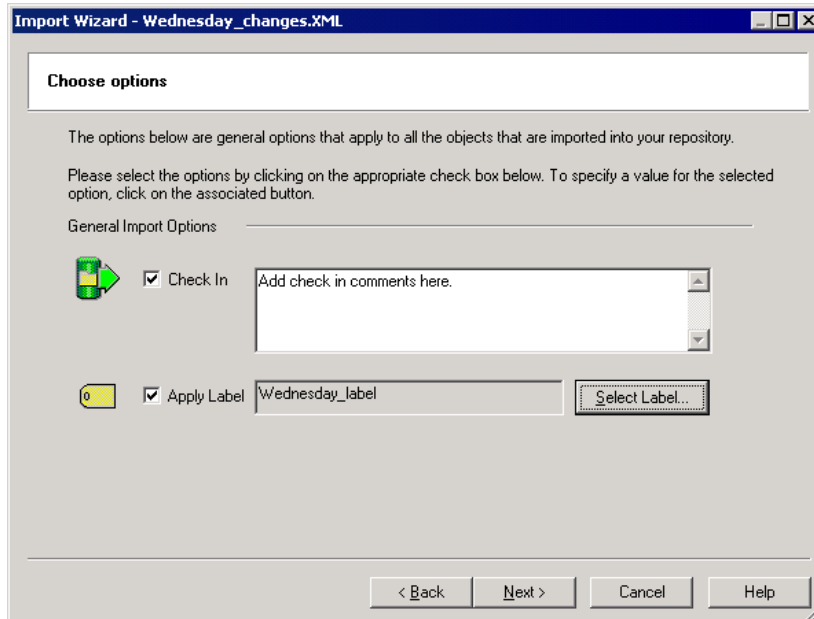


6. Click the Open button for a folder listed in the Import Wizard. The Folder Selection dialog box appears.
7. Select a folder in the destination repository and click OK. You must select a different folder for each folder listed in the Import Wizard. Click Next.

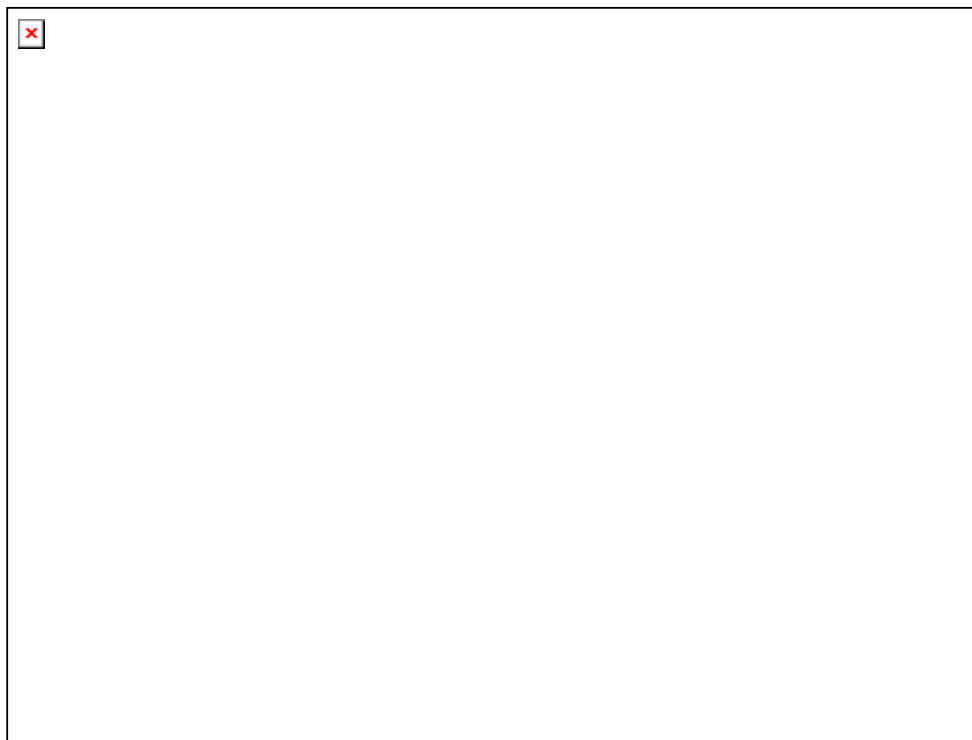




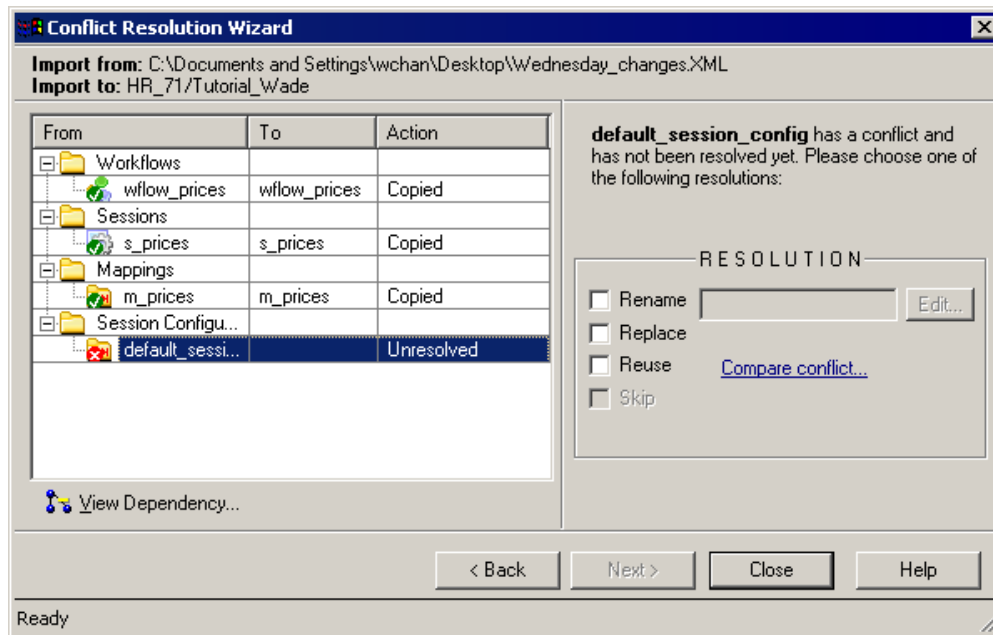
8. To check in all objects after importing them, select Check In and enter comments in the comment field.



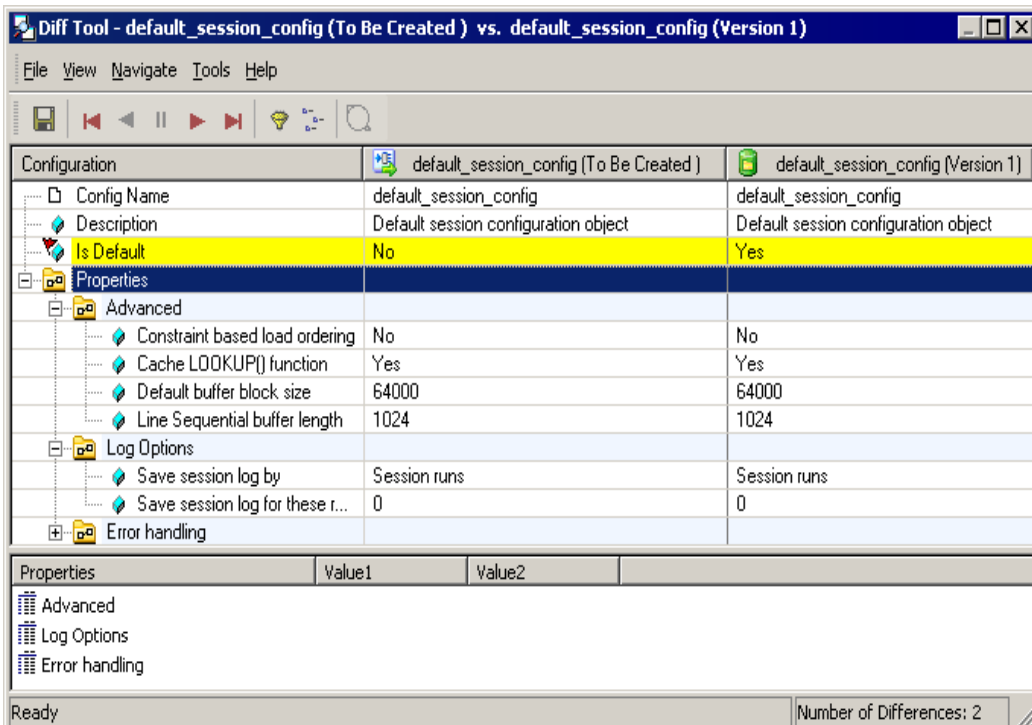
9. To apply a label to all objects you import, select Apply Label and click Select Label. In the Label Browser dialog box, choose the label and click OK.  
You can only apply a label to the objects if you choose to check them in.
10. Click Next.
11. To create a new rule, click New Rule. Choose to which objects to apply the rule and select a resolution. Click Next.



The Import Wizard opens the Conflict Resolution Wizard for objects in one of the folders listed in the XML file.



12. Click Compare Conflict to compare conflicting objects in the XML file and target repository. The Diff Tool window appears.



13. You can save the comparison as a text or HTML file.

If the objects in the XML file exist in the target repository, the Targets window appears instead of the Diff Tool window.

The Targets window displays the following information:

Field	Value
Table Name:	EMP_DETAILS
Repository:	Sample
Folder:	Tutorial_connie
Version:	(To Be Created)
Shortcut Properties:	

Buttons: Close, Help

14. Resolve object conflicts as they appear in the Conflict Resolution Wizard. Click Next to proceed through the Conflict Resolution Wizard.
15. Click Close when you resolve all the conflicts for this folder.

The Import Wizard opens the Conflict Resolution Wizard for objects in any other folder listed in the XML file. On resolving the conflicts for all objects, the Import Wizard proceeds with the import process. You can click View/Edit to view or edit the object conflicts for the objects in that folder.

**Resolve conflicts in folder**

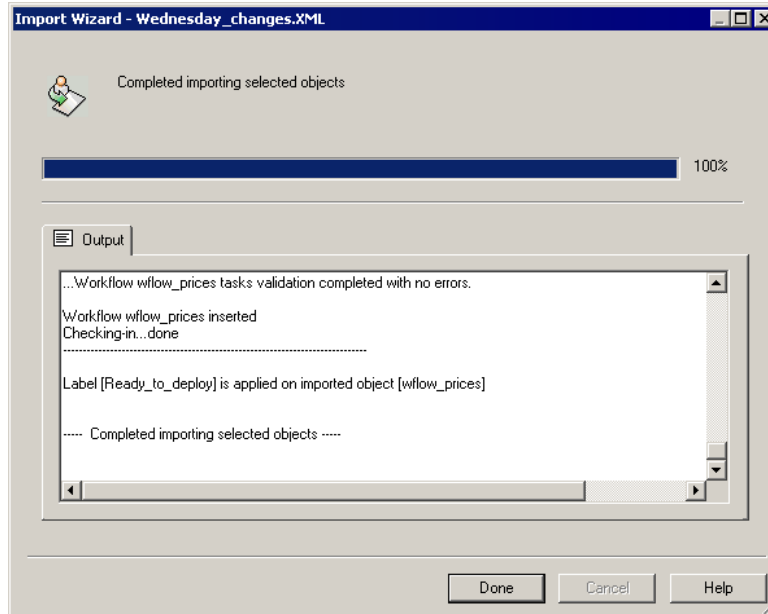
In this step, you can resolve object conflicts that were not resolved by the rules you created in the previous step. To resolve object conflicts in a folder, click the Resolve button for each folder listed.

Folder in XML File	Destination Folder	Status	Action
NORTHWEST::Orders	NORTHWEST::Orders_dev	- Resolved -	View/Edit...
NORTHWEST::HR	NORTHWEST::Orders_dlv	- Resolved -	View/Edit...

\* Folder names are displayed in the format <repository name>::<folder name>

Buttons: < Back, Import >, Cancel, Help

15. Click Import in the Import Wizard to import the objects into the repository. The PowerCenter Client imports the objects into the destination repository, and displays the progress of the import process. The Output window displays the results of the import process. Errors and warnings are designated by colored text.



16. Click Done.

## Copying Objects

The Workflow Manager, Designer, and Repository Manager provide a Copy Wizard that you use to copy repository objects. You can copy repository objects such as workflows, worklets, tasks, sessions, mappings, mapplets, sources, targets, and transformations. You can also copy segments of workflows or mappings. You can copy objects within the same folder, to a different folder, or to a different repository. If you want to copy an object to another folder, you must first open the target folder.

### Copy Wizard

The Copy Wizard checks for conflicts in the target folder and provides choices to resolve the conflicts. For example, if an item exists in the target folder, a description of the conflict appears in the Conflict Message section of the screen. The Copy Wizard displays possible resolutions in the Resolution area of the screen. For a duplicate object you can rename, reuse, replace, or skip copying the object.

*To copy an object using the Copy Wizard:*

1. Open the target folder.
2. In the Navigator, select the object you want to copy.
3. Drag or copy the object into the target folder.
4. Click Yes in the Copy Confirmation dialog box.

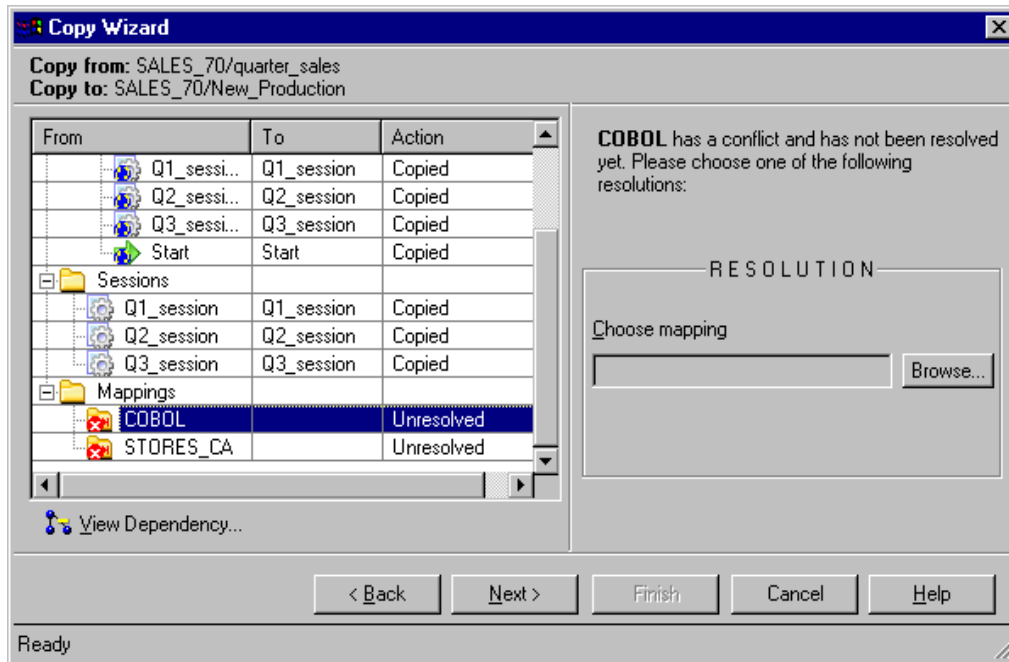
The Copy Wizard appears. The Copy Wizard displays objects by type. For example, the sessions display under the Sessions node, and mappings display under the Mappings node.

The Copy Wizard displays a red icon on objects with conflicts. It displays conflicts one object type at a time.

5. Click Next if you do not encounter a conflict.

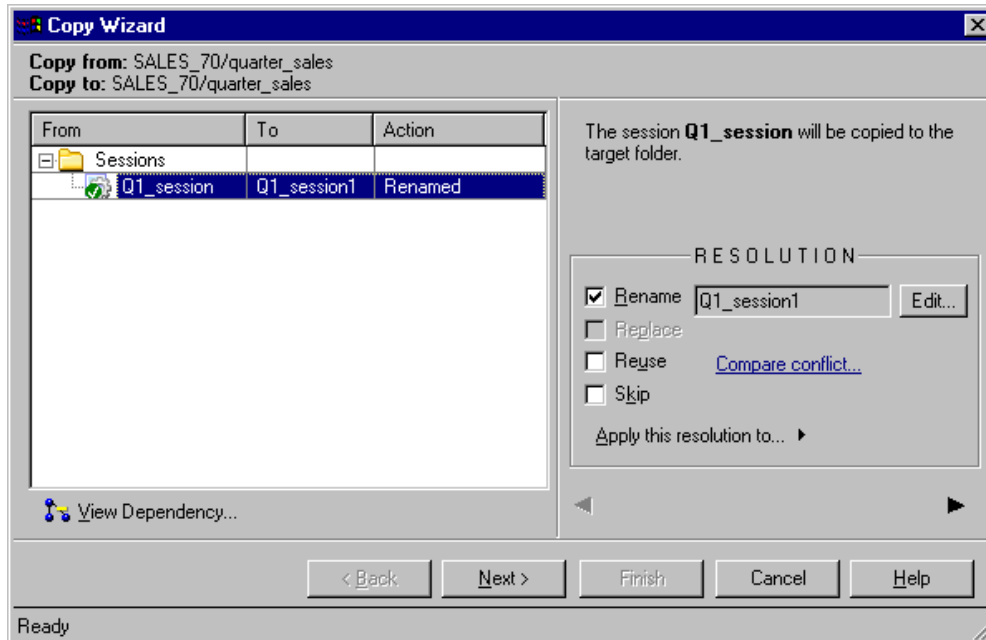
If you encounter a conflict, choose a resolution from the Resolution options. For example, the following Figure shows the first of two unresolved mapping conflicts to resolve. The

resolution option requires you to browse for an appropriate mapping. Click Browse to select a mapping.



6. Click Next to view the next conflict.

If you work with session or workflow conflicts, you can click Next Conflict/Option to configure additional options for sessions or workflows with conflicts. For example, use Next Conflict/Option if you want to apply default connections in the target during the copy. Next Conflict/Option appears when you have session and workflow conflicts, and you choose to Rename or Replace the target objects.



7. Repeat steps 5 to 6 until you resolve all conflicts. The Copy Summary information appears.
8. Click Finish to complete the copy process.

## Debugger

You can debug a valid mapping to gain troubleshooting information about data and error conditions. To debug a mapping, you configure and run the Debugger from within the Mapping Designer. The Debugger uses a session to run the mapping on the Integration Service. When you run the Debugger, it pauses at breakpoints and you can view and edit transformation output data.

You might want to run the Debugger in the following situations:

- **Before you run a session.** After you save a mapping, you can run some initial tests with a debug session before you create and configure a session in the Workflow Manager.
- **After you run a session.** If a session fails or if you receive unexpected results in the target, you can run the Debugger against the session. You might also want to run the Debugger against a session if you want to debug the mapping using the configured session properties.

### Debugger Session Types

You can select three different debugger session types when you configure the Debugger. The Debugger runs a workflow for each session type. You can choose from the following Debugger session types when you configure the Debugger:

- **Use an existing non-reusable session.** The Debugger uses existing source, target, and session configuration properties. When you run the Debugger, the Integration Service runs the non-reusable session and the existing workflow. The Debugger does not suspend on error.
- **Use an existing reusable session.** The Debugger uses existing source, target, and session configuration properties. When you run the Debugger, the Integration Service runs a debug instance of the reusable session and creates and runs a debug workflow for the session.
- **Create a debug session instance.** You can configure source, target, and session configuration properties through the Debugger Wizard. When you run the Debugger, the Integration Service runs a debug instance of the debug workflow and creates and runs a debug workflow for the session.

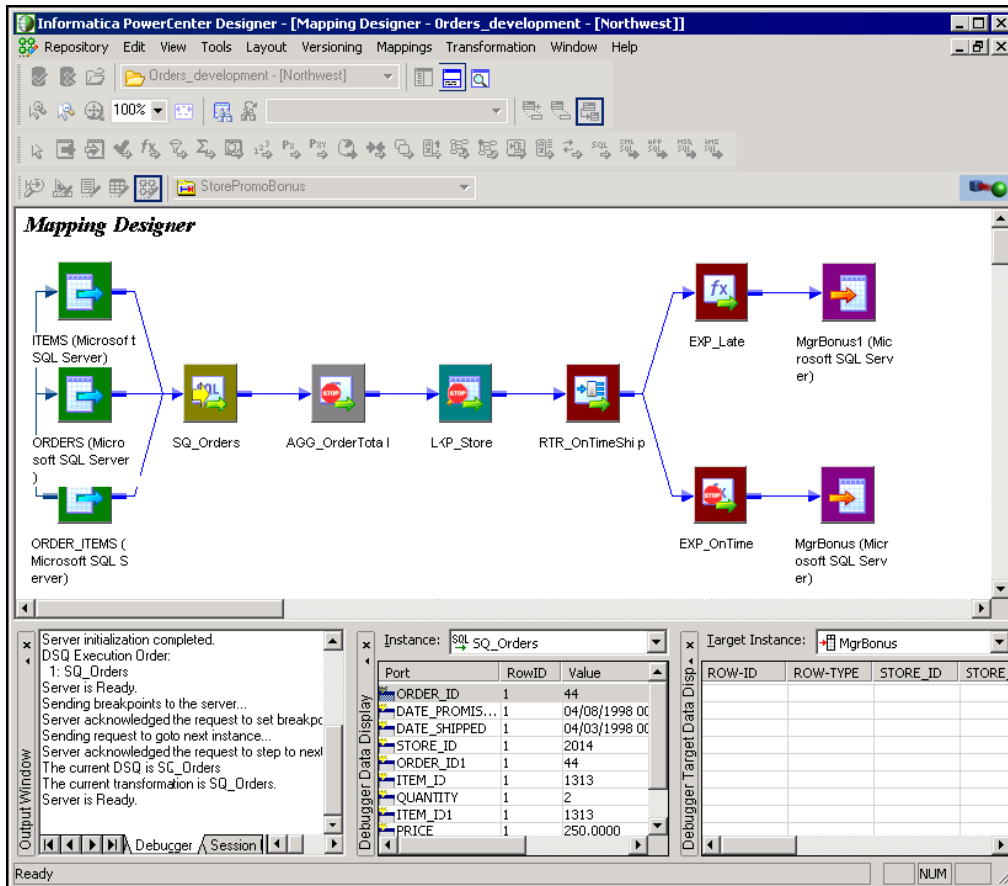
### Debugging Process

Complete the following process to debug a mapping:

- **Create breakpoints.** Create breakpoints in a mapping where you want the Integration Service to evaluate data and error conditions.
- **Configure the Debugger.** Use the Debugger Wizard to configure the Debugger for the mapping. Select the session type the Integration Service uses when it runs the Debugger. When you create a debug session, you configure a subset of session properties within the Debugger Wizard, such as source and target location. You can also choose to load or discard target data.
- **Run the Debugger.** Run the Debugger from within the Mapping Designer. When you run the Debugger, the Designer connects to the Integration Service. The Integration Service initializes the Debugger and runs the debugging session and workflow. The Integration Service reads the breakpoints and pauses the Debugger when the breakpoints evaluate to true.
- **Monitor the Debugger.** While you run the Debugger, you can monitor the target data, transformation and maplet output data, the debug log, and the session log. When you run the Debugger, the Designer displays the following windows:
  - » **Debug log.** View messages from the Debugger.
  - » **Target window.** View target data.
  - » **Instance window.** View transformation data.
- **Modify data and breakpoints.** When the Debugger pauses, you can modify data and see the effect on transformations, maplets, and targets as the data moves through the pipeline. You can also modify breakpoint information.

The Designer saves mapping breakpoint and Debugger information in the workspace files. You can copy breakpoint information and the Debugger configuration to another mapping. If you want to run the Debugger from another PowerCenter Client machine, you can copy the breakpoint information and the Debugger configuration to the other PowerCenter Client machine.

### Debugger Window



### Data Recovery

If an error causes a session to stop unexpectedly, the usual procedure would be refer to the session logs to determine the cause of the failure. Correct the errors, and then complete the session. In these cases, the data recovery depends on the configuration of the mapping and the session, the specific failure, and how much progress the session made before it failed. When the PowerCenter Server runs in recovery mode, it continues to commit data from the point of the last successful commit. This allows you to restart a failed session and complete it as if the session had run without pause.

Depending on the configuration and status of the workflow and session, you can choose one or more of the following recovery methods:

- Recover a suspended workflow
- Recover a failed workflow
- Recover a session task

*To Perform a Data Recovery:*

1. Configure the mapping
  - The input data should be sorted.
  - Verify all targets receive data from transformations that produce repeatable data.



2. Configure the session
  - The session should be enabled for recovery in settings.
  - The previous session run failed and the recovery information is accessible.
3. Configure the Workflow
  - The workflow needs to be configured as Suspend on Error, so that the errors can be corrected in between.
4. Configure the target database
  - Two recovery tables PM\_RECOVERY and PM\_TGT\_RUN\_ID needs to be created in the target database.

### Limitations of Data Recovery

Data recovery cannot be successful under the following circumstances

- If the number of partitions changed after the initial session failure
- If the recovery table is empty or removed from target databases
- Recovery cache file is empty
- The server should be the same Operating system
- The source data is not sorted
- The source or target data changed after the initial session failure
- The mapping uses a Normalizer or Sequence Generator transformation
- The data movement mode changes after the initial session failure
- The server code page changes after the initial session failure
- The session writes to a relational target in bulk mode, but the session is not configured to truncate the target table

### Partitioning

The Integration Services creates a default partition type at each partition point. If you have the Partitioning option, you can change the partition type. The partition type controls how the Integration Service distributes data among partitions at partition points.

When you configure the partitioning information for a pipeline, you must define a partition type at each partition point in the pipeline. The partition type determines how the Integration Service redistributes data across partition points.

You can define the following partition types in the Workflow Manager:

- **Database partitioning.** The Integration Service queries the IBM DB2 or Oracle system for table partition information. It reads partitioned data from the corresponding nodes in the database. Use database partitioning with Oracle or IBM DB2 source instances on a multi-node table space. Use database partitioning with DB2 targets
- **Hash partitioning.** Use hash partitioning when you want the Integration Service to distribute rows to the partitions by group. For example, you need to sort items by item ID, but you do not know how many items have a particular ID number.

You can use two types of hash partitioning:

- » **Hash auto-keys.** The Integration Service uses all grouped or sorted ports as a compound partition key. You may need to use hash auto-keys partitioning at Rank, Sorter, and unsorted Aggregator transformations.
- » **Hash user keys.** The Integration Service uses a hash function to group rows of data among partitions. You define the number of ports to generate the partition key.
- **Key range.** You specify one or more ports to form a compound partition key. The Integration Service passes data to each partition depending on the ranges you specify for each port. Use key range partitioning where the sources or targets in the pipeline are partitioned by key range.
- **Pass-through.** The Integration Service passes all rows at one partition point to the next partition point without redistributing them. Choose pass-through partitioning where you

want to create an additional pipeline stage to improve performance, but do not want to change the distribution of data across partitions.

- **Round-robin.** The Integration Service distributes data evenly among all partitions. Use round-robin partitioning where you want each partition to process approximately the same number of rows.

The partitioning information for a pipeline controls the following factors:

- The number of reader, transformation, and writer threads that the master thread creates for the pipeline.
- How the PowerCenter Server reads data from the source, including the number of connections to the source.
- How the PowerCenter Server distributes rows of data to each transformation as it processes the pipeline.
- How the PowerCenter Server writes data to the target, including the number of connections to each target in the pipeline.

## Performance Data Collection

---

You can configure a session to collect performance details and store them in the PowerCenter repository. You must also configure the Integration Service that runs the session to store the run-time information at the verbose level. The Integration Service stores run-time information in the PowerCenter repository. The Workflow Monitor displays performance details for each session that is configured to show performance details.

### Configuring Performance Details

*To configure performance details:*

1. In the Workflow Manager, open the session properties.
2. On the Properties tab, select Collect Performance Data to view performance details while the session runs.
3. Select Write Performance Data to Repository to view performance details for previous session runs.
4. Click OK.
5. Save the changes to the repository.

## Bottlenecks

---

Performance bottlenecks can be identified by the first step of Performance Tuning, to optimize session performance. To tune the performance of a session, first you identify a performance bottleneck, eliminate it, and then identify the next performance bottleneck until you are satisfied with the session performance. Performance bottlenecks can occur in the source and target databases, the mapping, the session, and the system. You can use the test load option to run sessions when you tune session performance.

### Eliminating Bottlenecks

Use the following methods to identify performance bottlenecks:

- **Run test sessions.** You can configure a test session to read from a flat file source or to write to a flat file target to identify source and target bottlenecks.
- **Study performance details and thread statistics.** You can analyze performance details to identify session bottlenecks. Performance details provide information such as read-from-disk and write-to-disk counters.
- **Monitor system performance.** You can use system monitoring tools to view the percentage of CPU usage, I/O waits, and paging to identify system bottlenecks.

Once you determine the location of a performance bottleneck, use the following guidelines to eliminate the bottleneck:

- **Eliminate source and target database bottlenecks.** Have the database administrator optimize database performance by optimizing the query, increasing the database network packet size, or configuring index and key constraints.
- **Eliminate mapping bottlenecks.** Fine tune the pipeline logic and transformation settings and options in mappings to eliminate mapping bottlenecks.
- **Eliminate session bottlenecks.** Optimize the session strategy and use performance details to help tune session configuration.
- **Eliminate system bottlenecks.** Have the system administrator analyze information from system monitoring tools and improve CPU and network performance.

## Summary

---

- ❖ Versioning is a concept where you can configure the repository to store multiple versions of objects
- ❖ In the PowerCenter Client, you can copy repository objects & import/export repository objects from/to XML files
- ❖ A deployment group is a global object that consists of versioned objects from one or more folders
- ❖ Debugger is an in-built tool for debugging a valid mapping to gain troubleshooting information about data and error conditions
- ❖ To tune performance Check 'Collect Performance Data' option in the session properties of a session in Server Manager

## Test your Understanding

---

1. \_\_\_\_\_ is used to store copies of previous versions of objects.
2. To edit an object, you must \_\_\_\_\_ the object.
3. In the mapping designer, troubleshooting information about data and error conditions for a valid mapping can be gained from using \_\_\_\_\_.
4. Choose the correct option: To copy an object to another folder, you must open the a) Source Folder b) Target Folder
5. What are the two types of Deployment Groups?

## References

### Websites

---

<http://www.informatica.com>

### Books

---

PDF manuals provided with Informatica Software Package

## STUDENT NOTES:

## STUDENT NOTES: