



Informatica® PowerCenter
10.2

Getting Started

© Copyright Informatica LLC 1998, 2020

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

Informatica, the Informatica logo, and PowerCenter are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties, including without limitation: Copyright DataDirect Technologies. All rights reserved. Copyright © Sun Microsystems. All rights reserved. Copyright © RSA Security Inc. All Rights Reserved. Copyright © Ordinal Technology Corp. All rights reserved. Copyright © Aandacht c.v. All rights reserved. Copyright Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright © Meta Integration Technology, Inc. All rights reserved. Copyright © Intalio. All rights reserved. Copyright © Oracle. All rights reserved. Copyright © Adobe Systems Incorporated. All rights reserved. Copyright © DataArt, Inc. All rights reserved. Copyright © ComponentSource. All rights reserved. Copyright © Microsoft Corporation. All rights reserved. Copyright © Rogue Wave Software, Inc. All rights reserved. Copyright © Teradata Corporation. All rights reserved. Copyright © Yahoo! Inc. All rights reserved. Copyright © Glyph & Cog, LLC. All rights reserved. Copyright © Thinkmap, Inc. All rights reserved. Copyright © Clearpace Software Limited. All rights reserved. Copyright © Information Builders, Inc. All rights reserved. Copyright © OSS Nokalva, Inc. All rights reserved. Copyright Edifecs, Inc. All rights reserved. Copyright Cleo Communications, Inc. All rights reserved. Copyright © International Organization for Standardization 1986. All rights reserved. Copyright © ej-technologies GmbH. All rights reserved. Copyright © Jaspersoft Corporation. All rights reserved. Copyright © International Business Machines Corporation. All rights reserved. Copyright © yWorks GmbH. All rights reserved. Copyright © Lucent Technologies. All rights reserved. Copyright © University of Toronto. All rights reserved. Copyright © Daniel Veillard. All rights reserved. Copyright © Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright © MicroQuill Software Publishing, Inc. All rights reserved. Copyright © PassMark Software Pty Ltd. All rights reserved. Copyright © LogiXML, Inc. All rights reserved. Copyright © 2003-2010 Lorenzi Davide, All rights reserved. Copyright © Red Hat, Inc. All rights reserved. Copyright © The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright © EMC Corporation. All rights reserved. Copyright © Flexera Software. All rights reserved. Copyright © Jinfonet Software. All rights reserved. Copyright © Apple Inc. All rights reserved. Copyright © Telerik Inc. All rights reserved. Copyright © BEA Systems. All rights reserved. Copyright © PDFlib GmbH. All rights reserved. Copyright © Orientation in Objects GmbH. All rights reserved. Copyright © Tanuki Software, Ltd. All rights reserved. Copyright © Ricebridge. All rights reserved. Copyright © Sencha, Inc. All rights reserved. Copyright © Scalable Systems, Inc. All rights reserved. Copyright © jqWidgets. All rights reserved. Copyright © Tableau Software, Inc. All rights reserved. Copyright © MaxMind, Inc. All Rights Reserved. Copyright © TMate Software s.r.o. All rights reserved. Copyright © MapR Technologies Inc. All rights reserved. Copyright © Amazon Corporate LLC. All rights reserved. Copyright © Highsoft. All rights reserved. Copyright © Python Software Foundation. All rights reserved. Copyright © BeOpen.com. All rights reserved. Copyright © CNRI. All rights reserved.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>), and/or other software which is licensed under various versions of the Apache License (the "License"). You may obtain a copy of these Licenses at <http://www.apache.org/licenses/>. Unless required by applicable law or agreed to in writing, software distributed under these Licenses is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the Licenses for the specific language governing permissions and limitations under the Licenses.

This product includes software which was developed by Mozilla (<http://www.mozilla.org/>), software copyright The JBoss Group, LLC, all rights reserved; software copyright © 1999-2006 by Bruno Lowagie and Paulo Soares and other software which is licensed under various versions of the GNU Lesser General Public License Agreement, which may be found at <http://www.gnu.org/licenses/lgpl.html>. The materials are provided free of charge by Informatica, "as-is", without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

The product includes ACE(TM) and TAO(TM) software copyrighted by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine, and Vanderbilt University, Copyright (©) 1993-2006, all rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (copyright The OpenSSL Project. All Rights Reserved) and redistribution of this software is subject to terms available at <http://www.openssl.org> and <http://www.openssl.org/source/license.html>.

This product includes Curl software which is Copyright 1996-2013, Daniel Stenberg, <daniel@haxx.se>. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://curl.haxx.se/docs/copyright.html>. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

The product includes software copyright 2001-2005 (©) MetaStuff, Ltd. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://www.dom4j.org/license.html>.

This product includes software copyright © 1996-2006 Per Bothner. All rights reserved. Your right to use such materials is set forth in the license which may be found at <http://www.gnu.org/software/kawa/Software-License.html>.

This product includes OSSP UUID software which is Copyright © 2002 Ralf S. Engelschall, Copyright © 2002 The OSSP Project Copyright © 2002 Cable & Wireless Deutschland. Permissions and limitations regarding this software are subject to terms available at <http://www.opensource.org/licenses/mit-license.php>.

This product includes software developed by Boost (<http://www.boost.org/>) or under the Boost software license. Permissions and limitations regarding this software are subject to terms available at http://www.boost.org/LICENSE_1_0.txt.

This product includes software copyright © 1997-2007 University of Cambridge. Permissions and limitations regarding this software are subject to terms available at <http://www.pcre.org/license.txt>.

This product includes software copyright © 2007 The Eclipse Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://www.eclipse.org/org/documents/epl-v10.php> and at <http://www.eclipse.org/org/documents/edl-v10.php>.

This product includes software licensed under the terms at <http://www.tcl.tk/software/tcltk/license.html>, <http://www.bosrup.com/web/overlib?License>, <http://www.stlport.org/doc/license.html>, <http://asm.ow2.org/license.html>, <http://www.cryptix.org/LICENSE.TXT>, <http://hsqldb.org/web/hsqldbLicense.html>, <http://httpunit.sourceforge.net/doc/license.html>, <http://jung.sourceforge.net/license.txt>, http://www.gzip.org/zlib/zlib_license.html, <http://www.openldap.org/software/release/license.html>, <http://www.libssh2.org>, <http://slf4j.org/license.html>, <http://www.sente.ch/software/OpenSourceLicense.html>, <http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3-license-agreement>, <http://antlr.org/license.html>, <http://aopalliance.sourceforge.net/>, <http://www.bouncycastle.org/license.html>, <http://www.jgraph.com/jgraphdownload.html>, <http://www.jcraft.com/jsch/LICENSE.txt>, http://jotm.objectweb.org/bsd_license.html, <http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>, <http://www.slf4j.org/license.html>, <http://nanoxml.sourceforge.net/orig/copyright.html>, <http://www.json.org/license.html>, <http://forge.ow2.org/projects/javaservice/>, <http://www.postgresql.org/about/license.html>, <http://www.sqlite.org/copyright.html>, <http://www.tcl.tk/software/tcltk/license.html>, <http://www.jaxen.org/faq.html>, <http://www.jdom.org/docs/faq.html>, <http://www.slf4j.org/license.html>, <http://www.iodbc.org/dataspace/iodbc/wiki/iODBC/License>, <http://www.keplerproject.org/md5/license.html>, <http://www.toedter.com/en/jcalendar/license.html>, <http://www.edankert.com/bounce/index.html>, <http://www.net-snmp.org/about/license.html>, <http://www.openmdx.org/#FAQ>, http://www.php.net/license/3_01.txt, <http://srp.stanford.edu/license.txt>, <http://www.schneider.com/blowfish.html>, <http://www.jmock.org/license.html>, <http://xsom.java.net>, <http://benalman.com/about/license/>, <https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>, <http://www.h2database.com/html/license.html#summary>, <http://jsoncpp.sourceforge.net/LICENSE>, <http://jdbc.postgresql.org/license.html>, <http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>, <https://github.com/rantav/hector/blob/master/LICENSE>, <http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>, <http://jibx.sourceforge.net/jibx-license.html>, <https://github.com/lyokato/libgeohash/blob/master/LICENSE>, <https://github.com/hjiang/jsonxx/blob/master/LICENSE>, <https://code.google.com/p/lz4/>, <https://github.com/jedisct1/libsodium/blob/master/>

LICENSE; <http://one-jar.sourceforge.net/index.php?page=documents&file=license>; <https://github.com/EsotericSoftware/kryo/blob/master/license.txt>; <http://www.scala-lang.org/license.html>; <https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>; <http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>; <https://aws.amazon.com/asl/>; <https://github.com/twbs/bootstrap/blob/master/LICENSE>; <https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt>; <https://github.com/documentcloud/underscore-contrib/blob/master/LICENSE>, and <https://github.com/apache/hbase/blob/master/LICENSE.txt>.

This product includes software licensed under the Academic Free License (<http://www.opensource.org/licenses/afl-3.0.php>), the Common Development and Distribution License (<http://www.opensource.org/licenses/cddl1.php>) the Common Public License (<http://www.opensource.org/licenses/cpl1.0.php>), the Sun Binary Code License Agreement Supplemental License Terms, the BSD License (<http://www.opensource.org/licenses/bsd-license.php>), the new BSD License (<http://opensource.org/licenses/BSD-3-Clause>), the MIT License (<http://www.opensource.org/licenses/mit-license.php>), the Artistic License (<http://www.opensource.org/licenses/artistic-license-1.0>) and the Initial Developer's Public License Version 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>).

This product includes software copyright © 2003-2006 Joe Walnes, 2006-2007 XStream Committers. All rights reserved. Permissions and limitations regarding this software are subject to terms available at <http://xstream.codehaus.org/license.html>. This product includes software developed by the Indiana University Extreme! Lab. For further information please visit <http://www.extreme.indiana.edu/>.

This product includes software Copyright (c) 2013 Frank Balluffi and Markus Moeller. All rights reserved. Permissions and limitations regarding this software are subject to terms of the MIT license.

See patents at <https://www.informatica.com/legal/patents.html>.

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, please report them to us in writing at Informatica LLC 2100 Seaport Blvd. Redwood City, CA 94063.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2020-06-18

Table of Contents

Preface	8
Informatica Resources.	8
Informatica Network.	8
Informatica Knowledge Base.	8
Informatica Documentation.	8
Informatica Product Availability Matrixes.	9
Informatica Velocity.	9
Informatica Marketplace.	9
Informatica Global Customer Support.	9
Chapter 1: Product Overview	10
Introduction.	10
Sources.	12
Targets.	13
Informatica Domain.	13
Service Manager.	14
Application Services.	14
PowerCenter Repository.	15
Informatica Administrator.	15
Domain Page.	15
Security Tab.	16
Domain Configuration.	16
PowerCenter Client.	17
PowerCenter Designer.	17
Mapping Architect for Visio.	18
Repository Manager.	19
Workflow Manager.	20
Workflow Monitor.	21
PowerCenter Repository Service.	22
PowerCenter Integration Service.	23
Web Services Hub.	23
Metadata Manager.	24
Metadata Manager Components.	24
Chapter 2: Before You Begin	26
Before You Begin Overview.	26
Getting Started.	26
Using Informatica Administrator in the Tutorial.	27
Using the PowerCenter Client in the Tutorial.	27
Informatica Domain and the PowerCenter Repository.	27

Domain.	27
Administrator.	28
PowerCenter Repository and User Account.	28
PowerCenter Source and Target.	29
Chapter 3: Tutorial Lesson 1.	31
Creating Users and Groups.	31
Logging In to Informatica Administrator.	31
Creating a Group.	32
Creating a User.	33
Creating a Folder in the PowerCenter Repository.	34
Folder Permissions.	34
Connecting to the Repository.	34
Creating a Folder.	35
Creating Source Tables.	36
Chapter 4: Tutorial Lesson 2.	39
Creating Source Definitions.	39
Viewing Source Definitions.	41
Creating Target Definitions and Target Tables.	42
Creating Target Definitions.	42
Creating Target Tables.	44
Chapter 5: Tutorial Lesson 3.	46
Creating a Pass-Through Mapping.	46
Creating a Mapping.	47
Connecting Transformations.	48
Creating Sessions and Workflows.	49
Configuring Database Connections in the Workflow Manager.	50
Creating a Reusable Session.	50
Creating a Workflow.	53
Running and Monitoring Workflows.	55
Opening the Workflow Monitor.	56
Running the Workflow.	56
Previewing Data.	58
Chapter 6: Tutorial Lesson 4.	59
Using Transformations.	59
Creating a Target Definition and Target.	60
Creating a Target Definition.	60
Creating a Target Table.	62
Creating a Mapping with Aggregate Values.	63
Creating a Mapping with T_ITEM_SUMMARY.	63

Creating an Aggregator Transformation.	63
Creating an Expression Transformation.	66
Creating a Lookup Transformation.	68
Connecting the Target.	69
Designer Tips.	70
Using the Overview Window.	70
Arranging Transformations.	71
Creating a Session and Workflow.	71
Creating the Session.	71
Creating the Workflow.	72
Running the Workflow.	73
Viewing the Logs.	74
Chapter 7: Tutorial Lesson 5.	75
Creating a Mapping with Fact and Dimension Tables.	75
Creating Targets.	76
Creating the Mapping.	77
Creating a Filter Transformation.	78
Creating a Sequence Generator Transformation.	79
Creating a Stored Procedure Transformation.	80
Completing the Mapping.	82
Creating a Workflow.	83
Creating the Workflow.	84
Adding a Non-Reusable Session.	84
Defining a Link Condition.	84
Running the Workflow.	86
Chapter 8: Tutorial Lesson 6.	88
Using XML Files.	88
Creating the XML Source.	89
Importing the XML Source.	89
Editing the XML Definition.	91
Creating the Target Definition.	95
Creating a Mapping with XML Sources and Targets.	97
Creating an Expression Transformation.	98
Creating Router Transformations.	98
Completing the Mapping.	100
Creating a Workflow.	102
Appendix A: Naming Conventions.	104
Suggested Naming Conventions.	104
Naming Convention.	104

Appendix B: Glossary.....	106
Index.....	125

Preface

PowerCenter® Getting Started is written for the developers and software engineers who are responsible for implementing a data warehouse. It provides a tutorial to help first-time users learn how to use PowerCenter. *PowerCenter Getting Started* assumes you have knowledge of your operating systems, relational database concepts, and the database engines, flat files, or mainframe systems in your environment. The guide also assumes you are familiar with the interface requirements for your supporting applications.

Informatica Resources

Informatica Network

Informatica Network hosts Informatica Global Customer Support, the Informatica Knowledge Base, and other product resources. To access Informatica Network, visit <https://network.informatica.com>.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base for product resources, including documentation, FAQs, and best practices.
- View product availability information.
- Review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

To access the Knowledge Base, visit <https://kb.informatica.com>. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

To get the latest documentation for your product, browse the Informatica Knowledge Base at https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com.

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. If you are an Informatica Network member, you can access PAMs at

<https://network.informatica.com/community/informatica-network/product-availability-matrixes>.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions.

If you are an Informatica Network member, you can access Informatica Velocity resources at <http://velocity.informatica.com>.

If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that augment, extend, or enhance your Informatica implementations. By leveraging any of the hundreds of solutions from Informatica developers and partners, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at <https://marketplace.informatica.com>.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through Online Support on Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:

<http://www.informatica.com/us/services-and-training/support-services/global-support-centers>.

If you are an Informatica Network member, you can use Online Support at <http://network.informatica.com>.

CHAPTER 1

Product Overview

This chapter includes the following topics:

- [Introduction, 10](#)
- [Informatica Domain, 13](#)
- [PowerCenter Repository, 15](#)
- [Informatica Administrator, 15](#)
- [Domain Configuration, 16](#)
- [PowerCenter Client, 17](#)
- [PowerCenter Repository Service, 22](#)
- [PowerCenter Integration Service, 23](#)
- [Web Services Hub, 23](#)
- [Metadata Manager, 24](#)

Introduction

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

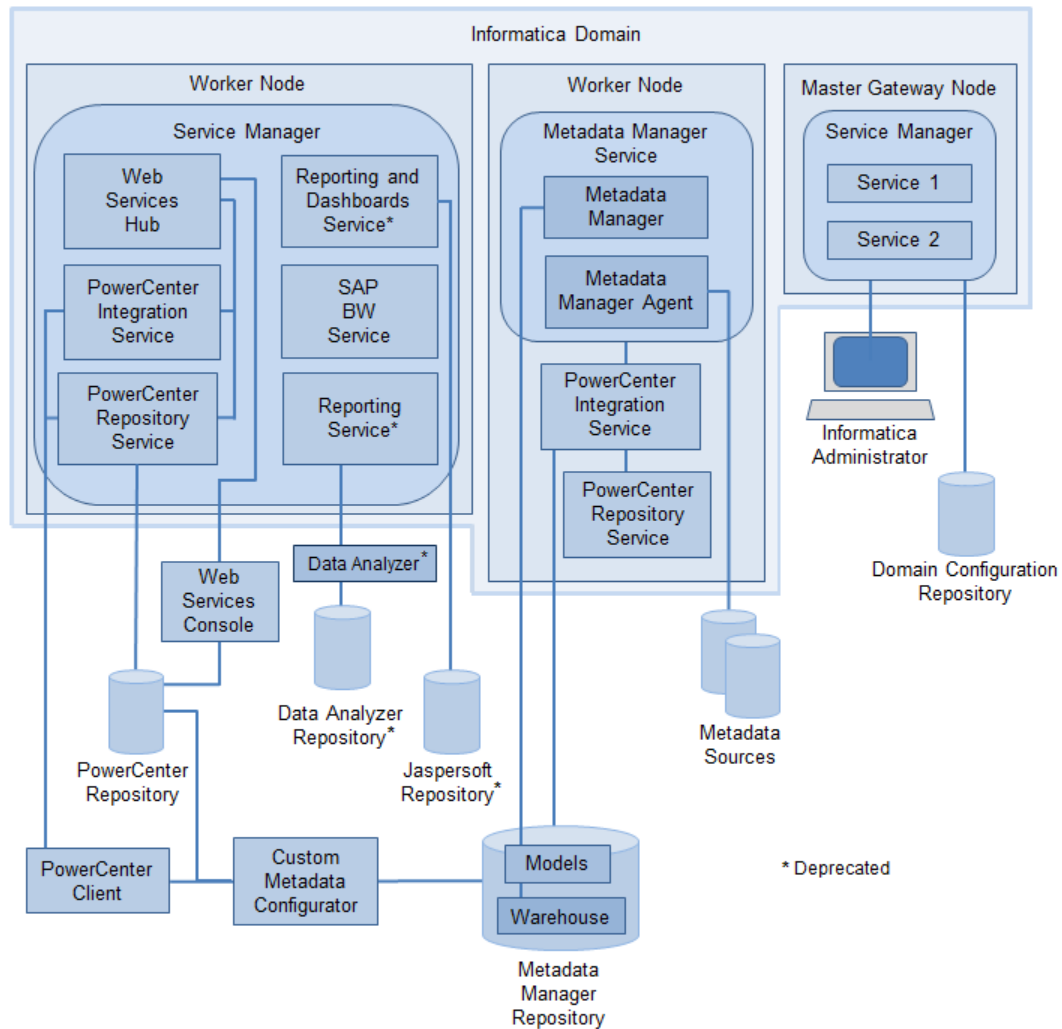
PowerCenter also provides the ability to view and analyze business information and browse and analyze metadata from disparate metadata repositories.

PowerCenter includes the following components:

- **Informatica domain.** The Informatica domain is the primary unit for management and administration within PowerCenter. The Service Manager runs on an Informatica domain. The Service Manager supports the domain and the application services. Application services represent server-based functionality. The domain supports PowerCenter and Informatica application services. PowerCenter application services include the PowerCenter Repository Service, PowerCenter Integration Service, Web Services Hub, and SAP BW Service. Informatica Services include the Data Integration Service, Model Repository Service, and the Analyst Service.
- **PowerCenter repository.** The PowerCenter repository resides in a relational database. The repository database tables contain the instructions required to extract, transform, and load data.
- **Informatica Administrator.** Informatica Administrator is a web application that you use to administer the Informatica domain and PowerCenter security.

- **Domain configuration.** The domain configuration is a set of relational database tables that stores the configuration information for the domain. The Service Manager on the master gateway node manages the domain configuration. The domain configuration is accessible to all gateway nodes in the domain.
- **PowerCenter Client.** The PowerCenter Client is an application used to define sources and targets, build mappings and maplets with the transformation logic, and create workflows to run the mapping logic. The PowerCenter Client connects to the repository through the PowerCenter Repository Service to modify repository metadata. It connects to the Integration Service to start workflows.
- **PowerCenter Repository Service.** The PowerCenter Repository Service accepts requests from the PowerCenter Client to create and modify repository metadata and accepts requests from the Integration Service for metadata when a workflow runs.
- **PowerCenter Integration Service.** The PowerCenter Integration Service extracts data from sources and loads data to targets.
- **Web Services Hub.** Web Services Hub is a gateway that exposes PowerCenter functionality to external clients through web services.
- **SAP BW Service.** The SAP BW Service extracts data from and loads data to SAP NetWeaver BI. If you use PowerExchange® for SAP NetWeaver BI, you must create and enable an SAP BW Service in the Informatica domain.
- **Metadata Manager Service.** The Metadata Manager Service runs the Metadata Manager web application. You can use Metadata Manager to browse and analyze metadata from disparate metadata repositories. Metadata Manager helps you understand and manage how information and processes are derived, how they are related, and how they are used. Metadata Manager stores information about the metadata to be analyzed in the Metadata Manager repository.

The following illustration shows PowerCenter components:



Sources

PowerCenter accesses the following sources:

- **Relational.** Oracle, Sybase ASE, Informix, IBM DB2, Microsoft SQL Server, SAP HANA, and Teradata.
- **File.** Fixed and delimited flat file, COBOL file, XML file, and web log.
- **Application.** You can purchase additional PowerExchange products to access business sources such as Hyperion Essbase, WebSphere MQ, IBM DB2 OLAP Server, JMS, Microsoft Message Queue, PeopleSoft, SAP NetWeaver, SAS, Siebel, TIBCO, and webMethods.
- **Mainframe.** You can purchase PowerExchange to access source data from mainframe databases such as Adabas, Datacom, IBM DB2 OS/390, IBM DB2 OS/400, IDMS, IDMS-X, IMS, and VSAM.
- **Other.** Microsoft Excel, Microsoft Access, and external web services.

Targets

PowerCenter can load data into the following targets:

- **Relational.** Oracle, Sybase ASE, Sybase IQ, Informix, IBM DB2, Microsoft SQL Server, SAP HANA, and Teradata.
- **File.** Fixed and delimited flat file and XML.
- **Application.** You can purchase additional PowerExchange products to load data into business sources such as Hyperion Essbase, WebSphere MQ, IBM DB2 OLAP Server, JMS, Microsoft Message Queue, PeopleSoft EPM, SAP NetWeaver, SAP NetWeaver BI, SAS, Siebel, TIBCO, and webMethods.
- **Mainframe.** You can purchase PowerExchange to load data into mainframe databases such as IBM DB2 for z/OS, IMS, and VSAM.
- **Other.** Microsoft Excel, Microsoft Access, and external web services.

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

Informatica Domain

PowerCenter has a service-oriented architecture that provides the ability to scale services and share resources across multiple machines. The Informatica domain supports the administration of the PowerCenter and Informatica services. A domain is the primary unit for management and administration of services in PowerCenter.

A domain contains the following components:

- **One or more nodes.** A node is the logical representation of a machine in a domain. A domain may contain more than one node. The node that hosts the domain is the master gateway for the domain. You can add other machines as nodes in the domain and configure the nodes to run application services such as the Integration Service or Repository Service. All service requests from other nodes in the domain go through the master gateway.

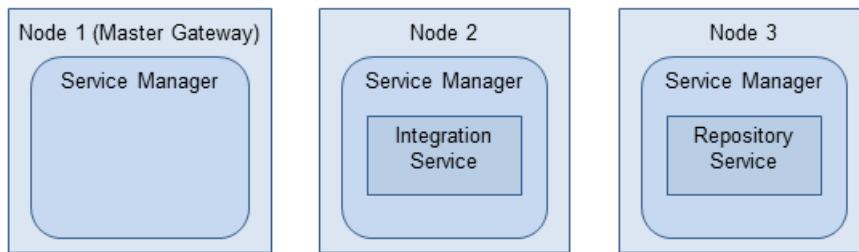
A node runs service processes, which are the runtime representation of the application services running on a node.

- **Service Manager.** The Service Manager is built into the domain to support the domain and the application services. The Service Manager runs on each node in the domain. The Service Manager starts and runs the application services on a machine.
- **Application services.** A group of services that represent Informatica server-based functionality. The application services that run on each node in the domain depend on the way you configure the node and the application service.

You use Informatica Administrator to manage the domain.

If you have the high availability option, you can scale services and eliminate single points of failure for services. The Service Manager and application services can continue running despite temporary network or hardware failures. High availability includes resilience, failover, and recovery for services and tasks in a domain.

The following image shows a Informatica domain with three nodes:



This domain has a master gateway on Node 1. Node 2 runs a PowerCenter Integration Service, and Node 3 runs the PowerCenter Repository Service.

RELATED TOPICS:

- [“Informatica Administrator” on page 15](#)

Service Manager

The Service Manager supports the domain and the application services. The Service Manager performs the following functions:

- **Alerts.** Provides notifications about domain and service events.
- **Authentication.** Authenticates user requests from the Administrator tool, PowerCenter Client, and Metadata Manager.
- **Authorization.** Authorizes user requests for domain objects. Requests can come from the Administrator tool or from *infacmd*.
- **Domain configuration.** Manages domain configuration metadata.
- **Node configuration.** Manages node configuration metadata.
- **Licensing.** Registers license information and verifies license information when you run application services.
- **Logging.** Provides accumulated log events from each service in the domain. You can view logs in the Administrator tool and the Workflow Monitor.
- **User management.** Manages users, groups, roles, and privileges.

Application Services

When you install Informatica, the installation program installs the following application services:

- **Analyst Service.** Manages the connections to Informatica Analyst.
- **Data Integration Service.** Performs data integration tasks for Informatica Analyst, Informatica Developer, and external clients.
- **Model Repository Service.** Stores metadata for Informatica Developer, Informatica Analyst, the Data Integration Service, and the Informatica Administrator.
- **PowerCenter Repository Service.** Manages connections to the PowerCenter repository.
- **PowerCenter Integration Service.** Runs PowerCenter sessions and workflows.
- **Web Services Hub.** Exposes PowerCenter functionality to external clients through web services.

- **SAP BW Service.** Listens for RFC requests from SAP NetWeaver BI and initiates workflows to extract from or load to SAP NetWeaver BI.
- **Metadata Manager Service.** Runs the Metadata Manager application.

PowerCenter Repository

The PowerCenter repository resides in a relational database. The repository stores information required to extract, transform, and load data. It also stores administrative information such as permissions and privileges for users and groups that have access to the repository. PowerCenter applications access the PowerCenter repository through the Repository Service.

You administer the repository through Informatica Administrator and command line programs.

You can develop global and local repositories to share metadata:

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

You can view repository metadata in the Repository Manager. Informatica Metadata Exchange (MX) provides a set of relational views that allow easy SQL access to the PowerCenter metadata repository.

Informatica Administrator

Informatica Administrator is a web application that you use to administer the PowerCenter domain and PowerCenter security. You can also administer application services for the Informatica Analyst and Informatica Developer. Application services for Informatica Analyst and Informatica Developer include the Analyst Service, the Model Repository Service, and the Data Integration Service.

Domain Page

Administer the Informatica domain on the Domain page of the Administrator tool. Domain objects include services, nodes, and licenses.

You can complete the following tasks in the Domain page:

- **Manage application services.** Manage all application services in the domain, such as the Integration Service and Repository Service.
- **Configure nodes.** Configure node properties, such as the backup directory and resources. You can also shut down and restart nodes.
- **Manage domain objects.** Create and manage objects such as services, nodes, licenses, and folders. Folders allow you to organize domain objects and manage security by setting permissions for domain objects.

- **View and edit domain object properties.** View and edit properties for all objects in the domain, including the domain object.
- **View log events.** Use the Log Viewer to view domain, PowerCenter Integration Service, SAP BW Service, Web Services Hub, and PowerCenter Repository Service log events.
- **Generate and upload node diagnostics.** You can generate and upload node diagnostics to the Configuration Support Manager. In the Configuration Support Manager, you can diagnose issues in your Informatica environment and maintain details of your configuration.

Other domain management tasks include applying licenses and managing grids and resources.

Security Tab

You administer PowerCenter security on the Security tab of Informatica Administrator. You manage users and groups that can log in to the following PowerCenter applications:

- Administrator tool
- PowerCenter Client
- Metadata Manager

You can also manage users and groups for the Informatica Developer and Informatica Analyst.

You can complete the following tasks in the Security page:

- **Manage native users and groups.** Create, edit, and delete native users and groups.
- **Configure LDAP authentication and import LDAP users and groups.** Configure a connection to an LDAP directory service. Import users and groups from the LDAP directory service.
- **Manage roles.** Create, edit, and delete roles. Roles are collections of privileges. Privileges determine the actions that users can perform in PowerCenter applications.
- **Assign roles and privileges to users and groups.** Assign roles and privileges to users and groups for the domain and services.
- **Manage operating system profiles.** Create, edit, and delete operating system profiles. An operating system profile is a level of security that the Integration Services uses to run workflows. The operating system profile contains the operating system user name, service process variables, and environment variables. You can configure the Integration Service to use operating system profiles to run workflows.

Domain Configuration

The Service Manager maintains configuration information for an Informatica domain in relational database tables. The configuration is accessible to all gateway nodes in the domain. The domain configuration database stores the following types of information about the domain:

- **Domain configuration.** Domain metadata such as the host names and the port numbers of nodes in the domain. The domain configuration database also stores information on the master gateway node and all other nodes in the domain.
- **Usage.** Includes CPU usage for each application service and the number of Repository Services running in the domain.
- **Users and groups.** Information on the native and LDAP users and the relationships between users and groups.
- **Privileges and roles.** Information on the privileges and roles assigned to users and groups in the domain.

Each time you make a change to the domain, the Service Manager updates the domain configuration database. For example, when you add a node to the domain, the Service Manager adds the node information to the domain configuration. All gateway nodes connect to the domain configuration database to retrieve the domain information and to update the domain configuration.

PowerCenter Client

The PowerCenter Client application consists of the tools to manage the repository and to design mappings, maplets, and sessions to load the data. The PowerCenter Client application has the following tools:

- **Designer.** Use the Designer to create mappings that contain transformation instructions for the Integration Service.
- **Mapping Architect for Visio.** Use the Mapping Architect for Visio to create mapping templates that generate multiple mappings.
- **Repository Manager.** Use the Repository Manager to assign permissions to users and groups and manage folders.
- **Workflow Manager.** Use the Workflow Manager to create, schedule, and run workflows. A workflow is a set of instructions that describes how and when to run tasks related to extracting, transforming, and loading data.
- **Workflow Monitor.** Use the Workflow Monitor to monitor scheduled and running workflows for each Integration Service.

Install the client application on a Microsoft Windows computer.

PowerCenter Designer

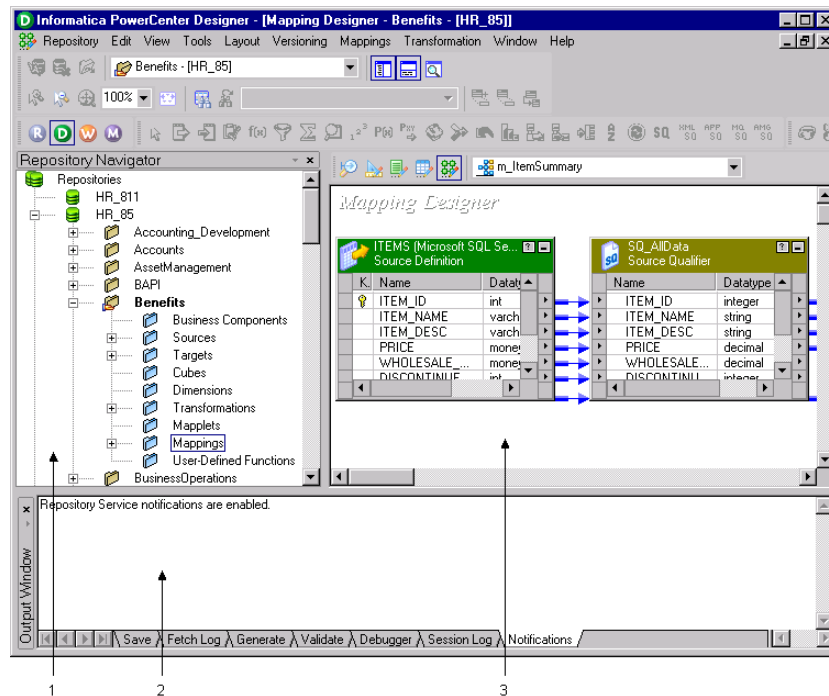
The Designer has the following tools that you use to analyze sources, design target schemas, and build source-to-target mappings:

- **Source Analyzer.** Import or create source definitions.
- **Target Designer.** Import or create target definitions.
- **Transformation Developer.** Develop transformations to use in mappings. You can also develop user-defined functions to use in expressions.
- **Maplet Designer.** Create sets of transformations to use in mappings.
- **Mapping Designer.** Create mappings that the Integration Service uses to extract, transform, and load data.

You can display the following windows in the Designer:

- **Navigator.** Connect to repositories and open folders within the Navigator. You can also copy objects and create shortcuts within the Navigator.
- **Workspace.** Open different tools in this window to create and edit repository objects, such as sources, targets, maplets, transformations, and mappings.
- **Output.** View details about tasks you perform, such as saving your work or validating a mapping.

The following figure shows the default Designer interface:



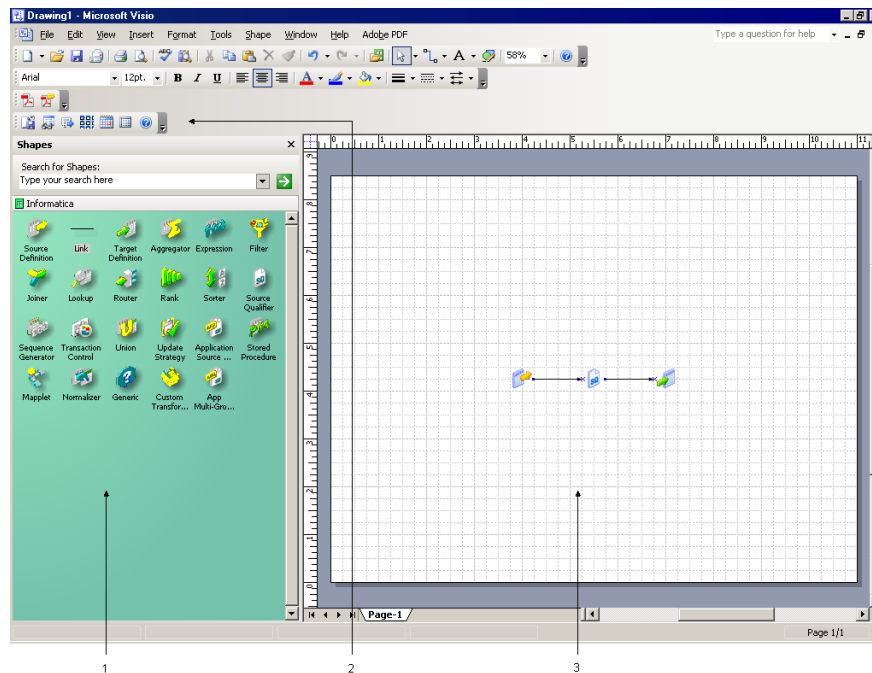
1. Navigator
2. Output
3. Workspace

Mapping Architect for Visio

Use Mapping Architect for Visio to create mapping templates using Microsoft Office Visio. When you work with a mapping template, you use the following main areas:

- **Informatica stencil.** Displays shapes that represent PowerCenter mapping objects. Drag a shape from the Informatica stencil to the drawing window to add a mapping object to a mapping template.
- **Informatica toolbar.** Displays buttons for tasks you can perform on a mapping template. Contains the online help button.
- **Drawing window.** Work area for the mapping template. Drag shapes from the Informatica stencil to the drawing window and set up links between the shapes. Set the properties for the mapping objects and the rules for data movement and transformation.

The following figure shows the Mapping Architect for Visio interface:



1. Informatica Stencil
2. Informatica Toolbar
3. Drawing Window

Repository Manager

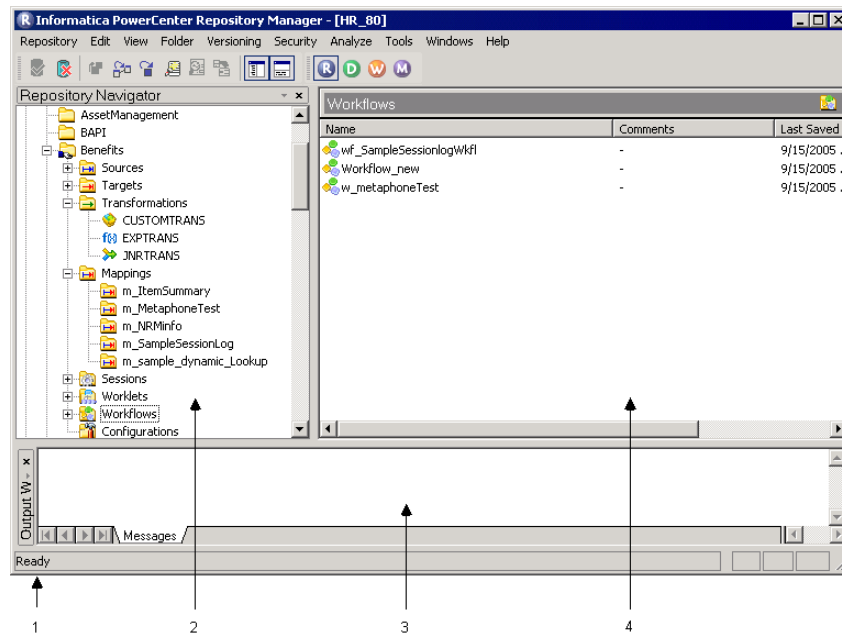
Use the Repository Manager to administer repositories. You can navigate through multiple folders and repositories, and complete the following tasks:

- **Manage user and group permissions.** Assign and revoke folder and global object 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.

The Repository Manager can display the following windows:

- **Navigators.** Displays all objects that you create in the Repository Manager, the Designer, and the Workflow Manager. It is organized first by repository and by folder.
- **Main.** Provides properties of the object selected in the Navigator. The columns in this window change depending on the object selected in the Navigator.
- **Output.** Provides the output of tasks executed within the Repository Manager.

The following figure shows the Repository Manager interface:



1. Status bar
2. Navigator
3. Output
4. Main

Repository Objects

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

- **Source definitions.** Definitions of database objects such as tables, views, synonyms, or files that provide source data.
- **Target definitions.** Definitions of database objects or files that contain the target data.
- **Mappings.** A set of source and target definitions along with transformations containing business logic that you build into the transformation. These are the instructions that the Integration Service uses to transform and move data.
- **Reusable transformations.** Transformations that you use in multiple mappings.
- **Mapplets.** A set of transformations that you use in multiple mappings.
- **Sessions and workflows.** Sessions and workflows store information about how and when the Integration Service moves data. A workflow is a set of instructions that describes how and when to run tasks related to extracting, transforming, and loading data. A session is a type of task that you can put in a workflow. Each session corresponds to a single mapping.

Workflow Manager

In the Workflow Manager, you define a set of instructions to execute tasks such as sessions, emails, and shell commands. This set of instructions is called a workflow.

The Workflow Manager has the following tools to help you develop a workflow:

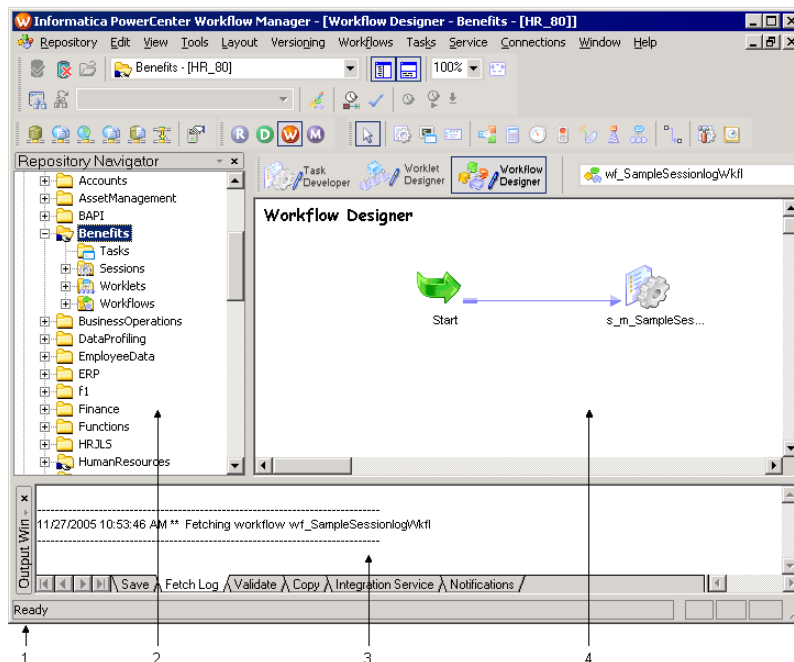
- **Task Developer.** Create tasks you want to accomplish in the workflow.
- **Worklet Designer.** Create a worklet in the Worklet Designer. A worklet is an object that groups a set of tasks. A worklet is similar to a workflow, but without scheduling information. You can nest worklets inside a workflow.
- **Workflow Designer.** Create a workflow by connecting tasks with links in the Workflow Designer. You can also create tasks in the Workflow Designer as you develop the workflow.

When you create a workflow in the Workflow Designer, you add tasks to the workflow. The Workflow Manager includes tasks, such as the Session task, the Command task, and the Email task so you can design a workflow. The Session task is based on a mapping you build in the Designer.

You then connect tasks with links to specify the order of execution for the tasks you created. Use conditional links and workflow variables to create branches in the workflow.

When the workflow start time arrives, the Integration Service retrieves the metadata from the repository to execute the tasks in the workflow. You can monitor the workflow status in the Workflow Monitor.

The following figure shows the Workflow Manager interface:



Workflow Monitor

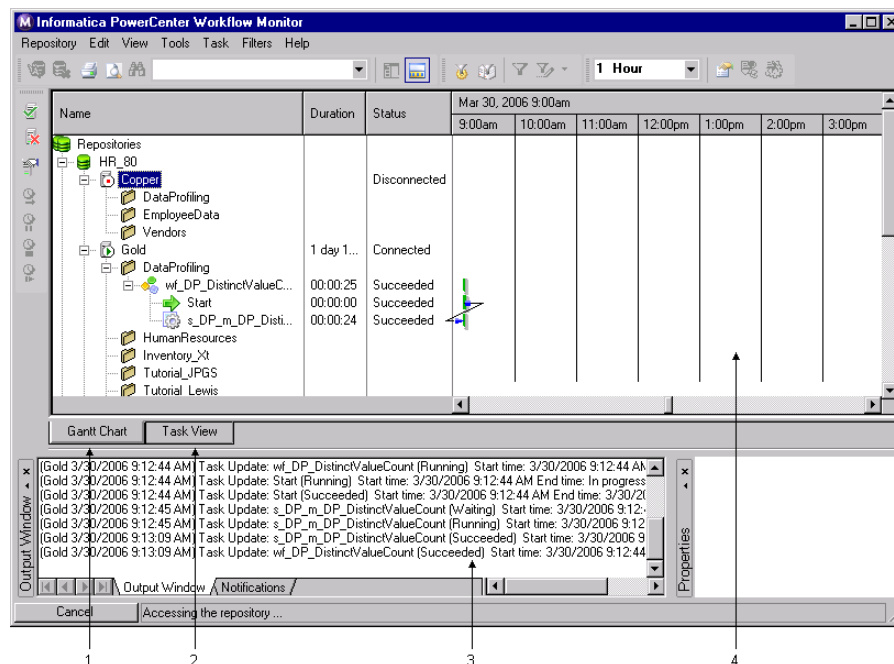
You can monitor workflows and tasks in the Workflow Monitor. You can view details about a workflow or task in Gantt Chart view or Task view. You can run, stop, abort, and resume workflows from the Workflow Monitor. You can view sessions and workflow log events in the Workflow Monitor Log Viewer.

The Workflow Monitor displays workflows that have run at least once. The Workflow Monitor continuously receives information from the Integration Service and Repository Service. It also fetches information from the repository to display historic information.

The Workflow Monitor consists of the following windows:

- **Navigator window.** Displays monitored repositories, servers, and repositories objects.
- **Output window.** Displays messages from the Integration Service and Repository Service.
- **Time window.** Displays progress of workflow runs.
- **Gantt Chart view.** Displays details about workflow runs in chronological format.
- **Task view.** Displays details about workflow runs in a report format.

The following figure shows the Workflow Monitor interface:



1. Gantt chart view
2. Task view
3. Output window
4. Time window

PowerCenter Repository Service

The PowerCenter Repository Service manages connections to the PowerCenter repository from repository clients. A repository client is any PowerCenter component that connects to the repository. The Repository Service is a separate, multi-threaded process that retrieves, inserts, and updates metadata in the repository database tables. The Repository Service ensures the consistency of metadata in the repository.

The Repository Service accepts connection requests from the following PowerCenter components:

- **PowerCenter Client.** Create and store mapping metadata and connection object information in the repository with the PowerCenter Designer and Workflow Manager. Retrieve workflow run status information and session logs with the Workflow Monitor. Create folders, organize and secure metadata, and assign permissions to users and groups in the Repository Manager.
- **Command line programs.** Use command line programs to perform repository metadata administration tasks and service-related functions.
- **PowerCenter Integration Service.** When you start the PowerCenter Integration Service, the service connects to the repository to schedule workflows. When you run a workflow, the Integration Service retrieves workflow task and mapping metadata from the repository. The Integration Service writes workflow status to the repository.
- **Web Services Hub.** When you start the Web Services Hub, it connects to the repository to access web-enabled workflows. The Web Services Hub retrieves workflow task and mapping metadata from the repository and writes workflow status to the repository.
- **SAP BW Service.** Listens for RFC requests from SAP NetWeaver BI and initiates workflows to extract from or load to SAP NetWeaver BI.

You install the Repository Service when you install PowerCenter Services. After you install the PowerCenter Services, you can use Informatica Administrator to manage the Repository Service.

PowerCenter Integration Service

The PowerCenter Integration Service reads workflow information from the repository. The Integration Service connects to the repository through the Repository Service to fetch metadata from the repository.

A workflow is a set of instructions that describes how and when to run tasks related to extracting, transforming, and loading data. The Integration Service runs workflow tasks. A session is a type of workflow task. A session is a set of instructions that describes how to move data from sources to targets using a mapping.

A session extracts data from the mapping sources and stores the data in memory while it applies the transformation rules that you configure in the mapping. The Integration Service loads the transformed data into the mapping targets.

Other workflow tasks include commands, decisions, timers, pre-session SQL commands, post-session SQL commands, and email notification.

The Integration Service can combine data from different platforms and source types. For example, you can join data from a flat file and an Oracle source. The Integration Service can also load data to different platforms and target types.

You install the PowerCenter Integration Service when you install PowerCenter Services. After you install the PowerCenter Services, you can use Informatica Administrator to manage the Integration Service.

Web Services Hub

The Web Services Hub is the application service in the Informatica domain that is a web service gateway for external clients. The Web Services Hub processes SOAP requests from client applications that access

PowerCenter functionality through web services. Web service clients access the PowerCenter Integration Service and PowerCenter Repository Service through the Web Services Hub.

The Web Services Hub hosts the following web services:

- **Batch web services.** Includes operations to run and monitor the sessions and workflows. Batch web services also include operations that can access repository metadata. Batch web services install with PowerCenter.
- **Real-time web services.** Workflows enabled as web services that can receive requests and generate responses in SOAP message format. Create real-time web services when you enable PowerCenter workflows as web services.

Use Informatica Administrator to configure and manage the Web Services Hub. Use the Web Services Hub Console to view information about the web service and to download WSDL files to create web service clients.

Metadata Manager

Informatica Metadata Manager is a PowerCenter web application to browse, analyze, and manage metadata from disparate metadata repositories. Metadata Manager helps you understand how information and processes are derived, how they are related, and how they are used.

Metadata Manager extracts metadata from application, business intelligence, data integration, data modeling, and relational metadata sources. Metadata Manager uses PowerCenter workflows to extract metadata from metadata sources and load it into a centralized metadata warehouse called the Metadata Manager warehouse.

You can use Metadata Manager to browse and search metadata objects, trace data lineage, analyze metadata usage, and perform data profiling on the metadata in the Metadata Manager warehouse. You can also create and manage business glossaries.

The Metadata Manager Service in the Informatica domain runs the Metadata Manager application. Create a Metadata Manager Service in the Informatica Administrator to configure and run the Metadata Manager application.

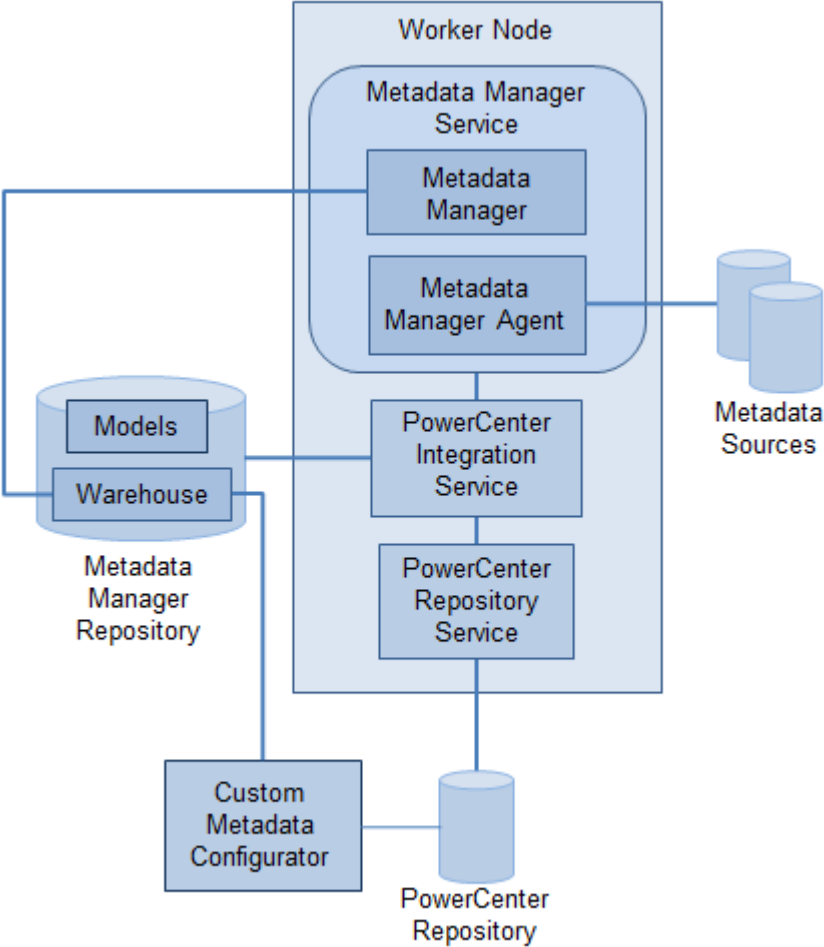
Metadata Manager Components

The Metadata Manager web application includes the following components:

- **Metadata Manager Service.** An application service in an Informatica domain that runs the Metadata Manager application and manages connections between the Metadata Manager components. You create and configure the Metadata Manager Service in the Administrator tool.
- **Metadata Manager application.** Manages the metadata in the Metadata Manager warehouse. Create and load resources in Metadata Manager. After you use Metadata Manager to load metadata for a resource, you can use the Metadata Manager application to browse and analyze metadata for the resource. You can also create custom models and manage security on the metadata in the Metadata Manager warehouse.
- **Metadata Manager Agent.** Runs within the Metadata Manager application or on a separate machine. Metadata Exchanges uses the Metadata Manager Agent to extract metadata from metadata sources and convert it to IME interface-based format.
- **Metadata Manager repository.** A centralized location in a relational database that stores metadata from disparate metadata sources. The repository also stores Metadata Manager metadata and the packaged and custom models for each metadata source type.

- **PowerCenter repository.** Stores the PowerCenter workflows that extract source metadata from IME-based files and load it into the Metadata Manager warehouse.
- **PowerCenter Integration Service.** Runs the workflows that extract the metadata from IME-based files and load it into the Metadata Manager warehouse.
- **PowerCenter Repository Service.** Manages connections to the PowerCenter repository. The repository stores the workflows that extract metadata from IME interface-based files.
- **Custom Metadata Configurator.** Creates custom resource templates to extract metadata from metadata sources for which Metadata Manager does not package a resource type.

The following figure shows the Metadata Manager components:



CHAPTER 2

Before You Begin

This chapter includes the following topics:

- [Before You Begin Overview, 26](#)
- [Informatica Domain and the PowerCenter Repository, 27](#)
- [PowerCenter Source and Target, 29](#)

Before You Begin Overview

PowerCenter Getting Started provides lessons that introduce you to PowerCenter and how to use it to load transformed data into file and relational targets. The lessons in this book are for PowerCenter beginners.

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

- Create users and groups.
- Add source definitions to the repository.
- Create targets and add their definitions to the repository.
- Map data between sources and targets.
- Instruct the Integration Service to write data to targets.
- Monitor the Integration Service as it writes data to targets.

In general, you can set the pace for completing the tutorial. However, you should complete an entire lesson in one session. Each lesson builds on a sequence of related tasks.

For more information, case studies, and updates about using Informatica products, refer to Informatica MySupport Portal at the following location: <http://mysupport.informatica.com>.

Getting Started

The administrator must install and configure the PowerCenter Services and Client. Verify that the administrator has completed the following steps:

- Installed the PowerCenter Services and created an Informatica domain.
- Created a PowerCenter repository.
- Installed the PowerCenter Client.

You also need information to connect to the Informatica domain, the repository, and the source and the target database tables. Use the tables in [“Informatica Domain and the PowerCenter Repository” on page 27](#)

to write down the domain and repository information. Use the tables in [“PowerCenter Source and Target” on page 29](#) to write down the source and target connectivity information. Contact the administrator for the necessary information.

Before you begin the lessons, read [Chapter 1, “Product Overview” on page 10](#). The product overview explains the different components that work together to extract, transform, and load data.

Using Informatica Administrator in the Tutorial

Informatica Administrator is the administration tool for the Informatica domain. In this tutorial, use Informatica Administrator to perform the following tasks:

- **Create a group with all privileges on a PowerCenter Repository Service.** The privileges allow users to design mappings and run workflows in the PowerCenter Client.
- **Create a user account and assign it to the group.** The user inherits the privileges of the group.

Using the PowerCenter Client in the Tutorial

The PowerCenter Client consists of applications that you use to design mappings and mapplets, create sessions and workflows to load the data, and monitor workflow progress.

In this tutorial, you learn about the following applications and tools:

- **PowerCenter Repository Manager.** Create a folder in the Repository Manager to store the metadata you create in the lessons.
- **PowerCenter Designer.** Create the source and the target definitions. Create mappings that contain transformation instructions for the PowerCenter Integration Service. In this tutorial, you learn about the following tools in the Designer:
 - **Source Analyzer.** Import or create source definitions.
 - **Target Designer.** Import or create target definitions. You also create tables in the target database based on the target definitions.
 - **Mapping Designer.** Create mappings that the PowerCenter Integration Service uses to extract, transform, and load data.
- **Workflow Manager.** Create and run the workflows and the tasks in the Workflow Manager. A workflow is a set of instructions that describes how and when to run tasks to extract, transform, and load data.
- **Workflow Monitor.** Monitor scheduled and running workflows for each Integration Service.

Informatica Domain and the PowerCenter Repository

To use the lessons in this book, you need to connect to the Informatica domain and a PowerCenter repository in the domain. Log in to Informatica Administrator using the default administrator account.

Domain

Use the tables in this section to record the domain connectivity and default administrator information. If necessary, contact the Informatica administrator for the information.

The following table can be used to record the domain information:

Domain Information	Value
Domain Name	
Gateway Host	
Gateway Port	

Administrator

The following table can be used to record the information you need to connect to Informatica Administrator as the default administrator:

Informatica Admin Information	Value
Default Administrator User Name	Administrator
Default Administrator Password	

Use the default administrator account for the lessons [“Creating Users and Groups” on page 31](#). For all other lessons, you use the user account that you create in lesson [“Creating a User” on page 33](#) to log in to the PowerCenter Client.

Note: The default administrator user name is Administrator. If you do not have the password for the default administrator, ask the Informatica administrator to provide this information or set up a domain administrator account that you can use. Record the user name and password of the domain administrator.

PowerCenter Repository and User Account

The following table can be used to record the information you need to connect to the PowerCenter repository in each PowerCenter Client tool:

Repository Information	Value
Repository Name	
User Name	
Password	
Security Domain	Native

Note: Ask the Informatica administrator to provide the name of a PowerCenter repository where you can create the folder, mappings, and workflows in this tutorial. The user account you use to connect to the repository is the user account you create in [“Creating a User” on page 33](#).

PowerCenter Source and Target

In this tutorial, you create mappings to read data from relational tables, transform the data, and write the transformed data to relational tables. The PowerCenter Client uses ODBC drivers to connect to the relational tables.

You must have a relational database available and an ODBC data source to connect to the tables in the relational database. You can use separate ODBC data sources to connect to the source tables and target tables.

The following table can be used to record the information you need for the ODBC data sources:

ODBC Data Source Information	Source Connection	Target Connection
ODBC Data Source Name		
Database User Name		
Database Password		

The following table can be used to record the information you need to create database connections in the Workflow Manager:

Workflow Manager Connectivity Information	Source Connection Object	Target Connection Object
Database Type		
User Name		
Password		
Connect String		
Code Page		
Database Name		
Server Name		
Domain Name		
Note: You may not need all properties in this table.		

The following table lists the native connect string syntax to use for different databases:

Database	Native Connect String	Example
IBM DB2	<i>dbname</i>	mydatabase
Microsoft SQL Server	<i>servername@dbname</i>	sqlserver@mydatabase
Oracle	<i>dbname.world</i> (same as TNSNAMES entry)	oracle.world
Sybase ASE	<i>servername@dbname</i>	sambrown@mydatabase
Teradata	Teradata* ODBC_data_source_name or ODBC_data_source_name@db_name or ODBC_data_source_name@db_user_name	TeradataODBC TeradataODBC@mydatabase TeradataODBC@sambrown

CHAPTER 3

Tutorial Lesson 1

This chapter includes the following topics:

- [Creating Users and Groups, 31](#)
- [Creating a Folder in the PowerCenter Repository, 34](#)
- [Creating Source Tables, 36](#)

Creating Users and Groups

You need a user account to access the services and objects in the Informatica domain and to use the PowerCenter Client. Users can perform tasks in PowerCenter based on the privileges and permissions assigned to them.

When you install PowerCenter, the installer creates a default administrator user account. You can use the default administrator account to initially log in to the Informatica domain and create PowerCenter services, domain objects, and user accounts.

The privileges assigned to a user determine the task or set of tasks a user or group of users can perform in PowerCenter applications. You can organize users into groups based on the tasks they are allowed to perform in PowerCenter. Create a group and assign it a set of privileges. Then assign users who require the same privileges to the group. All users who belong to the group can perform the tasks allowed by the group privileges.

In this lesson, you complete the following tasks:

1. Log in to Informatica Administrator using the default administrator account.
If necessary, ask the PowerCenter administrator for the user name and password. Otherwise, ask the PowerCenter administrator to complete the lessons in this chapter for you.
2. In the Administrator tool, create the TUTORIAL group and assign privileges to the TUTORIAL group.
3. Create a user account and assign the user to the TUTORIAL group.
4. Log in to the PowerCenter Repository Manager with the new user account.

Logging In to Informatica Administrator

Use the default administrator user name and password you entered in [“Domain” on page 27](#). Otherwise, ask the Informatica administrator to perform the tasks in this section for you.

1. Open Microsoft Internet Explorer or Mozilla Firefox.

- In the Address field, enter the following URL for the Informatica Administrator login page:

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

If you configure HTTPS for Informatica Administrator, the URL redirects to the HTTPS enabled site. If the node is configured for HTTPS with a keystore that uses a self-signed certificate, a warning message appears. To enter the site, accept the certificate. The Informatica Administrator login page appears.
- Enter the default administrator user name and password.
 Use the Administrator user name and password you recorded in ["Administrator" on page 28](#).
- If the security domain is configured for LDAP, select **Native**.
- Click **Login**.
- If the Administration Assistant appears, click **Administrator**.

Creating a Group

In the following steps, you create a group and assign privileges to the group.

- In the Administrator tool, click the **Security** tab.
- In Groups, **Actions > Create Group**.
- Enter the information for the group.

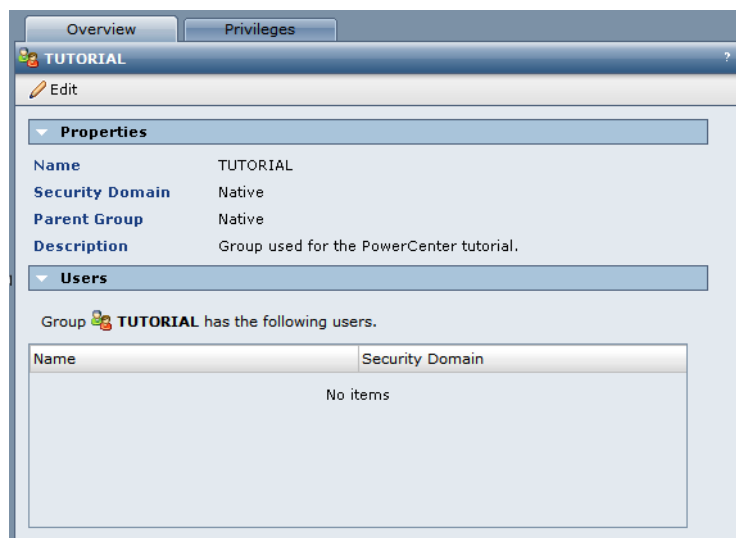
The following table describes the information you can enter for the group:

Property	Value
Name	TUTORIAL
Description	Group used for the PowerCenter tutorial.

- Click **OK** to save the group.

The TUTORIAL group appears on the list of native groups in the Groups section of the Navigator. The Properties area contains the name, security domain, parent group, and description for the new group.

The following image shows the Properties area and the Users area that contains no items:



- Click the **Privileges** tab.

6. Click **Edit**.
7. In the **Edit Roles and Privileges** dialog box, click the **Privileges** tab.
8. Expand the privileges list for the PowerCenter Repository Service that you plan to use.
9. Click the box next to the Repository Service name to assign all privileges to the TUTORIAL group.
10. Click **OK**.

Users in the TUTORIAL group have the privileges to create workflows in any folder for which they have read and write permission.

Creating a User

The final step is to create a user account and add the user to the TUTORIAL group. You use this account throughout the rest of this tutorial.

1. On the **Security** tab, click **Users, Actions > Create User**.

The **Create User** dialog box appears.

2. Enter a login name for the user account.

You use this user name when you log in to the PowerCenter Client to complete the rest of the tutorial.

3. Enter a password and confirm.

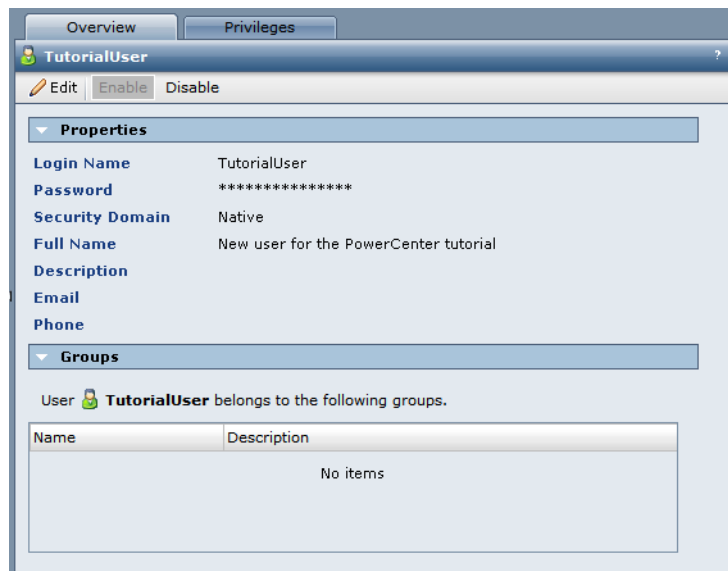
You must retype the password. Do not copy and paste the password.

4. Enter the full name of the user.

5. Click **OK** to save the user account.

The details for the new user account are in the Properties area.

The following image shows the Properties area with the details for the new user and the Groups area that contains no items:



6. Click the **Overview** tab.
7. Click **Edit**.
8. In the **Edit Properties** dialog box, click the **Groups** tab.
9. Select the group name TUTORIAL in the All Groups column and click **Add**.

The TUTORIAL group appears in Assigned Groups list.

10. Click **OK** to save the group assignment.

The user account has all the privileges for the TUTORIAL group.

Creating a Folder in the PowerCenter Repository

In this section, you create a tutorial repository folder. You save all objects you create in the tutorial to this folder.

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

Folder Permissions

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

Folder permissions work closely with privileges. Privileges grant access to specific tasks, while permissions grant access to specific folders with read, write, and execute access. Folders have the following types of permissions:

- 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 run or schedule workflows in the folder.

When you create a folder, you are the owner of the folder. The folder owner has all permissions on the folder which cannot be changed.

Connecting to the Repository

To complete this tutorial, you need to connect to the PowerCenter repository.

1. Launch the PowerCenter Repository Manager.
2. Click **Repository > Add Repository**.

The **Add Repository** dialog box appears.

3. Enter the repository and user name.

Use the name of the repository in [“PowerCenter Repository and User Account” on page 28](#).

Use the name of the user account you created in [“Creating a User” on page 33](#).

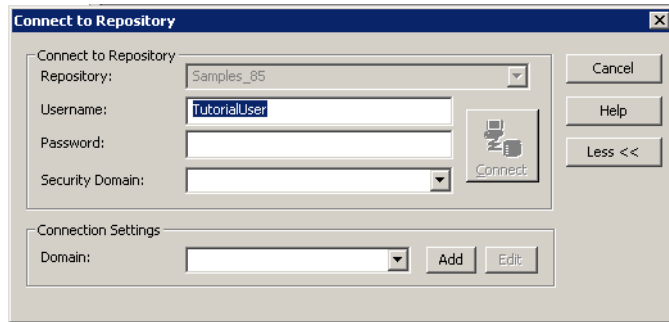
4. Click **OK**.

The repository appears in the Navigator.

5. Click **Repository > Connect**.

The **Connect to Repository** dialog box appears.

The following image shows the **Connect to Repository** dialog box:



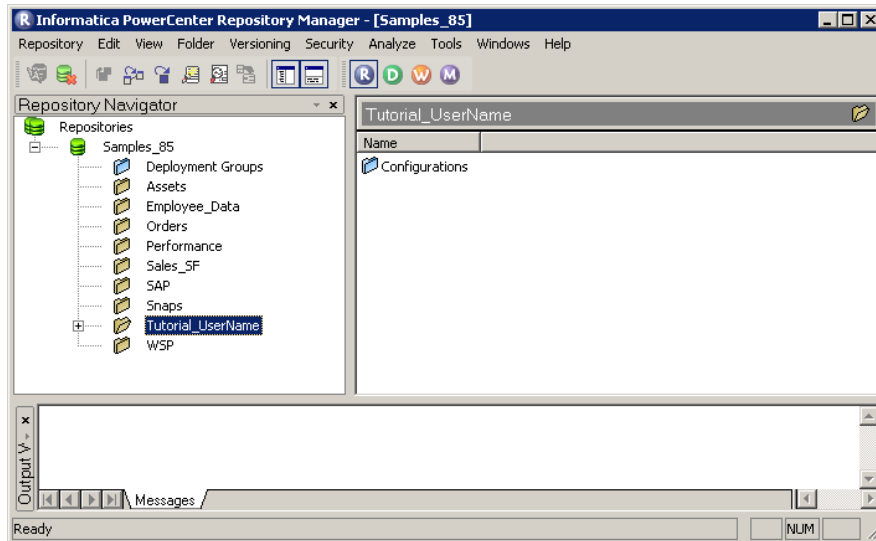
6. In the connection settings section, click **Add** to add the domain connection information.
The **Add Domain** dialog box appears.
7. Enter the domain name, gateway host, and gateway port number from ["Domain" on page 27](#).
8. Click **OK**.
If a message indicates that the domain exists, click **Yes** to replace the existing domain.
9. In the **Connect to Repository** dialog box, enter the user and password for the tutorial user.
10. Select the Native security domain.
11. Click **Connect**.

Creating a Folder

For this tutorial, you create a folder where you will define the data sources and targets, build mappings, and run workflows in later lessons.

1. In the Repository Manager, click **Folder > Create**.
2. Enter your name prefixed by Tutorial_ as the name of the folder.
By default, the user account logged in is the owner of the folder and has full permissions on the folder.
3. Click **OK**.
The Repository Manager displays a message that the folder has been successfully created.
4. Click **OK**.
The new folder appears in the repository.

The following image shows the new folder in the Repository Manager:



5. Exit the Repository Manager.

Creating Source Tables

Before you continue with the other lessons, you need to create the source tables in the database. In this section, you run an SQL script in the Target Designer to create sample source tables. The SQL script creates sources with 7-bit ASCII table names and data. The source tables exist in the database if the tutorial is run multiple times.

When you run the SQL script, you create the following source tables:

- CUSTOMERS
- DEPARTMENT
- DISTRIBUTORS
- EMPLOYEES
- ITEMS
- ITEMS_IN_PROMOTIONS
- JOBS
- MANUFACTURERS
- ORDERS
- ORDER_ITEMS
- PROMOTIONS
- STORES

The Target Designer generates SQL based on the definitions in the workspace. Generally, you use the Target Designer to create target tables in the target database. In this lesson, you use this feature to generate the source tutorial tables from the tutorial SQL scripts that ship with the product. When you run the SQL script,

you also create a stored procedure that you will use to create a Stored Procedure transformation in another lesson.

1. Launch the Designer, double-click the icon for the repository, and log in to the repository.
Use your user profile to open the connection.
2. Double-click the Tutorial_*yourname* folder.
3. Click **Tools > Target Designer** to open the Target Designer.
4. Click **Targets > Create**.
The **Create Target Table** dialog box appears.
You must create a dummy target definition to access the **Generate/Execute SQL** option.
5. Enter any name for the target and select any database type.
6. Click **Create**.
An empty definition appears in the workspace.
7. Click **Done**.
8. Click **Targets > Generate/Execute SQL**.
The **Database Object Generation** dialog box gives you several options for creating tables.
9. Click the **Connect** button to connect to the source database.
10. Select the ODBC data source that you created in order to connect to the source database.
Use the information you entered in ["PowerCenter Source and Target" on page 29](#).
11. Enter the database user name and password and click **Connect**.
You now have an open connection to the source database. When you are connected, the **Disconnect** button appears and the ODBC name of the source database appears in the dialog box.
12. Make sure the **Output** window is open at the bottom of the Designer.
If it is not open, click **View > Output**.
13. Click the **Browse** button to find the SQL file.
The SQL file is installed in the following directory:
C:\PowerCenterClientInstallationDir\client\bin
14. Select the SQL file appropriate to the source database platform you are using. Click **Open**.
The following table describes the database platform name and the corresponding SQL file:

Platform	File
Informix	smp1_inf.sql
Microsoft SQL Server	smp1_ms.sql
Oracle	smp1_ora.sql
Sybase ASE	smp1_syb.sql
DB2	smp1_db2.sql
Teradata	smp1_tera.sql

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

15. Click **Execute SQL File**.

The database executes the SQL script to create the sample source database objects and to insert values into the source tables. While the script is running, the Output window displays the progress. The Designer generates and executes SQL scripts in Unicode (UCS-2) format.

16. When the script completes, click **Disconnect**, and click **Close**.

CHAPTER 4

Tutorial Lesson 2

This chapter includes the following topics:

- [Creating Source Definitions, 39](#)
- [Creating Target Definitions and Target Tables, 42](#)

Creating Source Definitions

Now that you have created the source tables with sample data, you are ready to create the source definitions in the repository. The repository contains a description of source tables, not the actual data contained in them. After you add these source definitions to the repository, you use them in a mapping.

1. In the Designer, click **Tools > Source Analyzer** to open the Source Analyzer.
2. Double-click the tutorial folder to view its contents.
Every folder contains nodes for sources, targets, schemas, mappings, mapplets, cubes, dimensions, user-defined functions, and transformations.
3. Click **Sources > Import from Database**.
4. From the **ODBC data source** button, select the ODBC data source that you created to access source tables.
5. Enter the user name and password to connect to this database. Also, enter the name of the source table owner, if necessary.
Use the database connection information you entered in [“PowerCenter Source and Target” on page 29](#).
In Oracle, the owner name is the same as the user name. Make sure that the owner name is in all caps. For example, JDOE.
6. Click **Connect**.
7. In the Select tables list, expand the database owner and the TABLES heading.
A list of all the tables you created by running the SQL script appears in addition to any tables already in the database.
8. Select the following tables:
 - CUSTOMERS
 - DEPARTMENT
 - DISTRIBUTORS
 - EMPLOYEES

- ITEMS
- ITEMS_IN_PROMOTIONS
- JOBS
- MANUFACTURERS
- ORDERS
- ORDER_ITEMS
- PROMOTIONS
- STORES

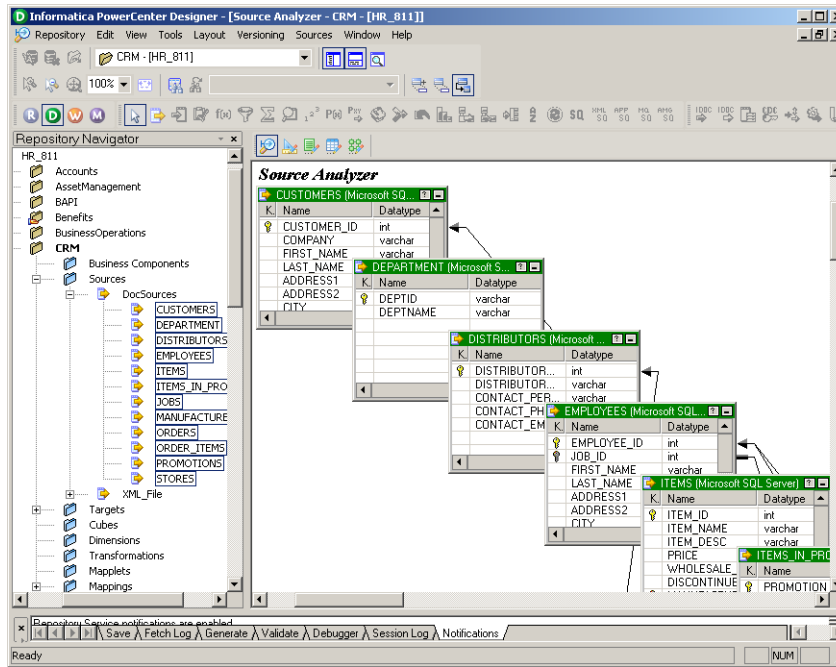
Hold down the CTRL key to select multiple tables. Or, hold down the SHIFT key to select a block of tables. You may need to scroll down the list of tables to select all tables.

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

9. Click **OK** to import the source definitions into the repository.

The Designer displays the newly imported sources in the workspace. You can click **Layout > Scale To Fit** to arrange all definitions in the workspace.

The following image shows the Designer with the newly imported sources in the workspace:



A database definition (DBD) node appears under the Sources node in the tutorial folder. This entry has the same name as the ODBC data source that you used to import sources. If you double-click the DBD node, the list of all the imported sources appears.

Viewing Source Definitions

You can view details for each source definition.

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

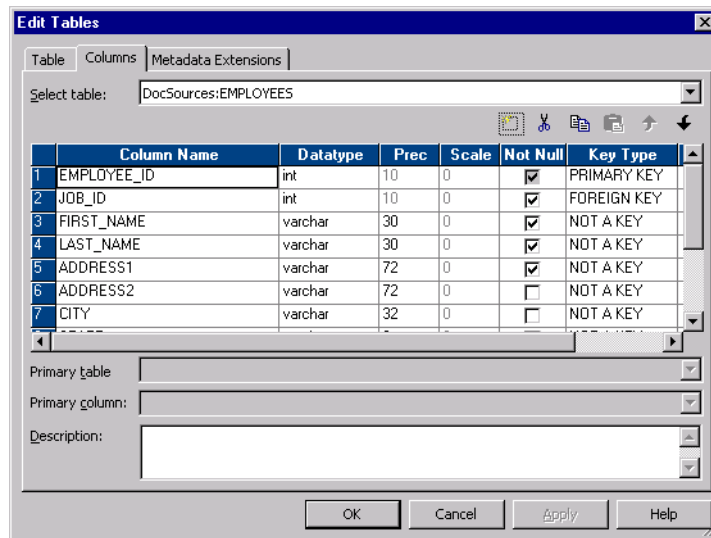
The **Edit Tables** dialog box appears and displays all the properties of this source definition. The **Table** tab shows the name of the table, owner name, and the database type. You can add a comment in the Description section. Business name is empty.

2. Click the **Columns** tab.

The **Columns** tab displays the column descriptions for the source table.

Note: The source definition must match the structure of the source table. Therefore, you must not modify source column definitions after you import them.

The following image shows the **Columns** tab for EMPLOYEES source definition:



3. Click the **Metadata Extensions** tab.

Metadata extensions allow you to extend the metadata stored in the repository by associating information with individual repository objects. For example, you can store contact information, such as name or email address, with the sources you create.

In this lesson, you create user-defined metadata extensions that define the date you created the source definition and the name of the person who created the source definition.

4. Click the **Add Metadata Extension** button.
5. Name the row SourceCreationDate and enter today's date as the value.
6. Click the **Add Metadata Extension** button to add another metadata extension and name it SourceCreator.
7. Enter your first name as the value in the SourceCreator row.
8. Click **Apply**.
9. Click **OK** to close the dialog box.
10. Click **Repository > Save** to save the changes to the repository.

Creating Target Definitions and Target Tables

You can import target definitions from existing target tables, or you can create the definitions and then generate and run the SQL to create the target tables. In this lesson, you create a target definition in the Target Designer, and then create a target table based on the definition.

Creating Target Definitions

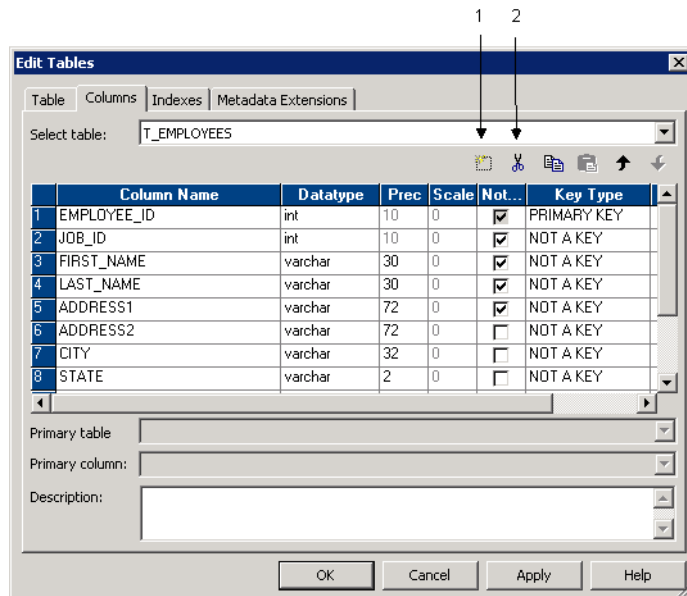
The next step is to create the metadata for the target tables in the repository. The actual tables that the target definitions describe do not exist yet.

Target definitions define the structure of tables in the target database or the structure of file targets the Integration Service creates when you run a session. If you add a relational target definition to the repository that does not exist in a database, you need to create target table. You do this by generating and executing the necessary SQL code within the Target Designer.

In the following steps, you copy the EMPLOYEES source definition into the Target Designer to create the target definition. Then, you modify the target definition by deleting and adding columns to create the definition you want.

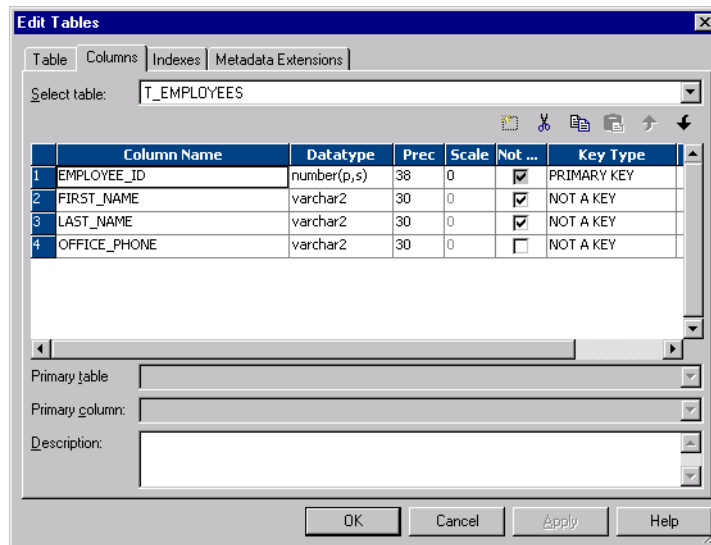
1. In the Designer, click **Tools > Target Designer** to open the Target Designer.
2. Drag the EMPLOYEES source definition from the Navigator to the Target Designer workspace.
The Designer creates a target definition, EMPLOYEES, with the same column definitions as the EMPLOYEES source definition and the same database type.
Next, modify the target column definitions.
3. Double-click the title bar of the EMPLOYEES target definition to open it.
4. Click **Rename** and name the target definition T_EMPLOYEES.
Note: If you need to change the database type for the target definition, you can select the correct database type when you edit the target definition.
5. Click the **Columns** tab.
The target column definitions are the same as the EMPLOYEES source definition.

The following image shows the **Columns** tab for the target definition T_EMPLOYEES:



1. Add button
 2. Delete button
6. Select the following columns and click the **Delete** button.
- JOB_ID
 - ADDRESS1
 - ADDRESS2
 - CITY
 - STATE
 - POSTAL_CODE
 - HOME_PHONE
 - EMAIL

The following image shows the final target definition when you finish creating the target definition:



The EMPLOYEE_ID column is a primary key. The primary key cannot accept null values. The Designer selects Not Null and enables the Not Null option. You now have a column ready to receive data from the EMPLOYEE_ID column in the EMPLOYEES source table.

7. Click **OK** to save the changes and close the dialog box.
8. Click **Repository > Save**.

Creating Target Tables

Use the Target Designer to run an existing SQL script to create target tables.

Note: When you use the Target Designer to generate SQL, you can choose to drop the table in the database before you create it. To do this, select the Drop Table option. If the target database already contains tables, make sure it does not contain a table with the same name as the table you plan to create. If the table exists in the database, you lose the existing table and data.

1. In the workspace, select the T_EMPLOYEES target definition.
2. Click **Targets > Generate/Execute SQL**.

The **Database Object Generation** dialog box appears.

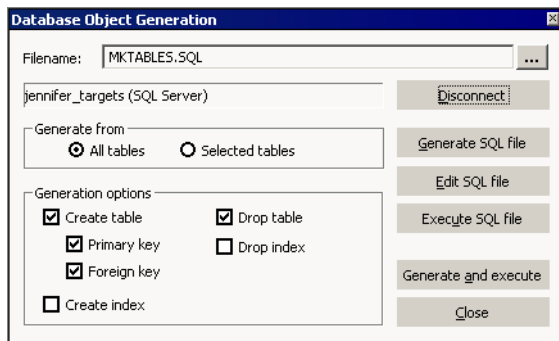
3. In the File Name field, enter the following text to generate SQL script called MKTABLES.SQL

```
C:\<PowerCenterClientInstallationDir>\MKTABLES.SQL
```

If you installed the PowerCenter Client in a different location, enter the appropriate drive letter and directory.

4. If you are connected to the source database from the previous lesson, click **Disconnect**, and then click **Connect**.
5. Select the ODBC data source to connect to the target database.
6. Enter the necessary user name and password, and then click **Connect**.

The following image shows the **Database Object Generation** dialog box with the File Name MKTABLES.SQL:



7. Select the **Create Table** and **Drop Table**.
8. Click **Generate and Execute**.

The Designer runs the DDL code in MKTABLES.SQL to create T_EMPLOYEES.

The results appear on the **Generate** tab in the Output window

Optionally, to edit the contents of the SQL file, click the **Edit SQL File**.

9. Click **Close** to exit the **Database Object Generation** dialog box.

CHAPTER 5

Tutorial Lesson 3

This chapter includes the following topics:

- [Creating a Pass-Through Mapping, 46](#)
- [Creating Sessions and Workflows, 49](#)
- [Running and Monitoring Workflows, 55](#)

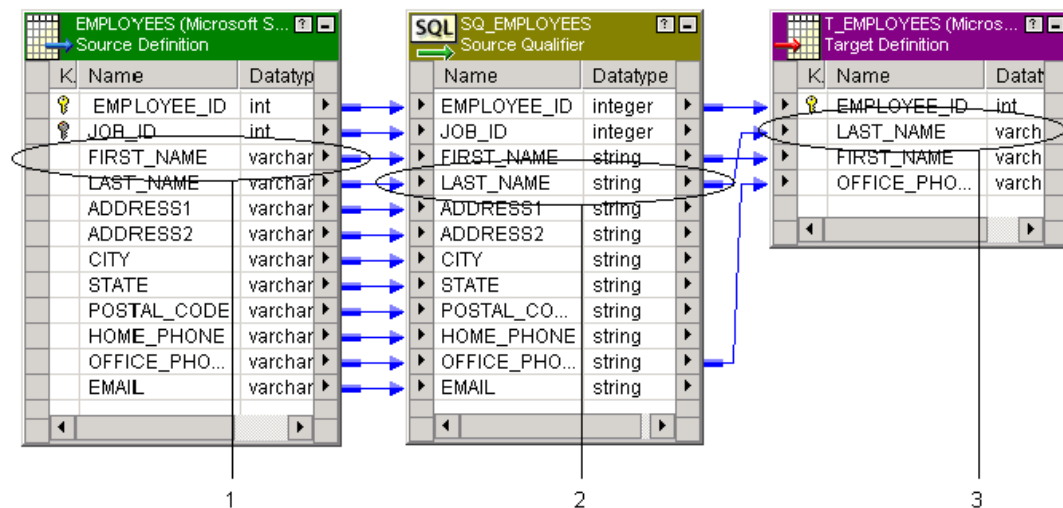
Creating a Pass-Through Mapping

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

The next step is to create a mapping to depict the flow of data between sources and targets. For this step, you create a pass-through mapping. A pass-through mapping inserts all the source rows into the target.

To create and edit mappings, you use the Mapping Designer tool in the Designer. You add transformations to a mapping that depict how the Integration Service extracts and transforms data before it loads a target.

The following figure shows a mapping between a source and a target with a Source Qualifier transformation:



The source qualifier represents the rows that the Integration Service reads from the source when it runs a session.

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

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

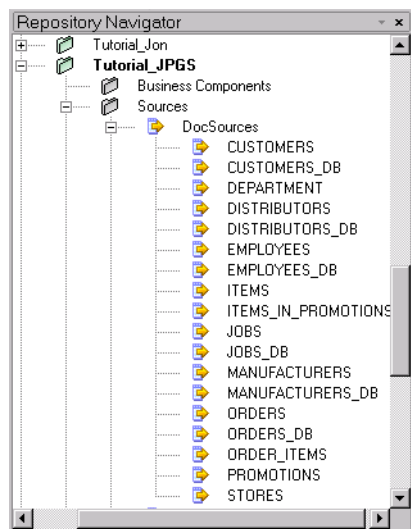
When you design mappings that contain different types of transformations, you can configure transformation ports as inputs, outputs, or both. You can rename ports and change the datatypes.

Creating a Mapping

In the following steps, you create a mapping and link columns in the source EMPLOYEES table to a Source Qualifier transformation.

1. Click **Tools > Mapping Designer** to open the Mapping Designer.
2. In the Navigator, expand the Sources node in the tutorial folder, and then expand the DBD node containing the tutorial sources.

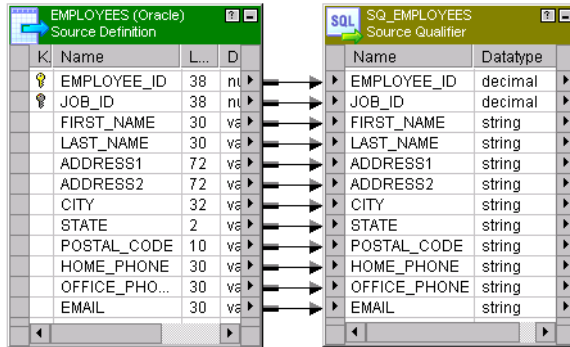
The following image shows the contents of the Sources node in the tutorial folder:



3. Drag the EMPLOYEES source definition into the Mapping Designer workspace.
The Designer creates a new mapping and prompts you to provide a name.
4. In the **Mapping Name** dialog box, enter m_PhoneList, and click **OK**.
The recommended naming convention for mappings is m_Mappingname.

The source definition appears in the workspace. The Designer creates a Source Qualifier transformation and connects it to the source definition.

The following image shows the EMPLOYEES source definition and the Source Qualifier transformation:



5. Expand the Targets node in the Navigator to open the list of all target definitions.
6. Drag the T_EMPLOYEES target definition into the workspace.
The target definition appears.
7. Click **Layout > Arrange**.
8. In the **Select Targets** dialog box, select T_EMPLOYEES target, and click **OK**.
The Designer rearranges the mapping.

The final step is to connect the Source Qualifier transformation to the target definition.

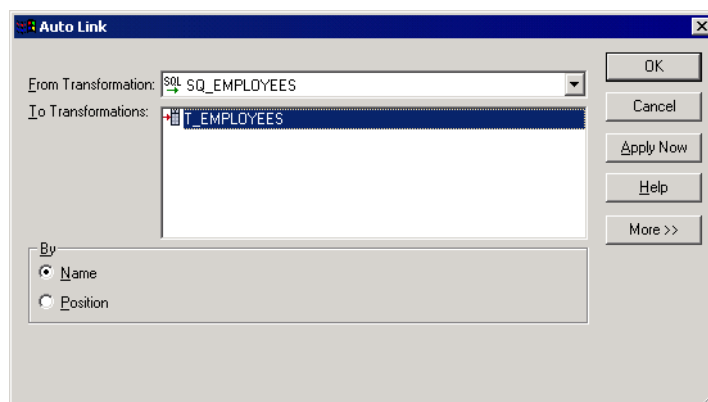
Connecting Transformations

The port names in the target definition are the same as some of the port names in the Source Qualifier transformation. When you need to link ports between transformations that have the same name, the Designer can link them based on name.

In the following steps, you use the autolink option to connect the Source Qualifier transformation to the target definition.

1. Click **Layout > Autolink**.
The **Auto Link** dialog box appears.

The following image shows the **Auto Link** dialog box:



2. Select T_EMPLOYEES in the To Transformations field. Verify that SQ_EMPLOYEES is in the From Transformation field.
3. Verify Name is selected, and click **OK**.

The Designer links ports from the Source Qualifier transformation to the target definition by name. A link appears between the ports in the Source Qualifier transformation and the target definition.

Note: To link ports with different names, you can drag from the port of one transformation to a port of another transformation or target. If you connect the wrong columns, select the link and press the Delete key.

4. Click **Layout > Arrange**.
5. In the **Select Targets** dialog box, select the T_EMPLOYEES target, and click **OK**.

The Designer rearranges the source definition, Source Qualifier transformation, and target definition from left to right, making it easy to see how one column maps to another.

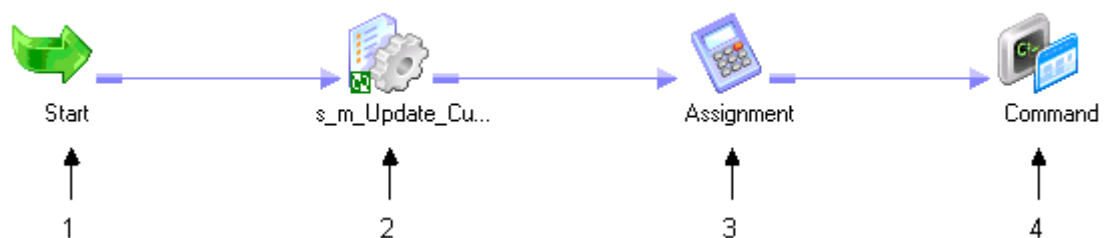
6. Optionally, to see all columns in the source definition, Source Qualifier transformation, or target definition window, drag the lower edge of the windows until all columns appear.
7. Click **Repository > Save** to save the new mapping to the repository.

Creating Sessions and Workflows

A session is a set of instructions that tells the Integration Service how to move data from sources to targets. A session is a task, similar to other tasks available in the Workflow Manager. You create a session for each mapping that you want the Integration Service to run. The Integration Service uses the instructions configured in the session and mapping to move data from sources to targets.

A workflow is a set of instructions that tells the Integration Service how to execute tasks, such as sessions, email notifications, and shell commands. You create a workflow for sessions that you want the Integration Service to run. You can include multiple sessions in a workflow to run sessions in parallel or sequentially. The Integration Service uses the instructions configured in the workflow to run sessions and other tasks.

The following figure shows a workflow with multiple branches and tasks:



1. Start task
2. Session task
3. Assignment task
4. Command task

You create and maintain tasks and workflows in the Workflow Manager.

In this lesson, you create a session and a workflow to run the session. Before you create a session in the Workflow Manager, you need to configure database connections in the Workflow Manager.

Configuring Database Connections in the Workflow Manager

Before you can create a session, you need to provide the Integration Service with the information it needs to connect to the source and target databases. Configure database connections in the Workflow Manager. Database connections are saved in the repository.

1. Launch the Workflow Manager.
2. In the Workflow Manager, select the repository in the Navigator, and then click **Repository > Connect**.
3. Enter a user name and password to connect to the repository and click **Connect**.
The native security domain is selected by default.
4. Click **Connections > Relational**.
The **Relational Connection Browser** dialog box appears.
5. Click **New** in the **Relational Connection Browser** dialog box.
The **Select Subtype** dialog box appears.
6. Select the appropriate database type and click **OK**.
The **Connection Object Definition** dialog box appears with options appropriate to the selected database platform.
7. In the Name field, enter TUTORIAL_SOURCE as the name of the database connection.
The Integration Service uses this name as a reference to this database connection.
8. Enter the user name and password to connect to the database.
9. Select a code page for the database connection.
The source code page must be a subset of the target code page.
10. In the Attributes section, enter the database name.
11. Enter additional information necessary to connect to this database, such as the connect string, and click **OK**.
Use the database connection information you created for the source database.
TUTORIAL_SOURCE now appears in the list of registered database connections in the **Relational Connection Browser** dialog box. TUTORIAL_TARGET appears in the list after step 12.
12. Repeat steps [5](#) through [11](#) to create another database connection called TUTORIAL_TARGET for the target database.
The target code page must be a superset of the source code page.
Use the database connection information you created for the target database.
13. Click **OK**.
TUTORIAL_SOURCE and TUTORIAL_TARGET appear in the list of registered database connections in the **Relational Connection Browser** dialog box.
14. Click **Close**.
You have finished configuring the connections to the source and target databases. The next step is to create a session for the mapping m_PhoneList.

Creating a Reusable Session

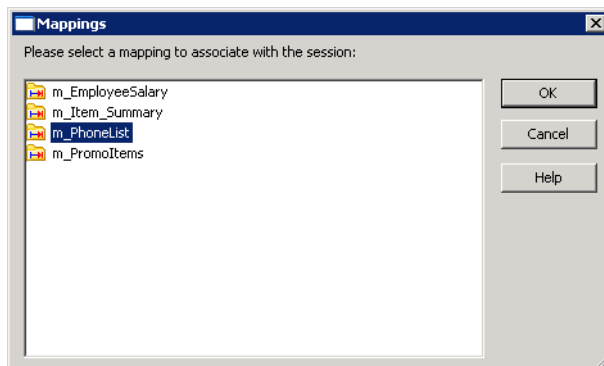
You can create reusable or non-reusable sessions in the Workflow Manager. Create reusable sessions in the Task Developer. When you create a reusable session, you can use it in multiple workflows. Create non-reusable sessions in the Workflow Designer. When you create a non-reusable session, you can use it only in that workflow.

In the following steps, you create a reusable session that uses the mapping m_PhoneList. Then, you create a workflow that uses the reusable session.

1. In the Workflow Manager Navigator, double-click the tutorial folder to open it.
2. Click **Tools > Task Developer** to open the Task Developer.
3. Click **Tasks > Create**.
4. Select Session as the task type to create.
5. Enter s_PhoneList as the session name and click Create.

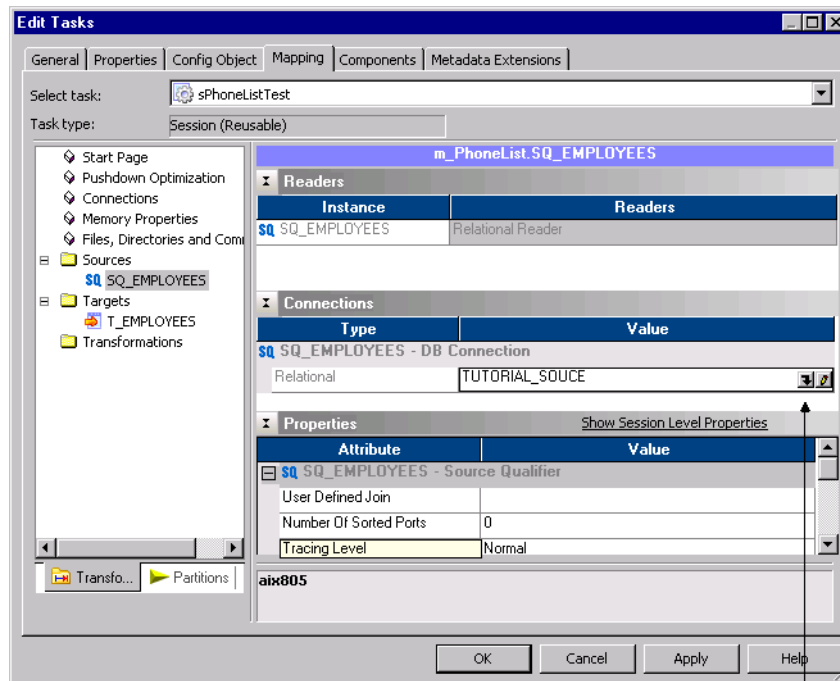
The **Mappings** dialog box appears.

The following image shows the **Mappings** dialog box:



6. Select the mapping m_PhoneList and click **OK**.
The Workflow Manager creates a reusable Session task in the Task Developer workspace.
7. Click **Done** in the **Create Task** dialog box.
8. In the workspace, double-click s_PhoneList to open the session properties.
9. On the **Mapping** tab, select Sources in the Transformations pane on the left.

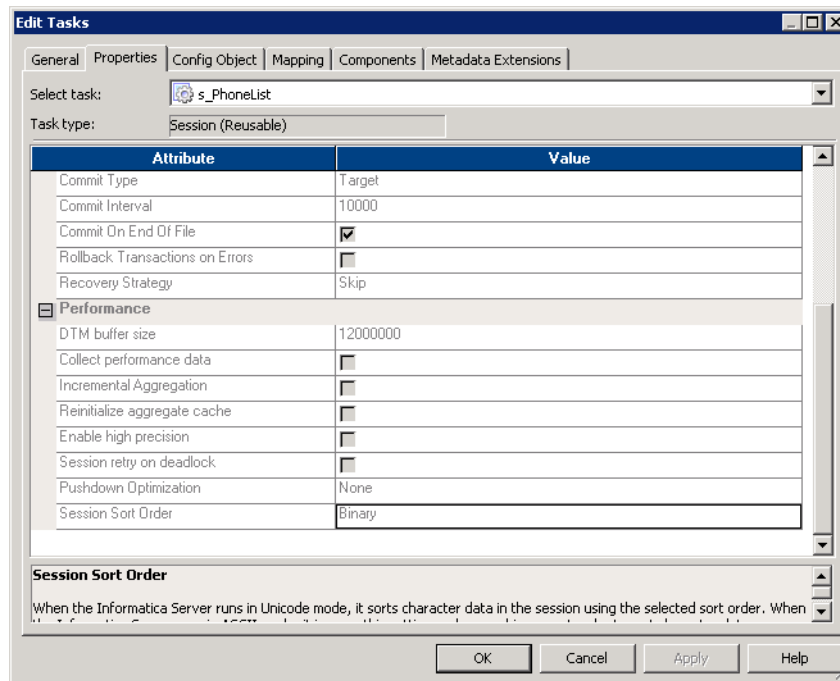
The following image shows the **Mapping** tab for the session s_PhoneList:



1

1. Open button
10. In the Connections settings on the right, click the **Open** button in the Value column for the SQ_EMPLOYEES - DB Connection.
The Relational Connection Browser appears.
11. Select TUTORIAL_SOURCE and click **OK**.
12. Select Targets in the Transformations pane.
13. In the Connections settings, click the **Open** button in the Value column for the T_EMPLOYEES - DB Connection.
The Relational Connection Browser appears.
14. Select TUTORIAL_TARGET and click **OK**.
15. In the Properties settings, change the Target Load Type to Normal.
16. On the **Properties** tab, select a session sort order associated with the Integration Service code page.
For English data, use the Binary sort order.

The following figure shows the Properties tab where you can configure the Binary sort order:



17. Click **OK** to save and close the session properties.

In this step, you use most of the default settings.

18. Click **Repository > Save** to save the session to the repository.

You have created a reusable session. The next step is to create a workflow that runs the session.

Creating a Workflow

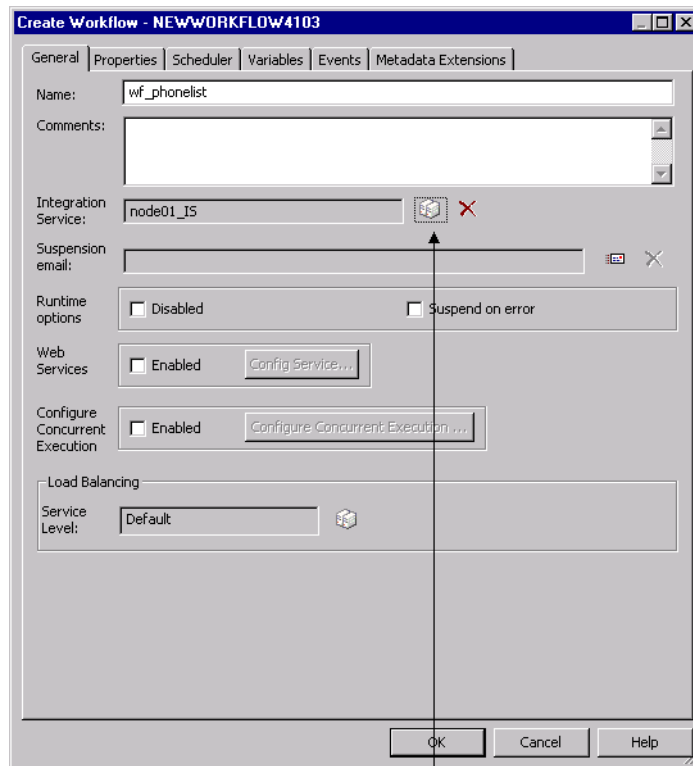
You create workflows in the Workflow Designer. When you create a workflow, you can include reusable tasks that you create in the Task Developer. You can also include non-reusable tasks that you create in the Workflow Designer.

In the following steps, you create a workflow that runs the session s_PhoneList.

1. Click **Tools > Workflow Designer**.
2. In the Navigator, expand the tutorial folder, and expand the Sessions node.
3. Drag the session s_PhoneList to the Workflow Designer workspace.

The **Create Workflow** dialog box appears.

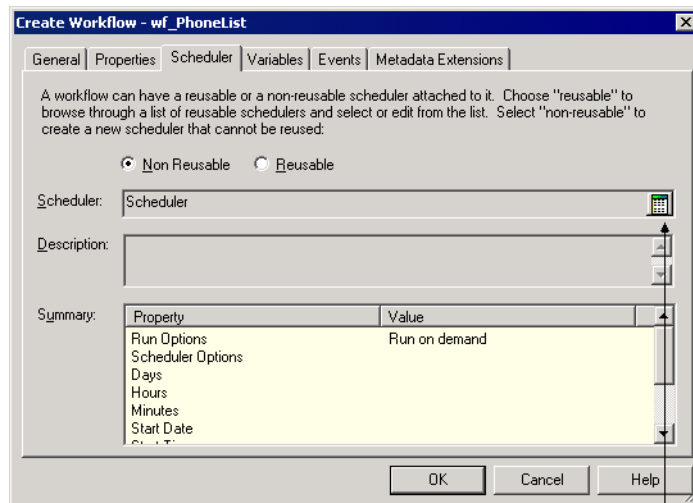
The following image shows **Create Workflow** dialog box:



1

1. Open button
4. Enter wf_PhoneList as the name for the workflow.
The recommended naming convention for workflows is wf_WorkflowName.
5. Click the **Open** button to choose an Integration Service to run the workflow.
The **Integration Service Browser** dialog box appears.
6. Select the appropriate Integration Service and click **OK**.
Note: Integration Services appear if they are running. If no Integration Service appears, contact an Informatica administrator.
7. Click the **Properties** tab to edit the workflow properties.
8. Enter wf_PhoneList.log for the workflow log file name.
9. Click the **Scheduler** tab.

The following image shows the contents of **Scheduler** tab:



1

1. Edit Scheduler button

10. Click **OK** to accept the default schedule for this workflow.

The workflow schedule runs on demand by default. The Integration Service runs the workflow when you manually start the workflow. You can configure workflows to run on a schedule. For example, you can schedule a workflow to run once a day or run on the last day of the month. Optionally, click the **Edit Scheduler** button to configure schedule options.

11. Click **OK** to close the **Create Workflow** dialog box.

The Workflow Manager creates a workflow in the workspace, including the reusable session you added. All workflows begin with the Start task, but you need to instruct the Integration Service which task to run next. To do this, you link tasks in the Workflow Manager.

Note: You can click **Workflows > Edit** to edit the workflow properties at any time.

12. Click **Tasks > Link Task**.

In this lesson, the Workflow Manager creates the link for you.

13. Drag from the Start task to the Session task.

The following image shows the link from Start task to the Session task:



14. Click **Repository > Save** to save the workflow in the repository.

You can run and monitor the workflow.

Running and Monitoring Workflows

When the PowerCenter Integration Service runs workflows, you can monitor workflow progress in the Workflow Monitor. You can view details about a workflow or task in either a Gantt Chart view or a Task view.

You can start, stop, and abort workflows from the Workflow Monitor. The Workflow Monitor displays workflows that have run at least once.

In the following steps, you run a workflow and monitor it.

Opening the Workflow Monitor

You can configure the Workflow Manager to open the Workflow Monitor when you run a workflow from the Workflow Manager.

You can also open the Workflow Monitor from the Workflow Manager Navigator or from the Windows Start menu.

1. In the Workflow Manager, click **Tools > Options**.
The **Options** dialog box appears.
2. On the **General** tab, view the **Launch Workflow Monitor When Workflow Is Started** option.
This option is selected by default.
3. Click **OK**.

Next, you run the workflow and open the Workflow Monitor.

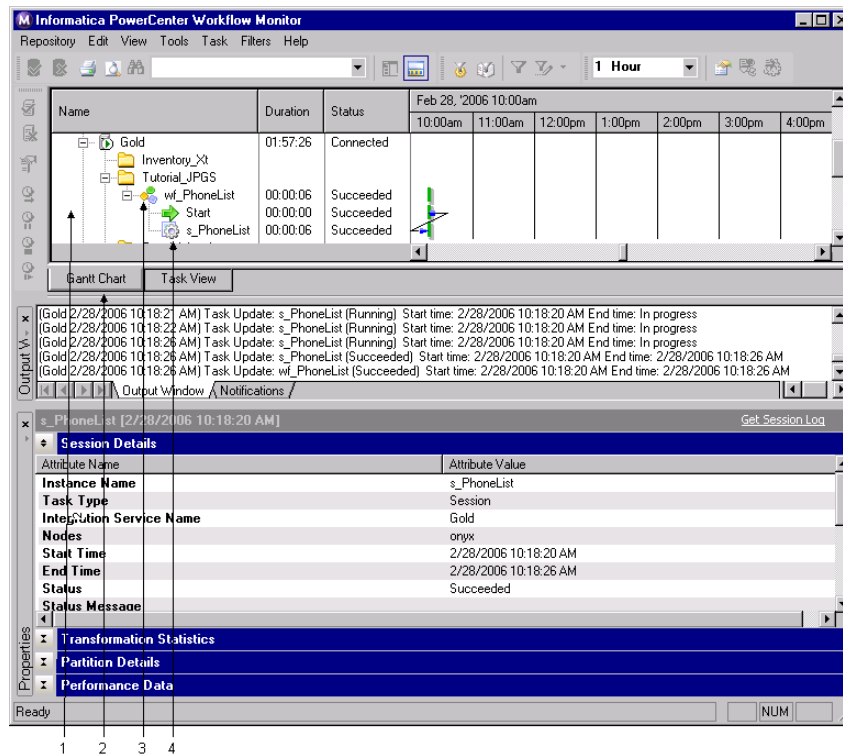
Running the Workflow

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

1. Verify the workflow is open in the Workflow Designer.
2. In the Workflow Manager, click **Workflows > Start Workflow**.
Tip: You can also right-click the workflow in the Navigator and select **Start Workflow**.

The Workflow Monitor opens, connects to the repository, and opens the Tutorial folder.

The following image shows the Workflow monitor:



1. Navigator
 2. Gantt Chart tab
 3. Workflow
 4. Session
3. Click the **Gantt Chart** tab at the bottom of the Time window to verify the Workflow Monitor is in the Gantt Chart view.
 4. In the Navigator, expand the node for the workflow.
All tasks in the workflow appear in the Navigator.

The following table shows the data session writes to the target:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	OFFICE_PHONE
1921	William	Nelson	415-541-5145
1922	Ian	Page	415-541-5145
1923	Lyle	Osborne	415-541-5145
1928	Leo	DeSouza	415-541-5145
2001	Ira	S.MacDonald	415-541-5145
2002	Andy	Hill	415-541-5145
2003	Monisha	Sawyer	415-541-5145
2006	Bender	St.Jean	415-541-5145
2100	Teddy	Johnson	415-541-5145
2102	Ono	Steadman	415-541-5145
2103	John	Markowitz	415-541-5145
2109	Tom	Centre	415-541-5145

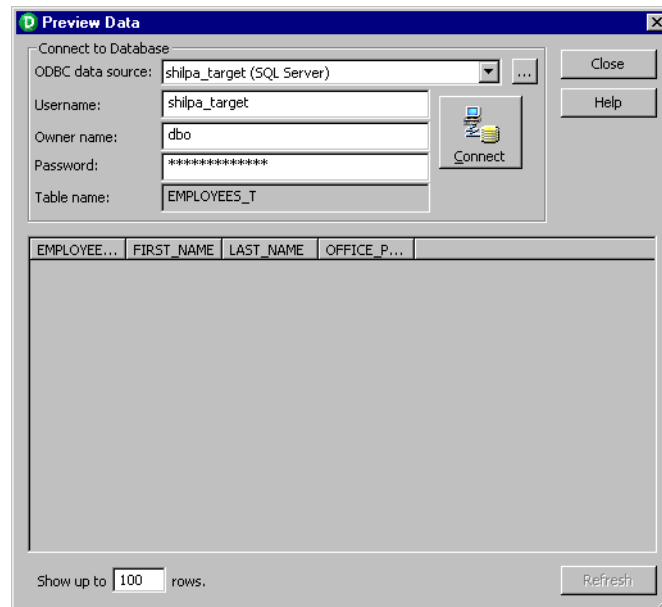
Previewing Data

You can preview the data that the PowerCenter Integration Service loaded in the target.

1. Open the Designer.
2. Click on the **Mapping Designer** button.
3. In the mapping m_PhoneList, right-click the target definition, T_EMPLOYEES and click **Preview Data**.

The **Preview Data** dialog box appears.

The following image shows **Preview Data** dialog box:



4. In the ODBC data source field, select the data source name that you used to create the target table.
5. Enter the database username, owner name and password.
6. Enter the number of rows you want to preview.
7. Click **Connect**.
The **Preview Data** dialog box displays the data that you loaded to T_EMPLOYEES.
8. Click **Close**.

You can preview relational tables, fixed-width and delimited flat files, and XML files with the Preview Data option.

CHAPTER 6

Tutorial Lesson 4

This chapter includes the following topics:

- [Using Transformations, 59](#)
- [Creating a Target Definition and Target, 60](#)
- [Creating a Mapping with Aggregate Values, 63](#)
- [Designer Tips, 70](#)
- [Creating a Session and Workflow, 71](#)

Using Transformations

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

A transformation is a part of a mapping that generates or modifies data. Every mapping includes a Source Qualifier transformation, representing all data read from a source and temporarily stored by the Integration Service. In addition, you can add transformations that calculate a sum, look up a value, or generate a unique ID before the source data reaches the target.

The following table lists some of the transformations that you can create:

Transformation	Description
Aggregator	Performs aggregate calculations.
Expression	Calculates a value.
Filter	Filters data.
Joiner	Joins data from different databases or flat file systems.
Lookup	Looks up and returns values from a relational table or a file.
Rank	Limits records to a top or bottom range.
Router	Routes data into multiple transformations based on group conditions.
Sequence Generator	Generates primary keys.
Sorter	Sorts data based on a sort key.

Transformation	Description
Source Qualifier	Represents the rows that the Integration Service reads from a relational or flat file source when it runs a workflow.
Stored Procedure	Calls a stored procedure.
Union	Merges data from multiple databases or flat file systems.
Update Strategy	Determines whether to insert, delete, update, or reject records.

In this lesson, you complete the following tasks:

1. Create a target definition to use in a mapping and create a target table based on the new target definition.
2. Create a mapping using the target definition. Add the following transformations to the mapping:
 - Lookup transformation. Finds the name of a manufacturer.
 - Aggregator transformation. Calculates the maximum, minimum, and average price of items from each manufacturer.
 - Expression transformation. Calculates the average profit of items, based on the average price.
3. Learn some tips for using the Designer.
4. Create a session and workflow to run the mapping and monitor the workflow in the Workflow Monitor.

Creating a Target Definition and Target

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

After you create the target definition, you create the table in the target database.

Creating a Target Definition

To create the target definition in this lesson, you copy the MANUFACTURERS source definition into the Target Designer. Then, you modify the target definition by adding columns to create the definition you want.

Note: You can also manually create a target definition, import the definition for an existing target from a database, or create a relational target from a transformation in the Designer.

1. Open the Designer, connect to the repository, and open the tutorial folder.
2. Click **Tools > Target Designer**.
3. Drag the MANUFACTURERS source definition from the Navigator to the Target Designer workspace.

The Designer creates a target definition, MANUFACTURERS, with the same column definitions as the MANUFACTURERS source definition and the same database type.

Next, you add target column definitions.
4. Double-click the MANUFACTURERS target definition to open it.

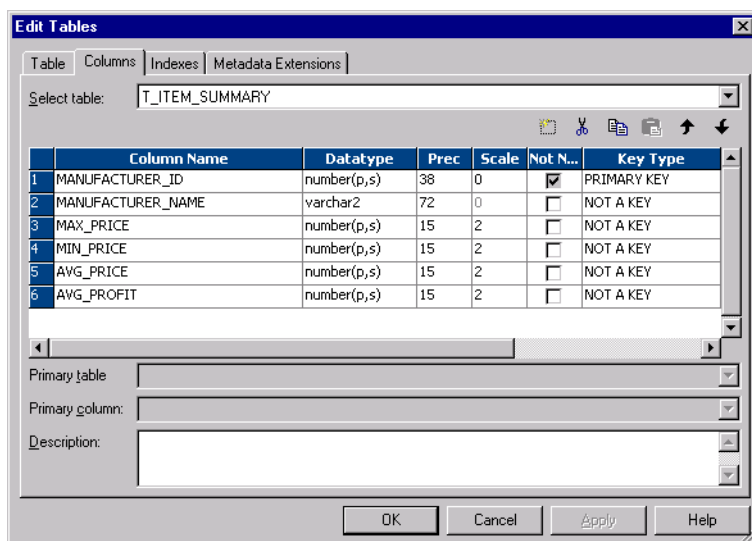
The **Edit Tables** dialog box appears.

5. Click **Rename** and name the target definition T_ITEM_SUMMARY.
6. Optionally, change the database type for the target definition. You can select the correct database type when you edit the target definition.
7. Click the **Columns** tab.
The target column definitions are the same as the MANUFACTURERS source definition.
8. For the MANUFACTURER_NAME column, change precision to 72 and clear the Not Null column.
9. Add the following columns with the Money datatype, default precision, and default scale:
 - MAX_PRICE
 - MIN_PRICE
 - AVG_PRICE
 - AVG_PROFIT

If the Money datatype does not exist in the database, use Number (p,s) or Decimal and change the precision to 15 and the scale to 2.

10. Click **Apply**.

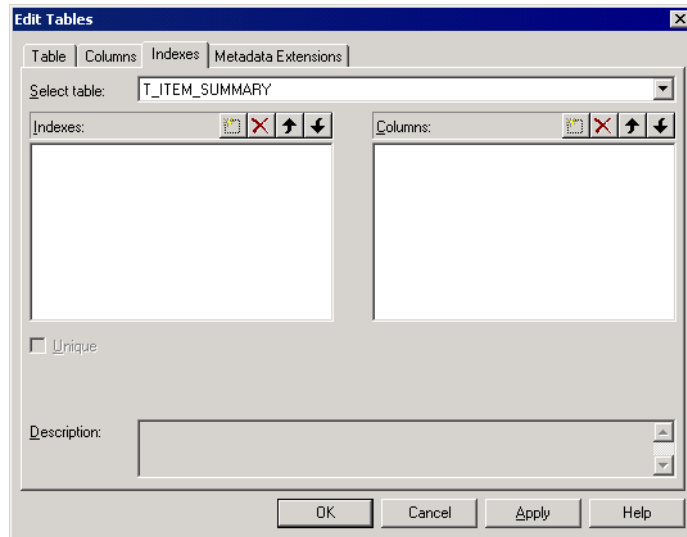
The following image shows the **Columns** tab for the target definition:



11. Click the **Indexes** tab to add an index to the target table.

If the target database is Oracle, skip to the final step. You cannot add an index to a column that already has the PRIMARY KEY constraint added to it.

The following image shows the **Indexes** tab for the target definition:



12. In the Indexes section, click the **Insert** button.
13. Enter **IDX_MANUFACTURER_ID** as the name of the index, and then press Enter.
14. Select the **Unique** index option.
15. In the Columns section, click **Add**.
The **Add Column To Index** dialog box lists the columns you added to the target definition.
16. Select **MANUFACTURER_ID** and click **OK**.
17. Click **OK** to save the changes to the target definition, and then click **Repository > Save**.

Creating a Target Table

In the following steps, you use the Designer to generate and execute the SQL script to create a target table based on the target definition you created.

1. Select the table **T_ITEM_SUMMARY**, and then click **Targets > Generate/Execute SQL**.
2. In the **Database Object Generation** dialog box, connect to the target database.
3. Click **Generate from Selected tables**, and select the **Create Table, Primary Key, and Create Index** options.
Leave the other options unchanged.
4. Click **Generate and execute**.
The Designer notifies you that the file **MKTABLES.SQL** already exists.
5. Click **OK** to override the contents of the file and create the target table.
The Designer runs the SQL script to create the **T_ITEM_SUMMARY** table.
6. Click **Close**.

Creating a Mapping with Aggregate Values

In the next step, you create a mapping with the following mapping logic:

- Finds the most expensive and least expensive item in the inventory for each manufacturer. Use an Aggregator transformation to perform these calculations.
- Calculates the average price and profitability of all items from a given manufacturer. Use an Aggregator and an Expression transformation to perform these calculations.

You need to configure the mapping to perform both simple and aggregate calculations. For example, use the MIN and MAX functions to find the most and least expensive items from each manufacturer.

Creating a Mapping with T_ITEM_SUMMARY

First, create a mapping with the target definition you just created.

1. Switch from the Target Designer to the Mapping Designer.
2. Click **Mappings > Create**.
3. If prompted to close the current mapping, click **Yes**.
4. In the **Mapping Name** dialog box, enter `m_ItemSummary` as the name of the mapping.
5. From the list of sources in the tutorial folder, drag the `ITEMS` source definition into the mapping.
6. From the list of targets in the tutorial folder, drag the `T_ITEM_SUMMARY` target definition into the mapping.

Creating an Aggregator Transformation

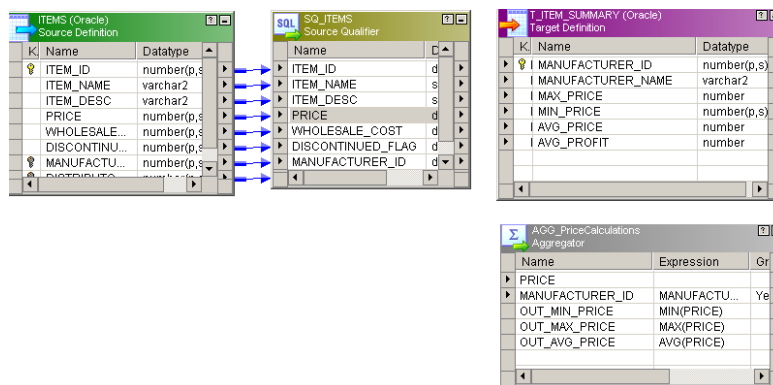
Next, add an Aggregator transformation to calculate the average, maximum, and minimum prices of items from each manufacturer.

1. Click **Transformation > Create** to create an Aggregator transformation.
2. Select **Aggregator** in the **Select the transformation type to create** dialog box and type `AGG_PriceCalculations` in the **Enter a new name for this transformation** dialog box. Click **Create**, and then click **Done**.

The naming convention for Aggregator transformations is `AGG_TransformationName`.

The Mapping Designer adds an Aggregator transformation to the mapping.

The following image shows a mapping with an unconnected Aggregator transformation:



3. Click **Layout > Link Columns**.

When you drag ports from one transformation to another, the Designer copies the port description and links the original port to its copy.

If you click **Layout > Copy Columns**, every port you drag is copied, but not linked.

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

A copy of the PRICE port now appears in the new Aggregator transformation. The new port has the same name and datatype as the port in the Source Qualifier transformation.

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

5. Drag the MANUFACTURER_ID port into the Aggregator transformation.

You need another input port, MANUFACTURER_ID, to provide the information for the equivalent of a GROUP BY statement. By adding this second input port, you can define the groups for the aggregate calculation. You will create a group to organize the data by manufacturer.

6. Double-click the Aggregator transformation, and then click the **Ports** tab.

7. Clear the **Output (O)** column for PRICE.

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

8. Select the **Group By** option for the MANUFACTURER_ID column.

9. Click the **Add** button three times to add three new ports.

When you select the **Group By** option for MANUFACTURER_ID, the Integration Service groups all incoming rows by manufacturer ID when it runs the session.

10. Configure the output ports.

The following table describes the output ports:

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

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

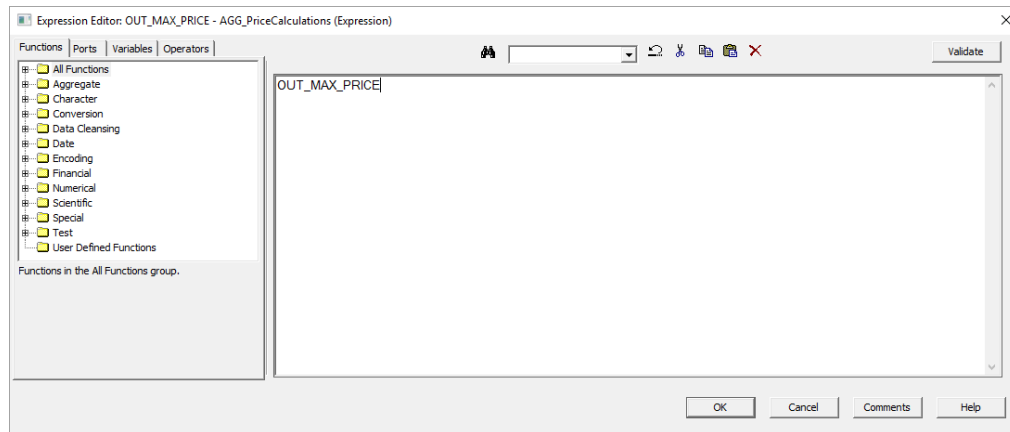
11. Click **Apply** to save the changes.

Entering an Aggregate Calculation

Now, you need to enter the expressions for all three output ports, using the functions MAX, MIN, and AVG to perform aggregate calculations.

1. Click the **Open** button in the Expression column of the OUT_MAX_PRICE port to open the **Expression Editor**.
2. Delete the text OUT_MAX_PRICE.

The following image shows the **Expression Editor** window:



The Formula section of the **Expression Editor** displays the expression as you develop it. Use other sections of this dialog box to select the input ports to provide values for an expression, enter literals and operators, and select functions to use in the expression.

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

A list of all aggregate functions now appears.

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

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

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

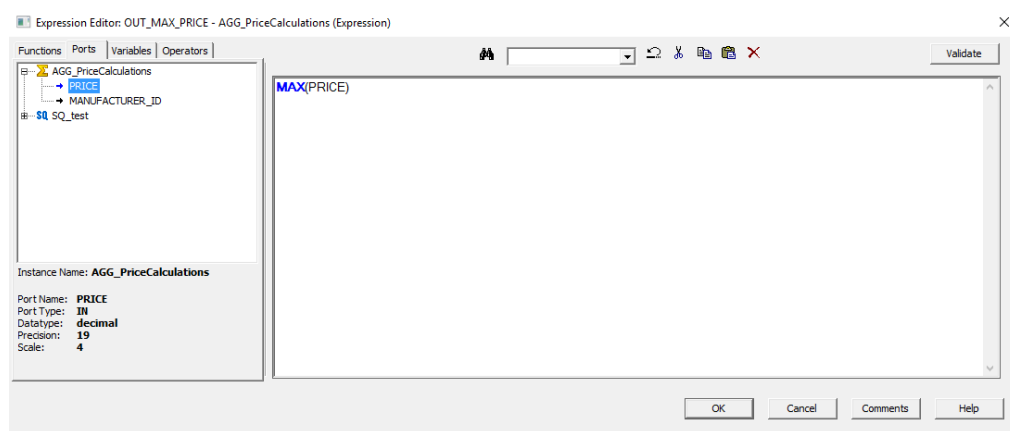
6. Click the **Ports** tab.

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

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

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

The following image shows the **Ports** tab in the **Expression Editor**:



8. Click **Validate**.

The Designer displays a message indicating that the expression parsed successfully.

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

Entering Remaining Aggregate Calculations

Next, enter the remaining aggregate calculations.

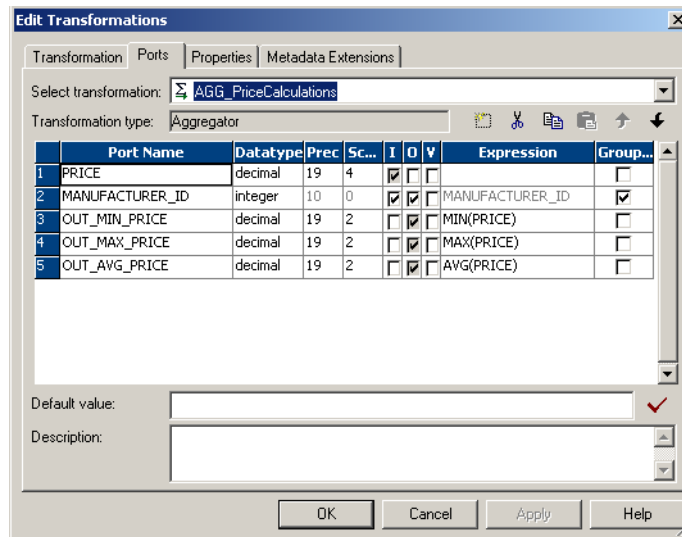
1. Enter and validate the expressions for the other output ports.

The following table describes the output ports and expressions:

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

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

The following image shows the ports configured for the Aggregator transformation:



2. Click **OK** to close the **Edit Transformations** dialog box.
3. Click **Repository > Save**.

When you save changes to the repository, the Designer validates the mapping. You can notice an error message indicating that you have not connected the targets. You connect the targets later in this lesson.

Creating an Expression Transformation

Now that you have calculated the highest, lowest, and average prices for items, the next step is to calculate the average profitability of items from each manufacturer.

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

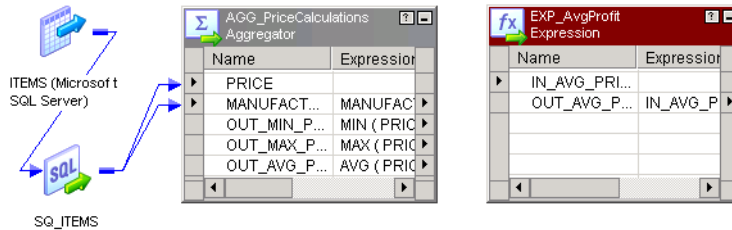
1. Click **Transformation > Create**.
2. Select **Expression** and name the transformation EXP_AvgProfit. Click **Create**, and then click **Done**.

The naming convention for Expression transformations is *EXP_TransformationName*.

The Mapping Designer adds an Expression transformation to the mapping.

The following mapping shows an unconnected Expression transformation:

Mapping Designer



Note: Right-click on the source definition and source qualifier and select **Iconize** to iconize them.

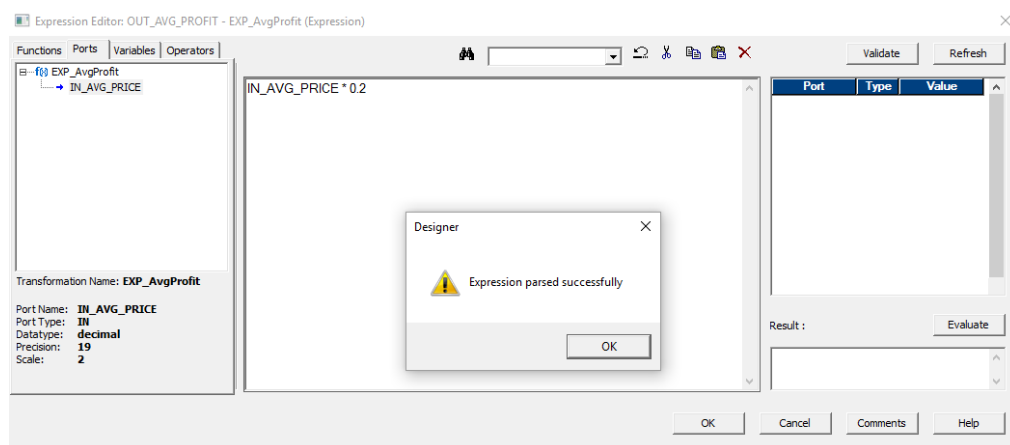
3. Open the Expression transformation.
4. Add an input port, IN_AVG_PRICE, using the Decimal datatype with precision of 19 and scale of 2.
5. Add an output port, OUT_AVG_PROFIT, using the Decimal datatype with precision of 19 and scale of 2. Clear the Input port.

Note: OUT_AVG_PROFIT is an output port, not an input/output port. You cannot enter expressions in input/output ports.

6. Enter the following expression for OUT_AVG_PROFIT:

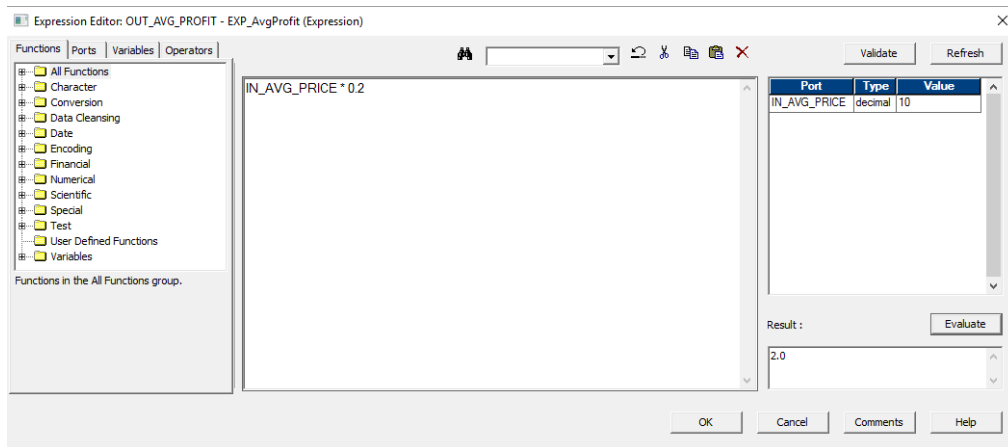
IN_AVG_PRICE * 0.2

7. To validate the expression, click Validate.



8. To reflect the latest changes from the expression condition in the test expression section in the right pane, click Refresh.

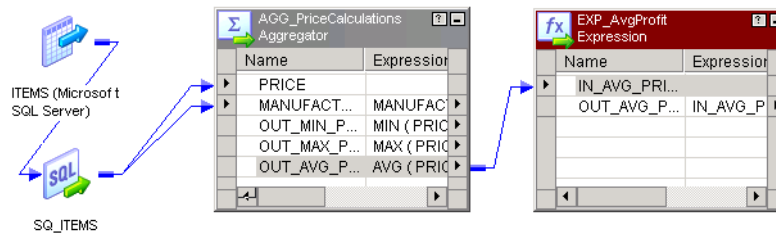
- In the right pane, enter sample values for the input ports used within the expression, and click Evaluate. The following image shows the sample value and the result in the right pane:



- Close the **Expression Editor** and then close the EXP_AvgProfit transformation.
- Connect OUT_AVG_PRICE from the Aggregator to IN_AVG_PRICE input port.

The following mapping shows a link from the Aggregator transformation port OUT_AVG_PRICE to the input port IN_AVG_PRICE:

Mapping Designer



- Connect the OUT_AVG_PROFIT port in the Expression Transformation to the AVG_PROFIT port in the T_ITEM_SUMMARY target definition.
- Click **Repository > Save**.

Creating a Lookup Transformation

The source table in this mapping includes information about the manufacturer ID. However, you want the manufacturer name in the target table to make the summary data easier to read. In the following steps, you use a Lookup transformation to find each manufacturer name in the MANUFACTURERS table based on the manufacturer ID in the source table.

- Create a Lookup transformation and name it LKP_Manufacturers.

The naming convention for Lookup transformations is *LKP_TransformationName*.

A dialog box prompts you to identify the source or target database to provide data for the lookup. When you run a session, the Integration Service must access the lookup table.

- Click **Source**.
- Select the MANUFACTURERS table from the list and click **OK**.
- Click **Done** to close the **Create Transformation** dialog box.

The Designer adds the transformation.

Use source and target definitions in the repository to identify a lookup source for the Lookup transformation. Alternatively, you can import a lookup source.

5. Open the Lookup transformation.

6. Add an input port, IN_MANUFACTURER_ID, with the same datatype as MANUFACTURER_ID.

In a later step, you connect the MANUFACTURER_ID port from the Aggregator transformation to this input port. IN_MANUFACTURER_ID receives MANUFACTURER_ID values from the Aggregator transformation. When the Lookup transformation receives a new value through this input port, it looks up the matching value from MANUFACTURERS.

Note: By default, the Lookup transformation queries and stores the contents of the lookup table before the rest of the transformation runs, so it performs the join through a local copy of the table that it has cached.

7. Click the **Condition** tab, and click the **Add** button.

An entry for the first condition in the lookup appears. Each row represents one condition in the WHERE clause that the Integration Service generates when querying records.

8. Verify the settings for the condition.

The following table describes the settings for the condition:

Lookup Table Column	Operator	Transformation Port
MANUFACTURER_ID	=	IN_MANUFACTURER_ID

Note: If the datatypes, including precision and scale, of these two columns do not match, the Designer displays a message and marks the mapping invalid.

9. View the **Properties** tab.

Do not change settings in this section of the dialog box.

10. Click **OK**.

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

11. Click **Layout > Link Columns**.

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

13. Connect the MANUFACTURER_ID port in the LKP_MANUFACTURERS Lookup transformation object to the MANUFACTURER_ID port in the T_ITEM_SUMMARY target definition object.

14. Click **Repository > Save**.

Connecting the Target

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

- Created a target definition and target table.
- Created a mapping.
- Added transformations.

The final step is to connect to the target.

1. Drag the output ports to the corresponding input ports in the target.

The following table describes the output ports that connect to the corresponding input ports:

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

2. Click **Repository > Save**.
Verify mapping validation in the Output window.

Designer Tips

This section includes tips for using the Designer. You learn how to complete the following tasks:

- Use the Overview window to navigate the workspace.
- Arrange the transformations in the workspace.

Using the Overview Window

When you create a mapping with many transformations, you might not be able to see the entire mapping in the workspace. In the following steps, you use the Overview window to navigate around the workspace containing the mapping you just created.

1. Click **View > Overview Window**.

You can also use the **Toggle Overview Window** icon.



An overview window appears, displaying a smaller version of the mapping.

2. Drag the viewing rectangle within this window.

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

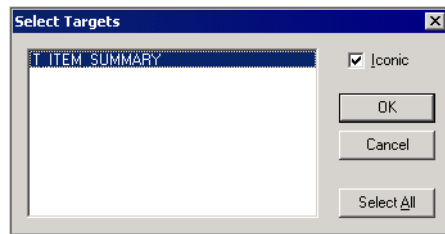
Arranging Transformations

The Designer can arrange the transformations in a mapping. When you use this option to arrange the mapping, you can arrange the transformations in normal view, or as icons.

1. Click **Layout > Arrange**.

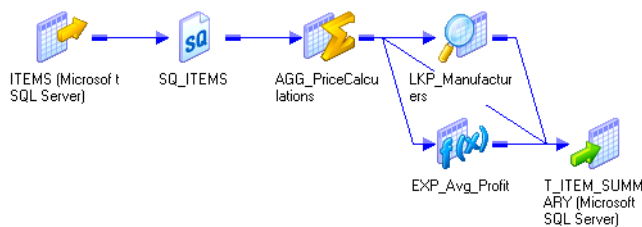
The **Select Targets** dialog box appears showing all target definitions in the mapping.

The following image shows the **Select Targets** dialog box:



2. Select **Iconic** to arrange the transformations as icons in the workspace.
3. Select **T_ITEM_SUMMARY** and click **OK**.

The following mapping shows an iconic view of the transformations in the pipeline connected to the **T_ITEM_SUMMARY** target definition:



Creating a Session and Workflow

You have two mappings:

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

You have a reusable session based on **m_PhoneList**. Next, you create a session for **m_ItemSummary** in the Workflow Manager. You create a workflow that runs both sessions.

Creating the Session

Open the Workflow Manager and connect to the repository if it is not open already.

1. Open the Task Developer and click **Tasks > Create**.
2. Create a Session task and name it **s_ItemSummary**. Click **Create**.

In the **Mappings** dialog box, select the mapping **m_ItemSummary** and click **OK**.

3. Click **Done**.
4. Open the session properties for s_ItemSummary by double clicking the icon in the workspace.
The **Edit Tasks** dialog box appears.
5. Click the Connections setting on the **Mapping** tab. Select the source database connection TUTORIAL_SOURCE for SQ_ITEMS.
Use the database connection you created in [“Configuring Database Connections in the Workflow Manager” on page 50](#).
6. Select the target database connection TUTORIAL_TARGET for T_ITEM_SUMMARY.
Use the database connection you created in [“Configuring Database Connections in the Workflow Manager” on page 50](#).
7. Click **OK** to close the **Edit Tasks** dialog box.
8. Click **Repository > Save**.

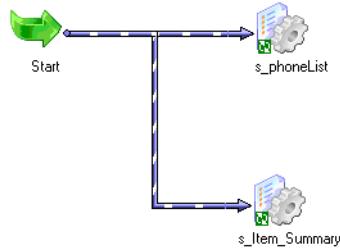
Now that you have two sessions, you can create a workflow and include both sessions in the workflow. When you run the workflow, the Integration Service runs all sessions in the workflow, either simultaneously or in sequence, depending on how you arrange the sessions in the workflow.

Creating the Workflow

You can group sessions in a workflow to improve performance or ensure that targets load in a set order. In the following steps, you create a workflow that runs the sessions s_PhoneList and s_ItemSummary concurrently.

1. Click **Tools > Workflow Designer**.
2. Click **Workflows > Create** to create a new workflow.
If a workflow is already open, the workspace is cleared.
The workflow properties appear.
3. Name the workflow wf_ItemSummary_PhoneList.
4. Click the **Open** button to select an Integration Service to run the workflow.
The **Integration Service Browser** dialog box appears.
5. Select an Integration Service and click **OK**.
6. Click the **Properties** tab and select **Write Backward Compatible Workflow Log File**.
The default name of the workflow log file is wf_ItemSummary_PhoneList.log.
7. Click the **Scheduler** tab.
By default, the workflow is scheduled to run on demand. Keep this default.
8. Click **OK** to close the **Create Workflow** dialog box.
The Workflow Manager creates a workflow in the workspace including the Start task.
9. From the Navigator, drag the s_ItemSummary session to the workspace. Then, drag the s_PhoneList session to the workspace.
10. Click the **Link Task** button on the toolbar.
11. Drag from the Start task to the s_ItemSummary Session task.
12. Drag from the Start task to the s_PhoneList Session task.

The following image shows the workflow wf_ItemSummary_PhoneList:



By default, when you link both sessions directly to the Start task, the Integration Service runs both sessions at the same time when you run the workflow. If you want the Integration Service to run the sessions one after the other, connect the Start task to one session, and connect that session to the other session.

13. Click **Repository > Save** to save the workflow in the repository.
You can now run and monitor the workflow.

Running the Workflow

After you create the workflow containing the sessions, you can run it and use the Workflow Monitor to monitor the workflow progress.

1. Right-click the Start task in the workspace and select **Start Workflow**.
Tip: You can also right-click the workflow in the Navigator and select **Start Workflow**.
The Workflow Monitor opens and connects to the repository and opens the tutorial folder.
If the Workflow Monitor does not show the current workflow tasks, right-click the tutorial folder and select **Get Previous Runs**.
2. Click the **Gantt Chart** tab at the bottom of the **Time** window to verify the Workflow Monitor is in Gantt Chart view.
Note: You can also click the **Task View** tab at the bottom of the Time window to view the Workflow Monitor in Task view. You can switch back and forth between views at any time.
3. In the Navigator, expand the node for the workflow.

All tasks in the workflow appear in the Navigator.

You can preview the results from the workflow by following the directions on [“Previewing Data” on page 58](#).

The following table shows the results from running the session s_ItemSummary:

MANUFACTURER_ID	MANUFACTURER_NAME	MAX_PRICE	MIN_PRICE	AVG_PRICE	AVG_PROFIT
100	Nike	365.00	169.95	261.24	52.25
101	OBrien	188.00	44.95	134.32	26.86
102	Mistral	390.00	70.00	200	40.00
103	Spinnaker	70.00	29.00	52.98	10.60
104	Head	179.00	52.00	98.67	19.73
105	Jesper	325.00	34.95	133.65	26.73
106	Acme	195.00	56.95	143.65	28.73
107	Medallion	235.00	19.95	98.65	19.73
108	Sportstar	280.00	18.00	149.00	29.80

MANUFACTURER_ID	MANUFACTURER_NAME	MAX_PRICE	MIN_PRICE	AVG_PRICE	AVG_PROFIT
109	WindJammer	430.00	395.00	412.50	82.50
110	Monsoon	280.00	280.00	280.00	56.00

Viewing the Logs

You can view workflow and session logs after you run the workflow. You can see detailed information about each event performed during the workflow run.

1. Choose to view log events for the workflow or session.
 - To view the session log, click the session and select **Get Session Log**.
 - To view the workflow log, click the workflow and select **Get Workflow Log**.
2. Select a row in the log.

The full text of the message appears in the section at the bottom of the window.
3. Sort the log file by column by clicking on the column heading.
4. Optionally, click **Find** to search for keywords in the log.
5. Optionally, click **Save As** to save the log as an XML document.

Log Files

When you created the workflow, the Workflow Manager assigned default workflow and session log names and locations on the **Properties** tab. The Integration Service writes the log files to the locations specified in the session properties.

CHAPTER 7

Tutorial Lesson 5

This chapter includes the following topics:

- [Creating a Mapping with Fact and Dimension Tables, 75](#)
- [Creating a Workflow, 83](#)

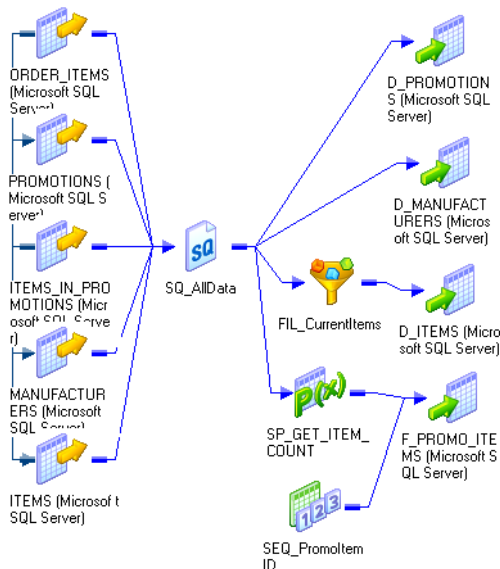
Creating a Mapping with Fact and Dimension Tables

In previous lessons, you used the Source Qualifier, Expression, Aggregator, and Lookup transformations in mappings. In this lesson, you learn how to use the following transformations:

- Stored Procedure. Call a stored procedure and capture its return values.
- Filter. Filter data that you do not need, such as discontinued items in the ITEMS table.
- Sequence Generator. Generate unique IDs before inserting rows into the target.

You create a mapping that outputs data to a fact table and its dimension tables.

The following figure shows the mapping you create in this lesson:



Creating Targets

Before you create a mapping, you need to create the target tables for the mapping. These instructions explain how to create the following target tables:

- F_PROMO_ITEMS. A fact table of promotional items.
 - D_ITEMS, D_PROMOTIONS, and D_MANUFACTURERS. Dimensional tables.
1. Open the Designer, connect to the repository, and open the tutorial folder.
 2. Click **Tools > Target Designer**.
To clear the workspace, right-click the workspace, and select **Clear All**.
 3. Click **Targets > Create**.
 4. In the **Create Target Table** dialog box, enter F_PROMO_ITEMS as the name of the target table, select the database type, and click **Create**.
 5. Repeat step 4 to create the other tables needed for this schema: D_ITEMS, D_PROMOTIONS, and D_MANUFACTURERS. When you have created all these tables, click **Done**.
 6. Open each target definition, and add the columns to the appropriate table.

The following table describes the columns for the table **D_ITEMS**:

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

The following table describes the columns for the table **D_PROMOTIONS**:

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

The following table describes the columns for the table **D_MANUFACTURERS**:

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

The following table describes the columns for the table **F_PROMO_ITEMS**:

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

The datatypes may vary, depending on the database you choose.

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

7. Click **OK**.
8. Click **Repository > Save**.

Creating Target Tables

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

1. Select all of the target definitions.
2. Click **Targets > Generate/Execute SQL**.
3. In the **Database Object Generation** dialog box, **connect** to the target database.
4. Select generate from **Selected tables**, and select the options for creating the tables and generating keys.
5. Click **Generate and Execute**.
6. Click **Close**.

Creating the Mapping

Create a mapping to filter out discontinued items and generate a unique ID for each row in the fact table. The mapping also finds the number of items that each customer ordered.

1. In the Designer, switch to the Mapping Designer, and create a mapping.
2. Name the mapping m_Promoltems.
3. From the list of target definitions, select the tables you just created and drag them into the mapping.
4. From the list of source definitions, add the following source definitions to the mapping:
 - PROMOTIONS
 - ITEMS_IN_PROMOTIONS
 - ITEMS

- MANUFACTURERS
 - ORDER_ITEMS
5. Delete all Source Qualifier transformations that the Designer creates when you add these source definitions.
 6. Add a Source Qualifier transformation named SQ_AllData to the mapping, and connect all the source definitions to it.
 7. Click **View > Navigator** to close the Navigator to allow extra space in the workspace.
 8. Click **Repository > Save**.

Creating a Filter Transformation

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

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

1. Create a Filter transformation and name it FIL_CurrentItems.
2. Drag the following ports from the Source Qualifier transformation into the Filter transformation:
 - ITEM_ID
 - ITEM_NAME
 - PRICE
 - DISCONTINUED_FLAG
3. Open the Filter transformation.
4. Click the **Properties** tab to specify the filter condition.
5. Click **Open** in the Filter Condition field.
The **Expression Editor** dialog box appears.
6. Select the word TRUE in the Formula field and press **Delete**.
7. Click the **Ports** tab.
8. Enter `DISCONTINUED_FLAG = 0` to remove discontinued items from the mapping.
9. Click **Validate**, and then click **OK**.
The new filter condition now appears in the Value field.
10. Click **OK** to return to the workspace.

Connecting the Filter Transformation

Now, you need to connect the Filter transformation to the D_ITEMS target table. Currently sold items are written to this target.

1. Connect the ports ITEM_ID, ITEM_NAME, and PRICE to the corresponding columns in D_ITEMS.
2. Click **Repository > Save**.

Creating a Sequence Generator Transformation

A Sequence Generator transformation generates unique values, such as primary keys, for a target in a mapping. You can also use it to cycle through a closed set of values. Many relational databases include sequences, which are special database objects that generate values. The Sequence Generator transformation functions like a sequence object in a database. However, in PowerCenter, you do not need to write SQL code to create and use the sequence in a mapping.

The Sequence Generator transformation has the following properties:

- The starting number (normally 1).
- The current value stored in the repository.
- The number that the Sequence Generator transformation adds to its current value for every request for a new ID.
- The maximum value in the sequence.
- A flag indicating whether the Sequence Generator transformation counter resets to the minimum value once it has reached its maximum value.

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

In the new mapping, you add a Sequence Generator transformation to generate IDs for the fact table F_PROMO_ITEMS. Every time the Integration Service inserts a new row into the target table, it generates a unique ID for PROMO_ITEM_ID.

1. Create a Sequence Generator transformation and name it SEQ_PromoItemID.
2. Open the Sequence Generator transformation.
3. Click the **Ports** tab.

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

Note: You cannot add any new ports to this transformation or reconfigure NEXTVAL and CURRVAL.

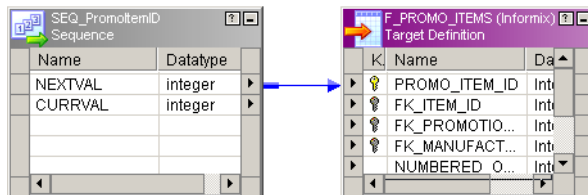
4. Click the **Properties** tab.

The properties for the Sequence Generator transformation appear. You do not have to change any of these settings.

5. Click **OK**.

6. Connect the NEXTVAL column from the Sequence Generator transformation to the PROMO_ITEM_ID column in the target table F_PROMO_ITEMS.

The following image shows a link from the column NEXTVAL to the column PROMO_ITEM_ID:



7. Click **Repository > Save**.

Creating a Stored Procedure Transformation

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

The following table shows the syntax for the stored procedure:

Database	Syntax
Oracle	<pre>CREATE FUNCTION SP_GET_ITEM_COUNT (ARG_ITEM_ID IN NUMBER) RETURN NUMBER IS SP_RESULT NUMBER; BEGIN SELECT COUNT(*) INTO SP_RESULT FROM ORDER_ITEMS WHERE ITEM_ID = ARG_ITEM_ID; RETURN (SP_RESULT); END;</pre>
Microsoft SQL Server	<pre>CREATE PROCEDURE SP_GET_ITEM_COUNT (@ITEM_ID INT) AS SELECT COUNT(*) FROM ORDER_ITEMS WHERE ITEM_ID = @ITEM_ID</pre>
Sybase ASE	<pre>CREATE PROCEDURE SP_GET_ITEM_COUNT (@ITEM_ID INT) AS SELECT COUNT(*) FROM ORDER_ITEMS WHERE ITEM_ID = @ITEM_ID</pre>
Informix	<pre>CREATE PROCEDURE SP_GET_ITEM_COUNT (ITEM_ID_INPUT INT) RETURNING INT; DEFINE CNT INT; SELECT COUNT(*) INTO CNT FROM ORDER_ITEMS WHERE ITEM_ID = ITEM_ID_INPUT; RETURN CNT;</pre>

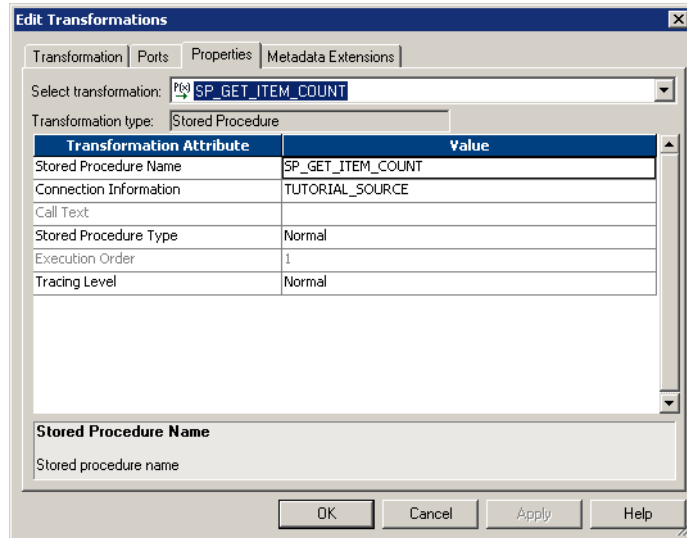
Database	Syntax
DB2	<pre> CREATE PROCEDURE SP_GET_ITEM_COUNT (IN ARG_ITEM_ID INT, OUT SP_RESULT INT, OUT SQLCODE_OUT INT) LANGUAGE SQL P1: BEGIN -- Declare variables DECLARE SQLCODE INT DEFAULT 0; -- Declare handler DECLARE EXIT HANDLER FOR SQLEXCEPTION SET SQLCODE_OUT = SQLCODE; SELECT COUNT(*) INTO SP_RESULT FROM ORDER_ITEMS WHERE ITEM_ID=ARG_ITEM_ID; SET SQLCODE_OUT = SQLCODE; END P1 </pre>
Teradata	<pre> CREATE PROCEDURE SP_GET_ITEM_COUNT (IN ARG_ITEM_ID integer, OUT SP_RESULT integer) BEGIN SELECT COUNT(*) INTO: SP_RESULT FROM ORDER_ITEMS WHERE ITEM_ID =: ARG_ITEM_ID; END; </pre>

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

1. Create a Stored Procedure transformation and name it SP_GET_ITEM_COUNT.
The **Import Stored Procedure** dialog box appears.
2. Select the ODBC connection for the source database. Enter a user name, owner name, and password.
Click **Connect**.
3. Select the stored procedure named SP_GET_ITEM_COUNT from the list and click **OK**.
4. In the **Create Transformation** dialog box, click **Done**.
The Stored Procedure transformation appears in the mapping.
5. Open the Stored Procedure transformation, and click the **Properties** tab.
6. Click **Open** in the Connection Information section.
The **Select Database** dialog box appears.
7. Select the source database and click **OK**.
You can call stored procedures in both source and target databases.

Note: You can also select the built-in database connection variable, **\$Source**. When you use **\$Source** or **\$Target**, the Integration Service determines which source database connection to use when it runs the session. If it cannot determine which connection to use, it fails the session.

The following image shows the **Properties** tab of Stored Procedure transformation **SP_GET_ITEM_COUNT**:



8. Click **OK**.
9. Connect the **ITEM_ID** column from the Source Qualifier transformation to the **ITEM_ID** column in the Stored Procedure transformation.
10. Connect the **RETURN_VALUE** column from the Stored Procedure transformation to the **NUMBER_ORDERED** column in the target table **F_PROMO_ITEMS**.
11. Click **Repository > Save**.

Completing the Mapping

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

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

The following table describes the Source Qualifier transformation columns, target table, and the column name:

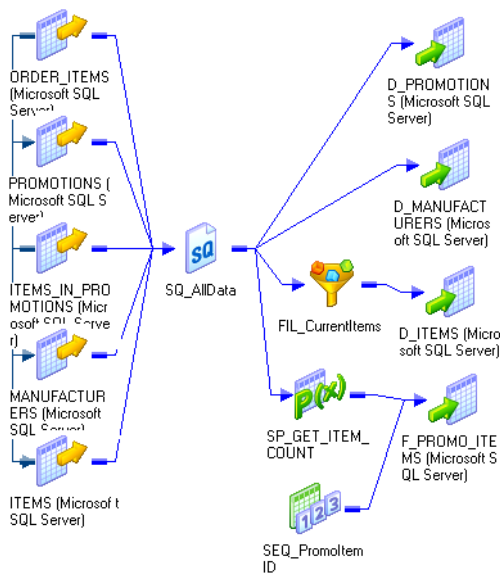
Source Qualifier	Target Table	Column
PROMOTION_ID	D_PROMOTIONS	PROMOTION_ID
PROMOTION_NAME	D_PROMOTIONS	PROMOTION_NAME
DESCRIPTION	D_PROMOTIONS	DESCRIPTION
START_DATE	D_PROMOTIONS	START_DATE
END_DATE	D_PROMOTIONS	END_DATE

Source Qualifier	Target Table	Column
MANUFACTURER_ID	D_MANUFACTURERS	MANUFACTURER_ID
MANUFACTURER_NAME	D_MANUFACTURERS	MANUFACTURER_NAME
ITEM_ID	F_PROMO_ITEMS	FK_ITEM_ID
PROMOTION_ID	F_PROMO_ITEMS	FK_PROMOTION_ID
MANUFACTURER_ID	F_PROMO_ITEMS	FK_MANUFACTURER_ID
DISCOUNT	F_PROMO_ITEMS	DISCOUNT
COMMENTS	F_PROMO_ITEMS	COMMENTS

2. Click **Repository > Save**.

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

The following image shows the mapping you created in this lesson:



Creating a Workflow

In this part of the lesson, you complete the following steps:

1. Create a workflow.
2. Add a non-reusable session to the workflow.
3. Define a link condition before the Session task.

Creating the Workflow

Open the Workflow Manager and connect to the repository.

1. Click **Tools > Workflow Designer**.
2. Click **Workflows > Create** to create a workflow.
The **Create Workflow** dialog box appears.
3. Name the workflow wf_PromoItems.
4. Click the **Open** button to select the Integration Service to run the workflow.
The **Integration Service Browser** dialog box appears.
5. Select the Integration Service you use and click **OK**.
6. Click the **Scheduler** tab.
By default, the workflow is scheduled to run on demand. Keep this default.
7. Click **OK** to close the **Create Workflow** dialog box.
The Workflow Manager creates a workflow in the workspace including the Start task.
Next, you add a non-reusable session in the workflow.

Adding a Non-Reusable Session

1. Click **Tasks > Create**.
The **Create Task** dialog box appears. The Workflow Designer provides more task types than the Task Developer. These tasks include the Email and Decision tasks.
2. Type "s_PromoItems" in the Name field and click **Create** to create a Session task.
3. In the **Mappings** dialog box, select the mapping m_PromoItems and click **OK**.
4. Click **Done**.
5. Open the session properties for s_PromoItems.
6. Click the **Mapping** tab.
7. Select the source database connection for the sources connected to the SQ_AllData Source Qualifier transformation.
8. Select the target database for each target definition.
9. Click **OK** to save the changes.
10. Click the **Link Tasks** button on the toolbar.
11. Drag from the Start task to s_PromoItems.
12. Click **Repository > Save** to save the workflow in the repository.

You can now define a link condition in the workflow.

Defining a Link Condition

After you create links between tasks, you can specify conditions for each link to determine the order of execution in the workflow. If you do not specify conditions for each link, the Integration Service executes the next task in the workflow by default.

If the link condition evaluates to True, the Integration Service runs the next task in the workflow. The Integration Service does not run the next task in the workflow if the link condition evaluates to False. You can also use pre-defined or user-defined workflow variables in the link condition.

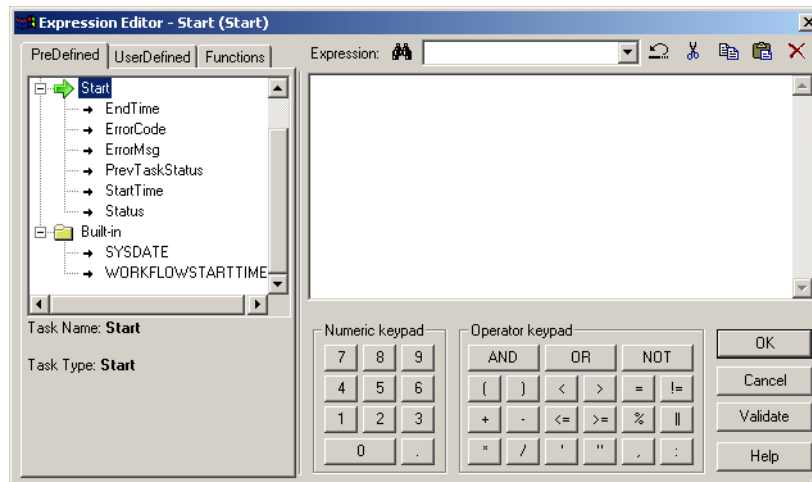
You can use the -- or // comment indicators with the **Expression Editor** to add comments. Use comments to describe the expression.

You can view results of link evaluation during workflow runs in the workflow log.

In the following steps, you create a link condition before the Session task and use the built-in workflow variable WORKFLOWSTARTTIME. You define the link condition so the Integration Service runs the session if the workflow start time is before the date you specify.

1. Double-click the link from the Start task to the Session task.

The following image shows the **Expression Editor** that appears:



2. Expand the Built-in node on the **PreDefined** tab.

The Workflow Manager displays the two built-in workflow variables, SYSDATE and WORKFLOWSTARTTIME.

3. Enter the following expression in the expression window. Enter a date later than today's date:

```
WORKFLOWSTARTTIME < TO_DATE('8/30/2007', 'MM/DD/YYYY')
```

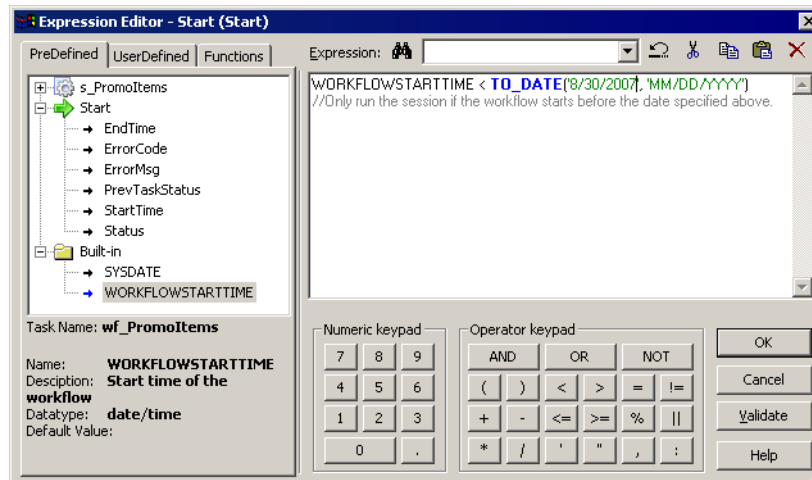
Tip: You can double-click the built-in workflow variable on the **PreDefined** tab and double-click the TO_DATE function on the **Functions** tab to enter the expression.

4. Press Enter to create a new line in the Expression.

Add a comment by typing the following text:

```
// Only run the session if the workflow starts before the date specified above.
```

The following image shows the contents of the **PreDefined** tab:



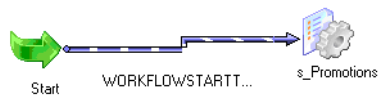
5. Click **Validate** to validate the expression.

The Workflow Manager displays a message in the **Output** window.

6. Click **OK**.

After you specify the link condition in the **Expression Editor**, the Workflow Manager validates the link condition and displays it next to the link in the workflow.

The following image displays the link condition next to the link in the workflow:



7. Click **Repository > Save** to save the workflow in the repository.

Next, you run and monitor the workflow.

Running the Workflow

After you create the workflow, you can run it and use the Workflow Monitor to monitor the workflow progress.

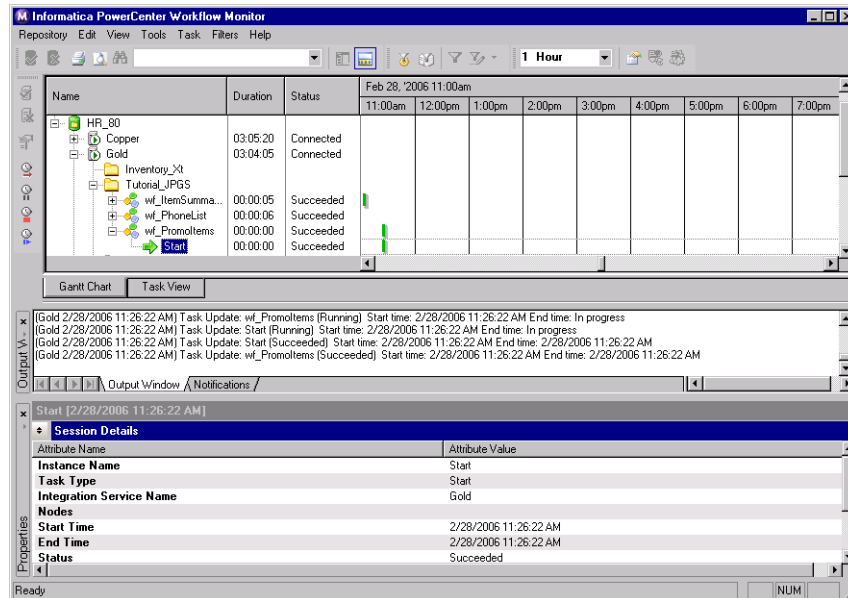
1. Right-click the workflow in the workspace and select **Start Workflow**.

Tip: You can also right-click the workflow in the Navigator and select **Start Workflow**.

The Workflow Monitor opens and connects to the repository and opens the tutorial folder. You can set the **Truncate target table** option on the **Mapping** tab to prevent errors when the workflow runs.

2. Click the **Gantt Chart** tab at the bottom of the **Time** window to verify the Workflow Monitor is in Gantt Chart view.
3. In the Navigator, expand the node for the workflow.

The following image shows that all the tasks in the workflow appear in the Navigator:



4. In the **Properties** window, click **Session Statistics** to view the workflow results.

If the **Properties** window is not open, click **View > Properties View**.

The results from running the s_Promoltems session are:

```

D_ITEMS Output Rows [13]
D_MANUFACTURERS Output Rows [11]
D_PROMOTIONS Output Rows [3]
F_PROMO_ITEMS Output Rows [36]
  
```

CHAPTER 8

Tutorial Lesson 6

This chapter includes the following topics:

- [Using XML Files, 88](#)
- [Creating the XML Source, 89](#)
- [Creating the Target Definition, 95](#)
- [Creating a Mapping with XML Sources and Targets, 97](#)
- [Creating a Workflow, 102](#)

Using XML Files

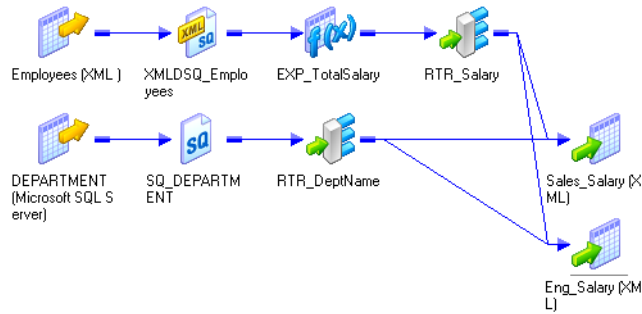
XML is a common means of exchanging data on the web. Use XML files as a source of data and as a target for transformed data.

In this lesson, you have an XML schema file that contains data on the salary of employees in different departments, and you have relational data that contains information about the different departments. You want to find out the total salary for employees in two departments, and you want to write the data to a separate XML target for each department.

In the XML schema file, employees can have three types of wages, which appear in the XML schema file as three occurrences of salary. You pivot the occurrences of employee salaries into three columns: BASESALARY, COMMISSION, and BONUS. Then you calculate the total salary in an Expression transformation.

You use a Router transformation to test for the department ID. You use another Router transformation to get the department name from the relational source. You send the salary data for the employees in the Engineering department to one XML target and the salary data for the employees in the Sales department to another XML target.

The following figure shows the mapping you create in this lesson:



Creating the XML Source

You use the **XML Wizard** to import an XML source definition. You then use the **XML Editor** to edit the definition.

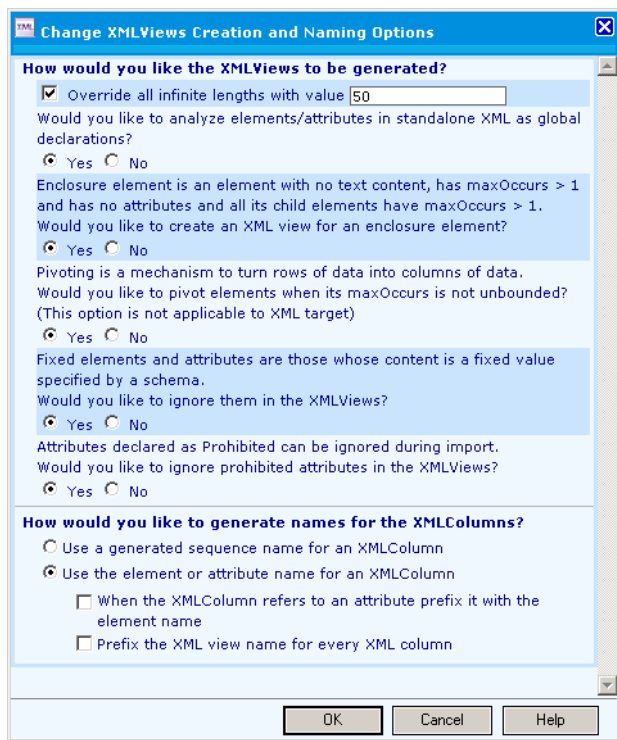
Importing the XML Source

You import the Employees.xsd file to create the XML source definition.

1. Open the Designer, connect to the repository, and open the tutorial folder.
2. Click **Tools > Source Analyzer**.
3. Click **Sources > Import XML Definition**.
4. Click **Advanced Options**.

The **Change XML Views Creation and Naming Options** dialog box appears.

The following image shows **Change XML Views Creation and Naming Options** dialog box:



5. Select **Override All Infinite Lengths with value** and enter 50.
6. Configure the options and click **OK** to save the changes.

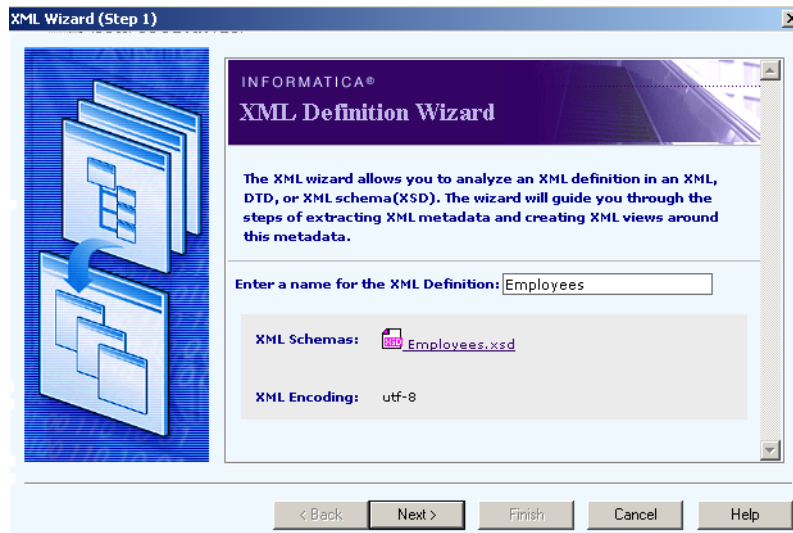
The following table describes the options and values:

Option	Value
Analyze attributes/elements in standalone XML as global declarations	Yes
Create an XML view for an enclosure element	Yes
Pivot elements when its maxOccurs is not unbounded	Yes
Ignore fixed elements and attributes whose content is a fixed value specified by a schema	Yes
Ignore prohibited attributes in XMLViews	Yes
Generate names for the XML columns/Select Use the element or attribute name for an XML column	Yes

7. In the **Import XML Definition** dialog box, navigate to the following directory:
`c:\PowerCenterClientInstallationDir\client\bin`
8. Select the **Employees.xsd** file. Click **Open**.

The **XML Wizard** appears.

The following image shows the **XML Wizard**:



9. Verify that the name for the XML definition is Employees and click **Next**.
10. Select **Do not generate XML views**.
11. Click **Finish** to create the XML definition.

When you skip creating XML views, the Designer imports metadata into the repository, but it does not create the XML view. In the next step, you use the XML Editor to add groups and columns to the XML view.

Editing the XML Definition

The Designer represents an XML hierarchy in an XML definition as a set of views. Each view represents a subset of the XML hierarchy. A view consists of columns and rows. Columns represent elements and attributes, and rows represent occurrences of elements. You use the XML Editor to edit the XML views.

In this lesson, you use the **XML Editor** to pivot the three occurrences of SALARY into three columns in an XML group. You do this because the multiple-occurring element SALARY represents three types of salary: a base salary, a commission, and a bonus that appear in the XML file as three instances of the SALARY element.

The following image shows contents of Employees.xml in the editor Notepad:

```

Employees.xml - Notepad
File Edit Format Help
<?xml version="1.0" encoding='utf-8'?>
<EMPLOYEES>
  <EMPLOYEE EMPID="105" DEPTID="FIN" >
    <LASTNAME>Smith</LASTNAME>
    <FIRSTNAME>Martha</FIRSTNAME>
    <ADDRESS>
      <STREETADDRESS>335 Westshore Road</STREETADDRESS>
      <CITY>Fausta City</CITY>
      <STATE>CA</STATE>
      <ZIP>97584</ZIP>
    </ADDRESS>
    <PHONE>(415)552-1623</PHONE>
    <EMAIL>msmith@acme.com</EMAIL>
    <EMAIL>martha@smith.com</EMAIL>
    <EMPLOYMENT EMPLSTAT="PF">
      <DATEOFHIRE>12-02-1997</DATEOFHIRE>
      <SALARY>65350</SALARY>
      <SALARY>5000</SALARY>
      <SALARY>0</SALARY>
    </EMPLOYMENT>
  </EMPLOYEE>
  <EMPLOYEE EMPID="53" DEPTID="ENG" >
    <LASTNAME>Loren</LASTNAME>
    <FIRSTNAME>Pamela</FIRSTNAME>
    <ADDRESS>
      <STREETADDRESS>19 B. Candido Street</STREETADDRESS>
      <CITY>Zamboanga City</CITY>
      <STATE>CA</STATE>
      <ZIP>94404</ZIP>
    </ADDRESS>
    <PHONE>(415)560-1023</PHONE>
    <PHONE>(650)584-7970</PHONE>
    <EMAIL>ploren@acme.com</EMAIL>
    <EMAIL>pam1@myyahoo.com</EMAIL>
    <EMAIL>pam1@mydomain.com</EMAIL>
    <EMPLOYMENT EMPLSTAT="TF">
      <DATEOFHIRE>07-05-1998</DATEOFHIRE>
      <SALARY>72450</SALARY>
      <SALARY>0</SALARY>
      <SALARY>2500</SALARY>
    </EMPLOYMENT>
  </EMPLOYEE>

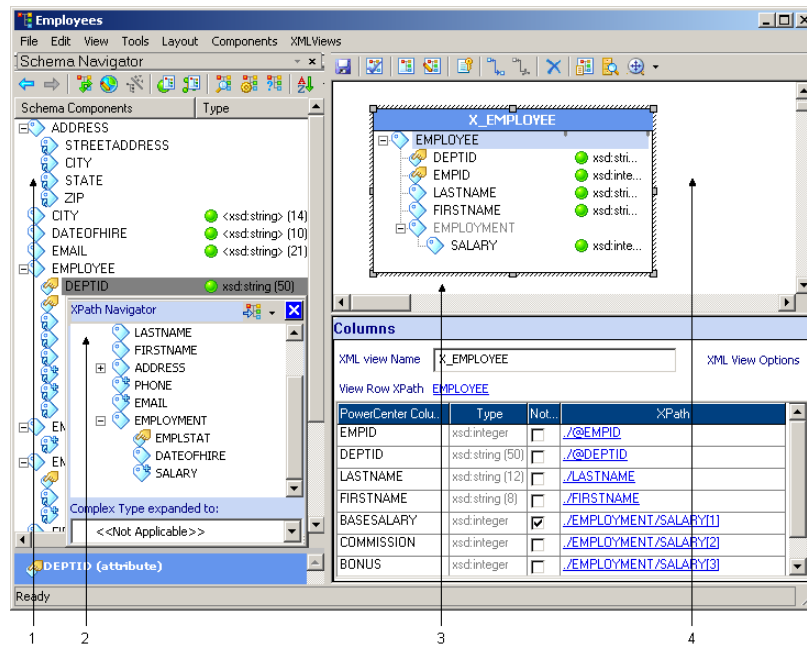
```

1. Bonus
2. Commission
3. Base salary

To work with these three instances separately, you pivot them to create three separate columns in the XML definition.

You create a custom XML view with columns from several groups. You then pivot the occurrence of SALARY to create the columns, BASESALARY, COMMISSION, and BONUS.

The following image shows the **XML Editor**:



1. Navigator

2. XPath Navigator

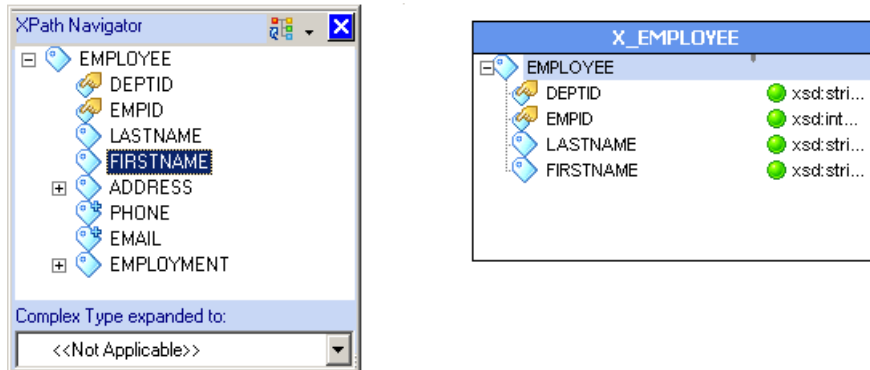
3. XML View

4. XML Workspace

1. Double-click the XML definition or right-click the XML definition and select **Edit XML Definition** to open the **XML Editor**.
2. Click **XMLViews > Create XML View** to create an XML view.
3. From the EMPLOYEE group, select DEPTID and right-click it.
4. Choose **Show XPath Navigator**.
5. Expand the EMPLOYMENT group so that the SALARY column appears.
6. From the **XPath Navigator**, select the following elements and attributes and drag them into the view:
 - DEPTID
 - EMPID
 - LASTNAME

- FIRSTNAME

The following image shows the **XPath Navigator** and X_EMPLOYEE view:



The XML Editor names the view X_EMPLOYEE.

Note: The **XML Wizard** can transpose the order of the DEPTID and EMPID attributes when it imports them. If this occurs, you can add the columns in the order they appear in the **Schema Navigator** or **XPath Navigator**. Transposing the order of attributes does not affect data consistency.

7. Click the **Mode** button on the **XPath Navigator** and choose **Advanced Mode**.
8. Select the SALARY column and drag it into the XML view.

Note: The **XPath Navigator** must include the EMPLOYEE column at the top when you drag SALARY to the XML view.

The following image shows the resulting view that includes the elements and attributes shown in the XML view:



9. Drag the SALARY column into the XML view two more times to create three pivoted columns.

Note: Although the columns appear in the column window, the view shows one instance of SALARY.

The wizard adds three columns in the column view and names them SALARY, SALARY0, and SALARY1.

The following image shows three columns SALARY, SALARY0, and SALARY1 in the columns view:

Columns			
XML view Name		X_EMPLOYEE	
View Row XPath		EMPLOYEE	
PowerCenter Column Name	Type	Not Null	XPath
LASTNAME	xsd:string (12)	<input type="checkbox"/>	./LASTNAME
FIRSTNAME	xsd:string (8)	<input type="checkbox"/>	./FIRSTNAME
SALARY	xsd:integer	<input type="checkbox"/>	./EMPLOYMENT/SALARY[1]
SALARY0	xsd:integer	<input type="checkbox"/>	./EMPLOYMENT/SALARY[2]
SALARY1	xsd:integer	<input type="checkbox"/>	./EMPLOYMENT/SALARY[3]

10. Rename the columns.

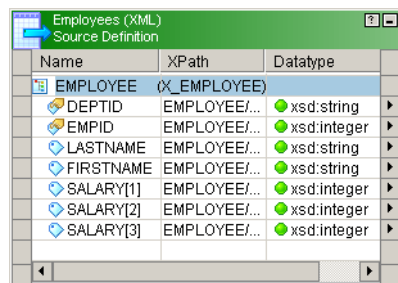
The following table describes the information to modify the name and pivot properties:

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

Note: To update the pivot occurrence, click the Xpath of the column you want to edit. The **Specify query predicate for XPath** window appears. Select the column name and change the pivot occurrence.

11. Click **File > Apply Changes** to save the changes to the view.
12. Click **File > Exit** to close the XML Editor.

The following image shows the source definition that appears in the Source Analyzer, with all of the listed EMPLOYEE attributes and elements:



Note: The pivoted SALARY columns do not display the names you entered in the Columns window. However, when you drag the ports to another transformation, the edited column names appear in the transformation.

13. Click **Repository > Save** to save the changes to the XML definition.

Creating the Target Definition

In this lesson, you import an XML schema and create a custom view based on the schema. The custom XML target definition you create meets the following criteria:

- Each department has a separate target and the structure for each target is the same.
- Each target contains salary and department information for employees in the Sales or Engineering department.

Because the structure for the target data is the same for the Engineering and Sales groups, use two instances of the target definition in the mapping. In the following steps, you import the Sales_Salary schema file and create a custom view based on the schema.

1. In the Designer, switch to the Target Designer.
If the workspace contains targets from other lessons, right-click the workspace and choose **Clear All**.
2. Click **Targets > Import XML Definition**.

3. Navigate to the Tutorial directory in the PowerCenter installation directory, and select the Sales_Salary.xsd file. Click **Open**.

The **XML Wizard** appears.

4. Name the XML definition SALES_SALARY and click **Next**.

5. Select **Do not generate XML views** and click **Finish**.

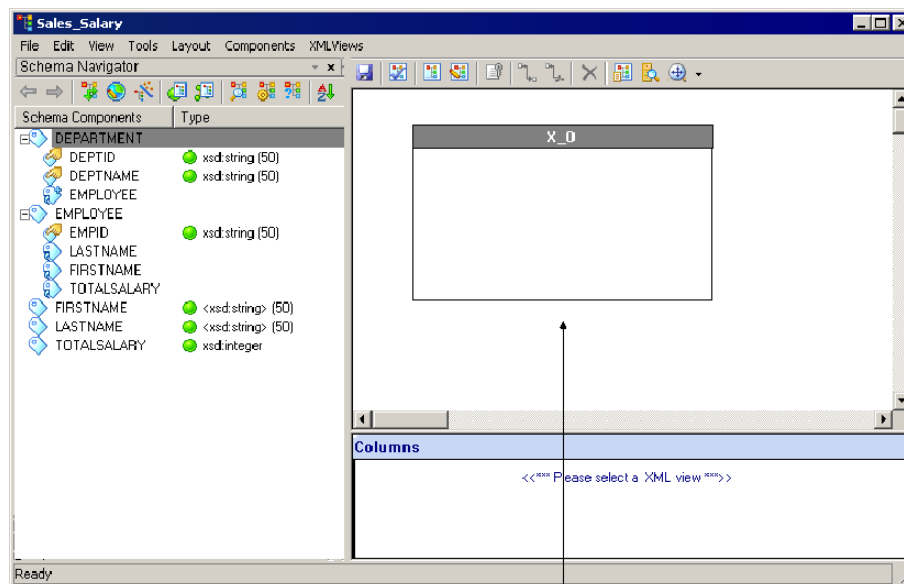
The **XML Wizard** creates the SALES_SALARY target with no columns or groups.

6. Double-click the XML definition to open the **XML Editor**.

7. Click **XMLViews > Create XML View**.

The **XML Editor** creates an empty XML view.

The following image shows an empty XML view:



1

1. Empty XML View

8. Right-click **DEPARTMENT** group in the **Schema Navigator** and select **Show XPath Navigator**.

9. From the **XPath Navigator**, drag **DEPTNAME** and **DEPTID** into the empty XML view.

The **XML Editor** names the view **X_DEPARTMENT**.

Note: The **XML Editor** may transpose the order of the attributes **DEPTNAME** and **DEPTID**. If this occurs, add the columns in the order they appear in the **Schema Navigator**. Transposing the order of attributes does not affect data consistency.

10. In the **X_DEPARTMENT** view, right-click the **DEPTID** column, and choose **Set as Primary Key**.

11. Click **XMLViews > Create XML View**.

The **XML Editor** creates an empty XML view.

12. From the **EMPLOYEE** group in the **Schema Navigator**, Navigate to the **XPath Navigator**.

13. From the **XPath Navigator**, drag **EMPID**, **FIRSTNAME**, **LASTNAME**, and **TOTALSALARY** into the empty XML view.

The **XML Editor** names the view **X_EMPLOYEE**.

14. Right-click the **X_EMPLOYEE** view and choose **Create Relationship**.

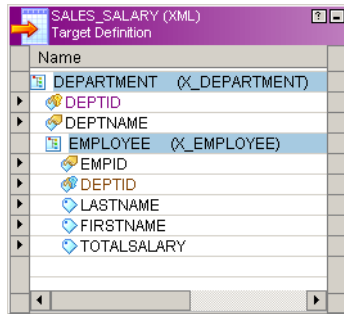
Drag the pointer from the **X_EMPLOYEE** view to the **X_DEPARTMENT** view to create a link.

15. The **XML Editor** creates a DEPARTMENT foreign key in the X_EMPLOYEE view that corresponds to the DEPTID primary key.

16. Click **File > Apply Changes**.

The XML definition now contains the groups DEPARTMENT and EMPLOYEE.

The following image shows an XML definition that contains the groups DEPARTMENT and EMPLOYEE:



17. Click **Repository > Save** to save the XML target definition.

Creating a Mapping with XML Sources and Targets

In the following steps, you create a mapping to transform the employee data. You add the following objects to the mapping:

- The Employees XML source definition you created.
- The DEPARTMENT relational source definition you created in [“Creating Source Definitions” on page 39](#).
- Two instances of the SALES_SALARY target definition you created.
- An Expression transformation to calculate the total salary for each employee.
- Two Router transformations to route salary and department.

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

1. In the Designer, switch to the Mapping Designer and create a mapping.
2. Name the mapping m_EmployeeSalary.
3. Drag the Employees XML source definition into the mapping.
4. Drag the DEPARTMENT relational source definition into the mapping.
By default, the Designer creates a source qualifier for each source.
5. Drag the SALES_SALARY target definition into the mapping two times.
6. Rename the second instance of SALES_SALARY as ENG_SALARY.
7. Click **Repository > Save**.

The Designer displays a warning that the mapping m_EmployeeSalary is not valid because you have not completed the mapping.

Next, you add an Expression transformation and two Router transformations. Then, you connect the source definitions to the Expression transformation. You connect the pipeline to the Router transformations and then to the two target definitions.

Creating an Expression Transformation

In the following steps, you use an Expression transformation to calculate the total salary for each employee. You use BASESALARY, COMMISSION, and BONUS as input columns to the Expression transformation and create a TotalSalary column as output.

1. Create an Expression transformation and name it EXP_TotalSalary.
2. Click **Done**.
3. Drag all the ports from the XML Source Qualifier transformation to the EXP_TotalSalary Expression transformation.

The input/output ports in the XML Source Qualifier transformation are linked to the input/output ports in the Expression transformation.

4. Open the Expression transformation.
5. On the **Ports** tab, add an output port named TotalSalary. Use the Decimal datatype with precision of 10 and scale of 2.
6. Enter the following expression for TotalSalary:

```
BASESALARY + COMMISSION + BONUS
```
7. Validate the expression and click **OK**.
8. Click **OK** to close the transformation.
9. Click **Repository > Save**.

Creating Router Transformations

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.

In the following steps, you add two Router transformations to the mapping, one for each department. In each Router transformation you create two groups. One group returns True for rows where the DeptID column contains 'SLS'. The other group returns True where the DeptID column contains 'ENG'. All rows that do not meet either condition go into the default group.

Routing Employee Salary Data

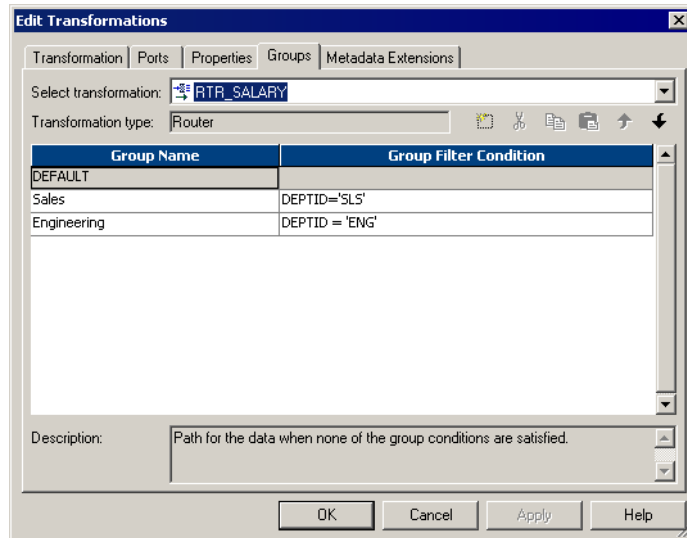
1. Create a Router transformation and name it RTR_Salary. Then click **Done**.
2. In the Expression transformation, select the following columns and drag them to RTR_Salary:
 - EmpID
 - DeptID
 - LastName
 - FirstName
 - TotalSalary

The Designer creates an input group and adds the columns you drag from the Expression transformation.

3. Open the RTR_Salary Router transformation and select the **Groups** tab.

The **Edit Transformations** dialog box appears, with RTR_SALARY listed in the Select Transformation field.

The following image shows the **Groups** tab in the **Edit Transformations** dialog box:



- On the **Groups** tab, add two groups.
Change the group names and set the filter conditions.

The following table describes the groups names and the filter conditions:

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

The Designer adds a default group to the list of groups. All rows that do not meet the condition you specify in the group filter condition are routed to the default group. If you do not connect the default group, the Integration Service drops the rows.

- Click **OK** to close the transformation.
- In the workspace, expand the RTR_Salary Router transformation to see all groups and ports.
- Click **Repository > Save**.

Next, you create another Router transformation to filter the Sales and Engineering department data from the DEPARTMENT relational source.

Routing Department Data

- Create a Router transformation and name it RTR_DeptName. Then click **Done**.
- Drag the DeptID and DeptName ports from the DEPARTMENT Source Qualifier transformation to the RTR_DeptName Router transformation.
- Open RTR_DeptName.
- On the **Groups** tab, add two groups.

Change the group names and set the filter conditions.

The following table describes the group names and the filter conditions:

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

- On the **Ports** tab, change the precision for DEPTNAME to 15.
- Click **OK** to close the transformation.
- In the workspace, expand the RTR_DeptName Router transformation to see all groups and columns.
- Click **Repository > Save**.

Completing the Mapping

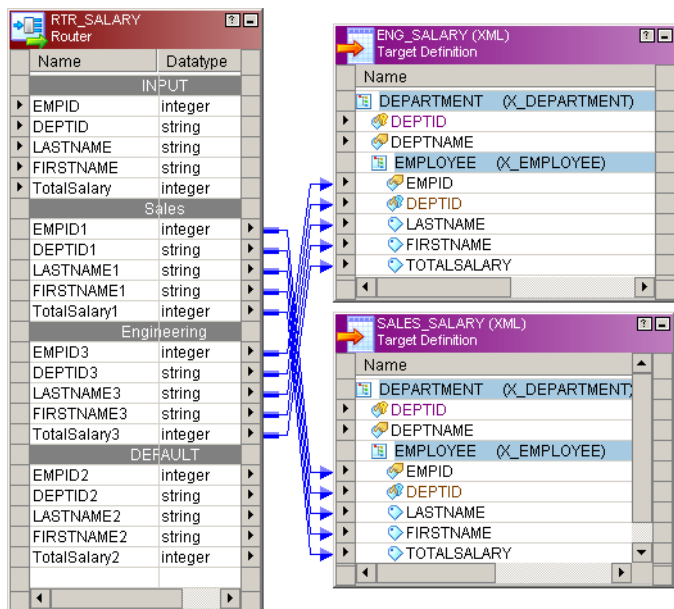
Connect the Router transformations to the targets to complete the mapping.

- Connect the ports from RTR_Salary groups to the ports in the XML target definitions.

The following table describes the ports in RTR_Salary groups and the ports in the XML target definitions:

Router Group	Router Port	Target	Target Group	Target Port
Sales	EMPID1	SALES_SALARY	EMPLOYEE	EMPID
Sales	DEPTID1	SALES_SALARY	EMPLOYEE	DEPTID (FK)
Sales	LASTNAME1	SALES_SALARY	EMPLOYEE	LASTNAME
Sales	FIRSTNAME 1	SALES_SALARY	EMPLOYEE	FIRSTNAME
Sales	TotalSalary	SALES_SALARY	EMPLOYEE	TOTALSALARY
Engineering	EMPID3	ENG_SALARY	EMPLOYEE	EMPID
Engineering	DEPTID3	ENG_SALARY	EMPLOYEE	DEPTID (FK)
Engineering	LASTNAME3	ENG_SALARY	EMPLOYEE	LASTNAME
Engineering	FIRSTNAME 3	ENG_SALARY	EMPLOYEE	FIRSTNAME
Engineering	TotalSalary3	ENG_SALARY	EMPLOYEE	TOTALSALARY

The following figure shows the Router transformation connected to the target definitions:



2. Connect the ports from RTR_DeptName groups to the ports in the XML target definitions.

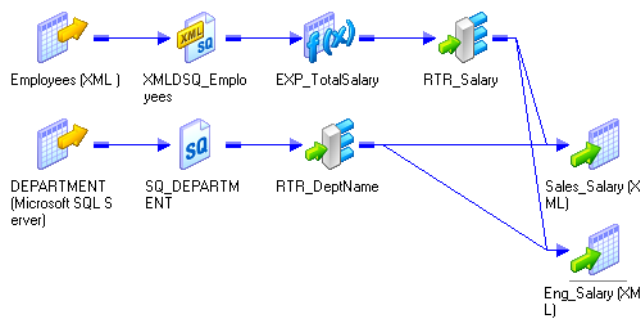
The following table describes the ports in RTR_DeptName and the ports in XML target definitions:

Router Group	Router Port	Target	Target Group	Target Port
Sales	DEPTID1	SALES_SALARY	DEPARTMENT	DEPTID
Sales	DEPTNAME1	SALES_SALARY	DEPARTMENT	DEPTNAME
Engineering	DEPTID3	ENG_SALARY	DEPARTMENT	DEPTID
Engineering	DEPTNAME3	ENG_SALARY	DEPARTMENT	DEPTNAME

3. Click **Repository > Save**.

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

The image shows the mapping m_EmployeeSalary:



Creating a Workflow

In the following steps, you create a workflow with a non-reusable session to run the mapping you just created.

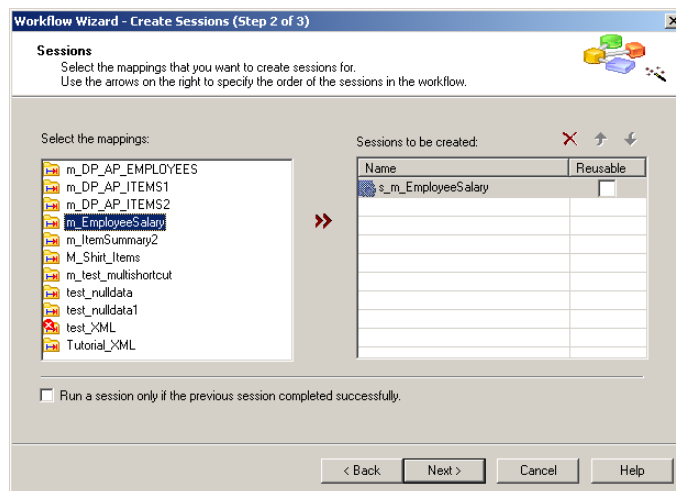
Note: Verify that the Integration Service that runs the workflow can access the source XML file. Copy the Employees.xml file from the Tutorial folder to the \$PMSourceFileDir directory for the Integration Service and rename it data.xml. Usually, this is the SrcFiles directory in the Integration Service installation directory.

1. Open the Workflow Manager.
2. Connect to the repository and open the tutorial folder.
3. Go to the Workflow Designer.
4. Click **Workflows > Wizard**.

The **Workflow Wizard** opens.

5. Name the workflow wf_EmployeeSalary and select a service on which to run the workflow. Then click **Next**.
6. Select the m_EmployeeSalary mapping to create a session.

The following image shows the mapping m_EmployeeSalary highlighted :



The **Workflow Wizard** creates a session called s_m_EmployeeSalary.

7. Click **Next**.
8. Click **Run on demand** and click **Next**.

The **Workflow Wizard** displays the settings you chose.

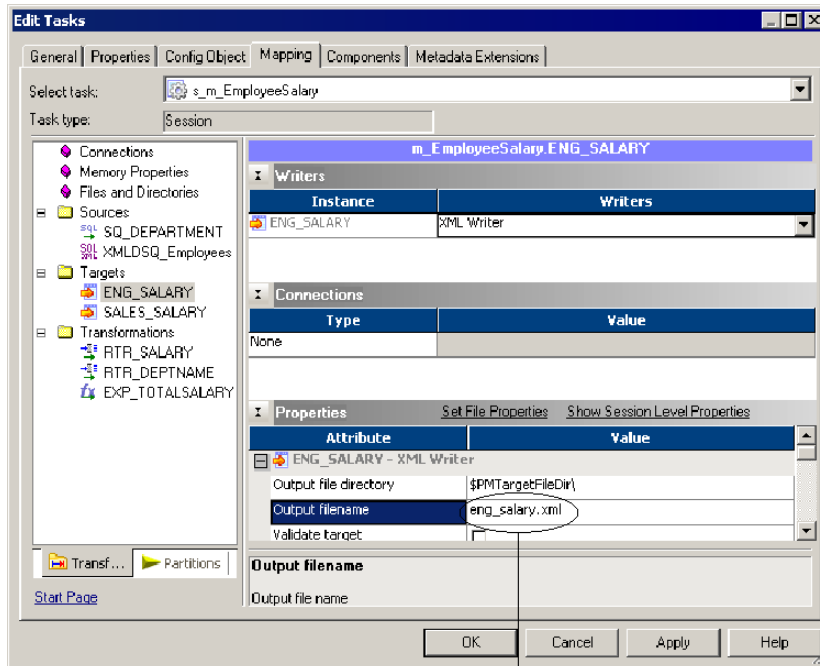
9. Click **Finish** to create the workflow.

The **Workflow Wizard** creates a Start task and session. You can add other tasks to the workflow later.

10. Click **Repository > Save** to save the workflow.
11. Double-click the s_m_EmployeeSalary session to open it for editing.
12. Click the **Mapping** tab.
13. Select the connection for the SQ_DEPARTMENT Source Qualifier transformation.
14. Select the XMLDSQ_Employees source on the **Mapping** tab.
15. Verify that the data.xml file is in the specified source file directory.

- Click the ENG_SALARY target instance on the **Mapping** tab and verify that the output file name is eng_salary.xml.

The following image shows the contents of the **Mapping** tab:



1

- Output file name
- Click the SALES_SALARY target instance on the **Mapping** tab and verify that the output file name is sales_salary.xml.
- Click **OK** to close the session.
- Click **Repository > Save**.
- Run and monitor the workflow.

The Integration Service creates the eng_salary.xml and sales_salary.xml files.

APPENDIX A

Naming Conventions

This appendix includes the following topic:

- [Suggested Naming Conventions, 104](#)

Suggested Naming Conventions

The following naming conventions appear throughout the PowerCenter documentation and client tools. Use the following naming conventions when you design mappings and create sessions.

Naming Convention

The following table lists the recommended naming conventions for transformations:

Transformation	Naming Convention
Aggregator	<i>AGG_TransformationName</i>
Application Source Qualifier	<i>ASQ_TransformationName</i>
Custom	<i>CT_TransformationName</i>
Expression	<i>EXP_TransformationName</i>
External Procedure	<i>EXT_TransformationName</i>
Filter	<i>FIL_TransformationName</i>
HTTP	<i>HTTP_TransformationName</i>
Java	<i>JTX_TransformationName</i>
Joiner	<i>JNR_TransformationName</i>
Lookup	<i>LKP_TransformationName</i>
MQ Source Qualifier	<i>SQ_MQ_TransformationName</i>
Normalizer	<i>NRM_TransformationName</i>

Transformation	Naming Convention
Rank	<i>RNK_TransformationName</i>
Router	<i>RTR_TransformationName</i>
Sequence Generator	<i>SEQ_TransformationName</i>
Sorter	<i>SRT_TransformationName</i>
Stored Procedure	<i>SP_TransformationName</i>
Source Qualifier	<i>SQ_TransformationName</i>
SQL	<i>SQL_TransformationName</i>
Transaction Control	<i>TC_TransformationName</i>
Union	<i>UN_TransformationName</i>
Unstructured Data Transformation	<i>UD_TransformationName</i>
Update Strategy	<i>UPD_TransformationName</i>
XML Generator	<i>XG_TransformationName</i>
XML Parser	<i>XP_TransformationName</i>
XML Source Qualifier	<i>XSQ_TransformationName</i>

The following table lists recommended naming conventions for other objects:

Object Type	Naming Convention
Targets	<i>t_TargetName</i>
Mappings	<i>m_MappingName</i>
Mapplets	<i>mplt_MappletName</i>
Sessions	<i>s_SessionName</i>
Worklets	<i>wl_WorkletName</i>

APPENDIX B

Glossary

active database

The database to which transformation logic is pushed during pushdown optimization.

active source

An active source is an active transformation the Integration Service uses to generate rows.

adaptive dispatch mode

A dispatch mode in which the Load Balancer dispatches tasks to the node with the most available CPUs.

application service

A service that runs on one or more nodes in the Informatica domain. You create and manage application services in Informatica Administrator or through the `infacmd` command program. Application services include services that can have multiple instances in the domain and system services that can have a single instance in the domain. Configure each application service based on your environment requirements.

A service that runs on the node in the Informatica domain. You manage application services in Informatica Administrator or through the `infacmd` command program. Configure each application service based on your environment requirements.

associated service

An application service that you associate with another application service. For example, you associate a Repository Service with an Integration Service.

An application service that is associated with another application service. For example, the Model Repository Service is associated with the Data Integration Service.

attachment view

View created in a web service source or target definition for a WSDL that contains a mime attachment. The attachment view has an $n:1$ relationship with the envelope view.

available resource

Any PowerCenter resource that is configured to be available to a node.

backup node

Any node that is configured to run a service process, but is not configured as a primary node.

blocking

The suspension of the data flow into an input group of a multiple input group transformation.

blurring

A masking rule that limits the range of numeric output values to a fixed or percent variance from the value of the source data. The Data Masking transformation returns numeric data that is close to the value of the source data.

bounds

A masking rule that limits the range of numeric output to a range of values. The Data Masking transformation returns numeric data between the minimum and maximum bounds.

buffer block

A block of memory that the Integration Services uses to move rows of data from the source to the target. The number of rows in a block depends on the size of the row data, the configured buffer block size, and the configured buffer memory size.

buffer block size

The size of the blocks of data used to move rows of data from the source to the target. You specify the buffer block size in bytes or as percentage of total memory. Configure this setting in the session properties.

buffer memory

Buffer memory allocated to a session. The Integration Service uses buffer memory to move data from sources to targets. The Integration Service divides buffer memory into buffer blocks.

buffer memory size

Total buffer memory allocated to a session specified in bytes or as a percentage of total memory.

built-in parameters and variables

Predefined parameters and variables that do not vary according to task type. They return system or run-time information such as session start time or workflow name. Built-in variables appear under the Built-in node in the Designer or Workflow Manager Expression Editor.

cache partitioning

A caching process that the Integration Service uses to create a separate cache for each partition. Each partition works with only the rows needed by that partition. The Integration Service can partition caches for the Aggregator, Joiner, Lookup, and Rank transformations.

child dependency

A dependent relationship between two objects in which the child object is used by the parent object.

child object

A dependent object used by another object, the parent object.

cold start

A start mode that restarts a task or workflow without recovery.

commit number

A number in a target recovery table that indicates the amount of messages that the Integration Service loaded to the target. During recovery, the Integration Service uses the commit number to determine if it wrote messages to all targets.

commit source

An active source that generates commits for a target in a source-based commit session.

compatible version

An earlier version of a client application or a local repository that you can use to access the latest version repository.

composite object

An object that contains a parent object and its child objects. For example, a mapping parent object contains child objects including sources, targets, and transformations.

concurrent workflow

A workflow configured to run multiple instances at the same time. When the Integration Service runs a concurrent workflow, you can view the instance in the Workflow Monitor by the workflow name, instance name, or run ID.

Configuration Support Manager

A web-based application that you can use to diagnose issues in your Informatica environment and identify Informatica updates. It helps you discover comprehensive information about your technical environment and diagnose issues before they become critical.

coupled group

An input group and output group that share ports in a transformation.

CPU profile

An index that ranks the computing throughput of each CPU and bus architecture in a grid. In adaptive dispatch mode, nodes with higher CPU profiles get precedence for dispatch.

custom role

A role that you can create, edit, and delete.

Custom transformation

A transformation that you bind to a procedure developed outside of the Designer interface to extend PowerCenter functionality. You can create Custom transformations with multiple input and output groups.

Custom transformation procedure

A C procedure you create using the functions provided with PowerCenter that defines the transformation logic of a Custom transformation.

Custom XML view

An XML view that you define instead of allowing the XML Wizard to choose the default root and columns in the view. You can create custom views using the XML Editor and the XML Wizard in the Designer.

data masking

A process that creates realistic test data from production source data. The format of the original columns and relationships between the rows are preserved in the masked data.

Data Masking transformation

A passive transformation that replaces sensitive columns in source data with realistic test data. Each masked column retains the datatype and the format of the original data.

Data Transformation Manager (DTM)

The component of the Data Integration Service that extracts, transforms, and loads data to complete a data transformation job. The DTM also runs profiles and data previews.

default permissions

The permissions that each user and group receives when added to the user list of a folder or global object. Default permissions are controlled by the permissions of the default group, "Other."

denormalized view

An XML view that contains more than one multiple-occurring element.

dependent object

An object used by another object. A dependent object is a child object.

dependent services

A service that depends on another service to run processes. For example, the Integration Service cannot run workflows if the Repository Service is not running.

deployment group

A global object that contains references to other objects from multiple folders across the repository. You can copy the objects referenced in a deployment group to multiple target folders in another repository. When you copy objects in a deployment group, the target repository creates new versions of the objects. You can create a static or dynamic deployment group.

Design Objects privilege group

A group of privileges that define user actions on the following repository objects: business components, mapping parameters and variables, mappings, mapplets, transformations, and user-defined functions.

deterministic output

Source or transformation output that does not change between session runs when the input data is consistent between runs.

digested password

Password security option for protected web services. The password is the value generated from hashing the password concatenated with a nonce value and a timestamp. The password must be hashed with the SHA-1 hash function and encoded to Base64.

direct staging

A staging method that allows the integration Service to read from a data source and stage the data on a local machine before it passes the data to downstream transformations. You can use direct staging when you read data from one source file.

dispatch mode

A mode used by the Load Balancer to dispatch tasks to nodes in a grid.

domain

A domain is the fundamental administrative unit for Informatica nodes and services.

A domain is the fundamental administrative unit for the Informatica node and services.

dynamic deployment group

A deployment group that is associated with an object query. When you copy a dynamic deployment group, the source repository runs the query and then copies the results to the target repository.

dynamic partitioning

The ability to scale the number of partitions without manually adding partitions in the session properties. Based on the session configuration, the Integration Service determines the number of partitions when it runs the session.

effective Transaction Control transformation

A Transaction Control transformation that does not have a downstream transformation that drops transaction boundaries.

effective transaction generator

A transaction generator that does not have a downstream transformation that drops transaction boundaries.

element view

A view created in a web service source or target definition for a multiple occurring element in the input or output message. The element view has an *n:1* relationship with the envelope view.

envelope view

A main view in a web service source or target definition that contains a primary key and the columns for the input or output message.

exclusive mode

An operating mode for the Repository Service. When you run the Repository Service in exclusive mode, you allow only one user to access the repository to perform administrative tasks that require a single user to access the repository and update the configuration.

failover

The migration of a service or task to another node when the node running or service process become unavailable.

fault view

A view created in a web service target definition if a fault message is defined for the operation. The fault view has an *n:1* relationship with the envelope view.

flush latency

A session condition that determines how often the Integration Service flushes data from the source.

gateway node

Receives service requests from clients and routes them to the appropriate service and node. A gateway node can run application services. In the Administrator tool, you can configure any node to serve as a gateway for a PowerCenter domain. A domain can have multiple gateway nodes.

global object

An object that exists at repository level and contains properties you can apply to multiple objects in the repository. Object queries, deployment groups, labels, and connection objects are global objects.

grid object

An alias assigned to a group of nodes to run sessions and workflows.

group

A set of ports that defines a row of incoming or outgoing data. A group is analogous to a table in a relational source or target definition.

hashed password

Password security option for protected web services. The password must be hashed with the MD5 or SHA-1 hash function and encoded to Base64.

high availability

A PowerCenter option that eliminates a single point of failure in a domain and provides minimal service interruption in the event of failure.

High Group History List

A file that the United States Social Security Administration provides that lists the Social Security Numbers it issues each month. The Data Masking transformation accesses this file when you mask Social Security Numbers.

idle database

The database that does not process transformation logic during pushdown optimization.

impacted object

An object that has been marked as impacted by the PowerCenter Client. The PowerCenter Client marks objects as impacted when a child object changes in such a way that the parent object may not be able to run.

incompatible object

An object that a compatible client application cannot access in the latest version repository.

indirect staging

A staging method that allows the Integration Service to create an indirect file with the names of multiple source files. The Integration Service stages the indirect file and the source files on a local machine before it passes the data to downstream transformations. You can use indirect staging when you have multiple source files.

ineffective Transaction Control transformation

A Transaction Control transformation that has a downstream transformation that drops transaction boundaries, such as an Aggregator transformation with Transaction transformation scope.

ineffective transaction generator

A transaction generator that has a downstream transformation that drops transaction boundaries, such as an Aggregator transformation with Transaction transformation scope.

Informatica domain

A collection of nodes and services that define the Informatica platform. You group nodes and services in a domain based on administration ownership.

Informatica Services

The name of the service or daemon that runs on each node. When you start Informatica Services on a node, you start the Service Manager on that node.

input group

A set of ports that defines a row of incoming data.

Integration Service

An application service that runs data integration workflows and loads metadata into the Metadata Manager warehouse.

Integration Service process

A process that accepts requests from the PowerCenter Client and from *pmcmd*. The Integration Service process manages workflow scheduling, locks and reads workflows, and starts DTM processes.

invalid object

An object that has been marked as invalid by the PowerCenter Client. When you validate or save a repository object, the PowerCenter Client verifies that the data can flow from all sources in a target load order group to the targets without the Integration Service blocking all sources.

key masking

A type of data masking that produces repeatable results for the same source data and masking rules. The Data Masking transformation requires a seed value for the port when you configure it for key masking.

label

A user-defined object that you can associate with any versioned object or group of versioned objects in the repository.

latency

A period of time from when source data changes on a source to when a session writes the data to a target.

Lightweight Directory Access Protocol (LDAP) authentication

One of the authentication methods used to authenticate users logging in to PowerCenter applications. In LDAP authentication, the Service Manager imports groups and user accounts from the LDAP directory service into an LDAP security domain in the domain configuration database. The Service Manager stores the group and user account information in the domain configuration database but passes authentication to the LDAP server.

limit on resilience timeout

An amount of time a service waits for a client to connect or reconnect to the service. The limit can override the client resilience timeout configured for a client.

linked domain

A domain that you link to when you need to access the repository metadata in that domain.

Load Balancer

A component of the Integration Service that dispatches Session, Command, and predefined Event-Wait tasks across nodes in a grid.

local domain

A PowerCenter domain that you create when you install PowerCenter. This is the domain you access when you log in to the Administrator tool.

Log Agent

A Service Manager function that provides accumulated log events from session and workflows. You can view session and workflow logs in the Workflow Monitor. The Log Agent runs on the nodes where the Integration Service process runs.

Log Manager

A Service Manager function that provides accumulated log events from each service in the domain. You can view logs in the Administrator tool. The Log Manager runs on the master gateway node.

mapping

A set of source and target definitions linked by transformation objects that define the rules for data transformation.

Mapping Architect for Visio

A PowerCenter Client feature that you use to create mapping templates in Visio and import them into the PowerCenter Designer.

mapplet

A mapplet is a set of transformations that you build in the Mapplet Designer. Create a mapplet when you want to reuse the logic in multiple mappings.

masking algorithm

Sets of characters and rotors of numbers that provide the logic to mask data. A masking algorithm provides random results that ensure that the original data cannot be disclosed from the masked data.

master gateway node

The entry point to the Informatica domain. When you configure multiple gateway nodes, one node acts as the master gateway node. If the master gateway node becomes unavailable, the Service Manager on other gateway nodes elect another master gateway node.

master service process

An Integration Service process that runs the workflow and workflow tasks. When you run workflows and sessions on a grid, the Integration Service designates one Integration Service process as the master service process. The master service process can distribute the Session, Command, and predefined Event-Wait tasks to worker service processes. The master service process monitors all Integration Service processes.

metadata explosion

The expansion of referenced or multiple-occurring elements in an XML definition. The relationship model you choose for an XML definition determines if metadata is limited or exploded to multiple areas within the definition. Limited data explosion reduces data redundancy.

Metadata Manager Service

An application service that runs the Metadata Manager application in a PowerCenter domain. It manages access to metadata in the Metadata Manager warehouse.

metric-based dispatch mode

A dispatch mode in which the Load Balancer checks current computing load against the resource provision thresholds and then dispatches tasks in a round-robin fashion to nodes where the thresholds are not exceeded.

mixed-version domain

A PowerCenter domain that supports multiple versions of application services.

native authentication

One of the authentication methods used to authenticate users logging in to PowerCenter applications. In native authentication, you create and manage users and groups in the Administrator tool. The Service Manager stores group and user account information and performs authentication in the domain configuration database.

The authentication method used to authenticate users logging in to PowerCenter Express applications. The Service Manager stores group and user account information and performs authentication in the domain configuration database.

node

A logical representation of a machine or a blade. Each node runs a Service Manager that performs domain operations on that node.

node diagnostics

Diagnostic information for a node that you generate in the Administrator tool and upload to Configuration Support Manager. You can use this information to identify issues within your Informatica environment.

nonce

A random value that can be used only once. In PowerCenter web services, a nonce value is used to generate a digested password. If the protected web service uses a digested password, the nonce value must be included in the security header of the SOAP message request.

normalized view

An XML view that contains no more than one multiple-occurring element. Normalized XML views reduce data redundancy.

normal mode

An operating mode for an Integration Service or Repository Service. Run the Integration Service in normal mode during daily Integration Service operations. Run the Repository Service in normal mode to allow multiple users to access the repository and update content.

object query

A user-defined object you use to search for versioned objects that meet specific conditions.

one-way mapping

A mapping that uses a web service client for the source. The Integration Service loads data to a target, often triggered by a real-time event through a web service request.

open transaction

A set of rows that are not bound by commit or rollback rows.

operating mode

The mode for an Integration Service or Repository Service. An Integration Service runs in normal or safe mode. A Repository Service runs in normal or exclusive mode.

operating system flush

A process that the Integration Service uses to flush messages that are in the operating system buffer to the recovery file. The operating system flush ensures that all messages are stored in the recovery file even if the Integration Service is not able to write the recovery information to the recovery file during message processing.

operating system profile

A type of security that the PowerCenter Integration Services on UNIX or Linux uses to isolate the run-time user environment. The operating system profile contains the operating system user name, service process variables, environment variables, and permissions. The PowerCenter Integration Service runs workflows with the system permissions of the operating system user and the properties defined in the operating system profile.

output group

A set of ports that defines a row of outgoing data.

parent dependency

A dependent relationship between two objects in which the parent object uses the child object.

parent object

An object that uses a dependent object, the child object.

permission

The level of access a user has to an object. Even if a user has the privilege to perform certain actions, the user may also require permission to perform the action on a particular object.

pipeline branch

A segment of a pipeline between any two mapping objects.

pipeline stage

The section of a pipeline executed between any two partition points.

pmdtm process

The Data Transformation Manager process.

pmserver process

The Integration Service process.

port dependency

The relationship between an output or input/output port and one or more input or input/output ports.

predefined parameters and variables

Parameters and variables whose values are set by the Integration Service. You cannot define values for predefined parameter and variable values in a parameter file. Predefined parameters and variables include built-in parameters and variables, email variables, and task-specific workflow variables.

predefined resource

An available built-in resource. This can include the operating system or any resource installed by the PowerCenter installation, such as a plug-in or a connection object.

primary node

A node that is configured as the default node to run a service process. By default, the Service Manager starts the service process on the primary node and uses a backup node if the primary node fails.

privilege

An action that a user can perform in PowerCenter applications. You assign privileges to users and groups for the domain and application services.

privilege group

An organization of privileges that defines common user actions.

pushdown compatible connections

Connections with identical property values that allow the Integration Service to identify tables within the same database management system. The required properties depend on the database management system associated with the connection object.

pushdown group

A group of transformations containing transformation logic that is pushed to the database during a session configured for pushdown optimization. The Integration Service creates one or more SQL statements based on the number of partitions in the pipeline.

pushdown optimization

A session option that allows you to push transformation logic to the source or target database.

random masking

A type of masking that produces random, non-repeatable results.

real-time data

Data that originates from a real-time source. Real-time data includes messages and messages queues, web services messages, and change data from a PowerExchange change data capture source.

real-time processing

On-demand processing of data from operational data sources, databases, and data warehouses. Real-time processing reads, processes, and writes data to targets continuously.

real-time session

A session in which the Integration Service generates a real-time flush based on the flush latency configuration and all transformations propagate the flush to the targets.

real-time source

The origin of real-time data. Real-time sources include JMS, WebSphere MQ, TIBCO, webMethods, MSMQ, SAP, and web services.

recovery

The automatic or manual completion of tasks after a service is interrupted. Automatic recovery is available for Integration Service and Repository Service tasks. You can also manually recover Integration Service workflows.

recovery ignore list

A list of message IDs that the Integration Service uses to prevent data duplication for JMS and WebSphere MQ sessions. The Integration Service writes recovery information to the list if there is a chance that the source did not receive an acknowledgement.

reference file

A Microsoft Excel or flat file that contains reference data. Use the Reference Table Manager to import data from reference files into reference tables.

reference table

A table that contains reference data such as default, valid, and cross-reference values. You can create, edit, import, and export reference data with the Reference Table Manager.

reference table staging area

A relational database that stores the reference tables. All reference tables that you create or import using the Reference Table Manager are stored within the staging area. You can create and manage multiple staging areas to restrict access to the reference tables.

repeatable data

A source or transformation output that is in the same order between session runs when the order of the input data is consistent.

repository client

Any PowerCenter component that connects to the repository. This includes the PowerCenter Client, Integration Service, *pmcmd*, *pmrep*, and MX SDK.

repository domain

A group of linked repositories consisting of one global repository and one or more local repositories.

Repository Service

An application service that manages the PowerCenter repository. It retrieves, inserts, and updates metadata in the repository database tables.

request-response mapping

A mapping that uses a web service source and target. When you create a request-response mapping, you use source and target definitions imported from the same WSDL file.

required resource

A PowerCenter resource that is required to run a task. A task fails if it cannot find any node where the required resource is available.

reserved words file

A file named *reswords.txt* that you create and maintain in the Integration Service installation directory. The Integration Service searches this file and places quotes around reserved words when it executes SQL against source, target, and lookup databases.

resilience

The ability for PowerCenter services to tolerate transient network failures until either the resilience timeout expires or the external system failure is fixed.

resilience timeout

The amount of time a client attempts to connect or reconnect to a service. A limit on resilience timeout can override the resilience timeout.

resource provision thresholds

Computing thresholds defined for a node that determine whether the Load Balancer can dispatch tasks to the node. The Load Balancer checks different thresholds depending on the dispatch mode.

role

A collection of privileges that you assign to a user or group. You assign roles to users and groups for the domain and application services.

round-robin dispatch mode

A dispatch mode in which the Load Balancer dispatches tasks to available nodes in a round-robin fashion up to the Maximum Processes resource provision threshold.

Run-time Objects privilege group

A group of privileges that define user actions on the following repository objects: session configuration objects, tasks, workflows, and worklets.

safe mode

An operating mode for the Integration Service. When you run the Integration Service in safe mode, only users with privilege to administer the Integration Service can run and get information about sessions and workflows. A subset of the high availability features are available in safe mode.

SAP BW Service

An application service that listens for RFC requests from SAP NetWeaver BI and initiates workflows to extract from or load to SAP NetWeaver BI.

security domain

A collection of user accounts and groups in a PowerCenter domain. Native authentication uses the Native security domain which contains the users and groups created and managed in the Administrator tool. LDAP authentication uses LDAP security domains which contain users and groups imported from the LDAP directory service. You can define multiple security domains for LDAP authentication.

seed

A random number required by key masking to generate non-colliding repeatable masked output.

sequential merge

A merge type that allows the Integration Service to create one merge file for all target partitions from the individual output files. The integration Service writes the merge file to the final target.

service level

A domain property that establishes priority among tasks that are waiting to be dispatched. When multiple tasks are waiting in the dispatch queue, the Load Balancer checks the service level of the associated workflow so that it dispatches high priority tasks before low priority tasks.

Service Manager

A service that manages all domain operations. It runs on all nodes in the domain to support the application services and the domain. When you start Informatica Services, you start the Service Manager. If the Service Manager is not running, the node is not available.

service mapping

A mapping that processes web service requests. A service mapping can contain source or target definitions imported from a Web Services Description Language (WSDL) file containing a web service operation. It can also contain flat file or XML source or target definitions.

service process

A run-time representation of a service running on a node.

service version

The version of an application service running in the PowerCenter domain. In a mixed-version domain you can create application services of multiple service versions.

service workflow

A workflow that contains exactly one web service input message source and at most one type of web service output message target. Configure service properties in the service workflow.

session

A task in a workflow that tells the Integration Service how to move data from sources to targets. A session corresponds to one mapping.

session recovery

The process that the Integration Service uses to complete failed sessions. When the Integration Service runs a recovery session that writes to a relational target in normal mode, it resumes writing to the target database table at the point at which the previous session failed. For other target types, the Integration Service performs the entire writer run again.

single-version domains

A PowerCenter domain that supports one version of application services.

source pipeline

A source qualifier and all of the transformations and target instances that receive data from that source qualifier.

Sources and Targets privilege group

A group of privileges that define user actions on the following repository objects: cubes, dimensions, source definitions, and target definitions.

state of operation

Workflow and session information the Integration Service stores in a shared location for recovery. The state of operation includes task status, workflow variable values, and processing checkpoints.

static deployment group

A deployment group that you must manually add objects to.

system-defined role

A role that you cannot edit or delete. The Administrator role is a system-defined role.

target connection group

A group of targets that the Integration Service uses to determine commits and loading. When the Integration Service performs a database transaction such as a commit, it performs the transaction for all targets in a target connection group.

target load order groups

A collection of source qualifiers, transformations, and targets linked together in a mapping.

task release

A process that the Workflow Monitor uses to remove older tasks from memory so you can monitor an Integration Service in online mode without exceeding memory limits.

task-specific workflow variables

Predefined workflow variables that vary according to task type. They appear under the task name in the Workflow Manager Expression Editor. `$Dec_TaskStatus.PrevTaskStatus` and `$s_MySession.ErrorMessage` are examples of task-specific workflow variables.

team-based development

The collaboration of team members on a development project. Collaboration includes functionality such as versioning through checking out and checking in repository objects.

terminating condition

A condition that determines when the Integration Service stops reading messages from a real-time source and ends the session.

transaction

A set of rows bound by commit or rollback rows.

transaction boundary

A row, such as a commit or rollback row, that defines the rows in a transaction. Transaction boundaries originate from transaction control points.

transaction control

The ability to define commit and rollback points through an expression in the Transaction Control transformation and session properties.

transaction control point

A transformation that defines or redefines the transaction boundary by dropping any incoming transaction boundary and generating new transaction boundaries.

Transaction Control transformation

A transformation used to define conditions to commit and rollback transactions from relational, XML, and dynamic WebSphere MQ targets.

transaction control unit

A group of targets connected to an active source that generates commits or an effective transaction generator. A transaction control unit may contain multiple target connection groups.

transaction generator

A transformation that generates both commit and rollback rows. Transaction generators drop incoming transaction boundaries and generate new transaction boundaries downstream. Transaction generators are Transaction Control transformations and Custom transformation configured to generate commits.

transformation

A repository object in a mapping that generates, modifies, or passes data. Each transformation performs a different function.

type view

A view created in a web service source or target definition for a complex type element in the input or output message. The type view has an *n:1* relationship with the envelope view.

user credential

Web service security option that requires a client application to log in to the PowerCenter repository and get a session ID. The Web Services Hub authenticates the client requests based on the session ID. This is the security option used for batch web services.

user-defined commit

A commit strategy that the Integration Service uses to commit and roll back transactions defined in a Transaction Control transformation or a Custom transformation configured to generate commits.

user-defined parameters and variables

Parameters and variables whose values can be set by the user. You normally define user-defined parameters and variables in a parameter file. However, some user-defined variables, such as user-defined workflow variables, can have initial values that you specify or are determined by their datatypes. You must specify values for user-defined session parameters.

user-defined property

A user-defined property is metadata that you define, such as PowerCenter metadata extensions. You can create user-defined properties in business intelligence, data modeling, or OLAP tools, such as IBM DB2 Cube Views or PowerCenter, and exchange the metadata between tools using the Metadata Export Wizard and Metadata Import Wizard.

user-defined resource

A PowerCenter resource that you define, such as a file directory or a shared library you want to make available to services.

user name token

Web service security option that requires a client application to provide a user name and password. The Web Services Hub authenticates the client requests based on the user name and password. The password can be a plain text password, hashed password, or digested password. This is the default security option for protected web services.

version

An incremental change of an object saved in the repository. The repository uses version numbers to differentiate versions.

versioned object

An object for which you can create multiple versions in a repository. The repository must be enabled for version control.

view root

The element in an XML view that is a parent to all the other elements in the view.

view row

The column in an XML view that triggers the Integration Service to generate a row of data for the view in a session.

Web Services Hub

An application service in the PowerCenter domain that uses the SOAP standard to receive requests and send responses to web service clients. It acts as a web service gateway to provide client applications access to PowerCenter functionality using web service standards and protocols.

Web Services Provider

The provider entity of the PowerCenter web service framework that makes PowerCenter workflows and data integration functionality accessible to external clients through web services.

worker node

Any node not configured to serve as a gateway. A worker node can run application services but cannot serve as a master gateway node.

workflow

A set of instructions that tells the Integration Service how to run tasks such as sessions, email notifications, and shell commands.

workflow instance

The representation of a workflow. You can choose to run one or more workflow instances associated with a concurrent workflow. When you run a concurrent workflow, you can run one instance multiple times concurrently, or you can run multiple instances concurrently.

workflow instance name

The name of a workflow instance for a concurrent workflow. You can create workflow instance names, or you can use the default instance name.

workflow run ID

A number that identifies a workflow instance that has run.

Workflow Wizard

A wizard that creates a workflow with a Start task and sequential Session tasks based on the mappings you choose.

worklet

A worklet is an object representing a set of tasks created to reuse a set of workflow logic in multiple workflows.

XML group

A set of ports in an XML definition that defines a row of incoming or outgoing data. An XML view becomes a group in a PowerCenter definition.

XML view

A portion of any arbitrary hierarchy in an XML definition. An XML view contains columns that are references to the elements and attributes in the hierarchy. In a web service definition, the XML view represents the elements and attributes defined in the input and output messages.

INDEX

A

Administrator tool
 Domain page [15](#)
 Security page [16](#)
Administrator Tool
 overview [15](#)

D

Domain page
 overview [15](#)

I

Informatica domains
 overview [13](#)
Informix
 database platform [36](#)
Integration Service
 overview [23](#)
Introduction
 overview [10](#)

M

Metadata Manager
 overview [24](#)

P

PowerCenter Client
 overview [17](#)

PowerCenter repository
 overview [15](#)

S

Security tab
 overview [16](#)
sessions
 creating [50, 71](#)
source
 viewing definitions [41](#)
sources
 supported [12](#)

T

targets
 supported [13](#)
transformations
 definition [59](#)

W

Web Services Hub
 overview [23](#)
Workflow Manager
 overview [20](#)
Workflow Monitor
 overview [20, 21](#)
workflows
 running [86](#)