



Informatica® Data Services
10.1.1

Getting Started Guide

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/org/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.openmxdm.org/#FAQ>, http://www.php.net/license/3_01.txt, <http://www.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/tinkerpops/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 LLC 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: 2018-07-19

Table of Contents

Preface	7
Informatica Resources.	7
Informatica Network.	7
Informatica Knowledge Base.	7
Informatica Documentation.	7
Informatica Product Availability Matrixes.	8
Informatica Velocity.	8
Informatica Marketplace.	8
Informatica Global Customer Support.	8
 Chapter 1: Introduction to Informatica Data Services	 9
Informatica Data Services Overview.	9
Data Services Tasks.	10
Define Business Logic.	10
Analyze the Structure and Quality of Data.	10
Create a Single View of Data.	11
Expose Heterogeneous Data as a Virtual Database.	11
Provide a Web Service.	11
Data Services Example.	12
Informatica Data Services Architecture.	13
Informatica Analyst.	14
Using Informatica Analyst.	15
Informatica Analyst User Interface.	15
Informatica Developer.	16
Using Informatica Developer.	17
Informatica Developer User Interface.	18
Informatica Administrator.	19
Using Informatica Administrator.	19
Informatica Administrator User Interface.	20
 Chapter 2: Introduction to the Tutorial.....	 21
Tutorial Overview.	21
Tutorial Lessons.	22
Tutorial Files.	23
Tutorial Prerequisites.	24
 Chapter 3: Setting Up Informatica Developer.....	 25
Setting Up Informatica Developer Overview.	25
Step 1. Start Informatica Developer.	26
Step 2. Connect to the Repository.	27

Adding the Domain and Repository.	28
Connecting to the Model Repository.	31
Step 3. Create a Project.	31
Step 4. Create a Folder.	32
Step 5. Select the Default Data Integration Service.	33
Informatica Developer Tips.	34
Chapter 4: Importing Physical Data Objects.....	35
Importing Physical Data Objects Overview.	35
Step 1. Import the Boston_Customers Flat File Data Object.	36
Step 2. Import the LA_Customers Flat File Data Object.	42
Informatica Developer Tips.	43
Chapter 5: Creating a Logical View of Data.....	44
Creating a Logical View of Data Overview.	44
Step 1. Import a Logical Data Object Model.	46
Step 2. Create a Logical Data Object Mapping.	49
Add a Read Mapping to the Logical Data Object.	50
Adding Sources to the Mapping.	51
Adding an Expression Transformation to the Mapping.	53
Adding a Union Transformation to the Mapping.	56
Step 3. Run the Logical Data Object Mapping.	57
Informatica Developer Tips.	58
Chapter 6: Creating a Virtual View of Data.....	60
Creating a Virtual View of Data Overview.	60
Step 1. Create an SQL Data Service.	62
Step 2. Run an SQL Query to Preview Data.	64
Step 3. Create an Application.	65
Step 4. Deploy the Application.	66
Informatica Developer Tips.	67
Chapter 7: Creating a Web Service.....	69
Creating a Web Service Overview.	69
Step 1. Create a Web Service.	71
Step 2. View and Validate the Operation Mapping.	75
Step 3. Test the Operation Mapping.	76
Step 4. Create an Application.	77
Step 5. Deploy the Application.	78
Informatica Developer Tips.	79
Chapter 8: Exporting a Mapping to PowerCenter.....	80
Exporting a Mapping to PowerCenter Overview.	80

Step 1. Export a Mapping to PowerCenter.	81
Informatica Developer Tips.	84
Appendix A: Glossary.	85

Preface

The *Informatica Data Services Getting Started Guide* is written for data services developers. It provides a tutorial to help first-time users learn how to use Informatica Developer for data services tasks. This guide assumes that you have an understanding of flat file concepts, relational database concepts, web services concepts, and the database engines in your environment.

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-matrices>.

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

Introduction to Informatica Data Services

This chapter includes the following topics:

- [Informatica Data Services Overview, 9](#)
- [Data Services Tasks, 10](#)
- [Data Services Example, 12](#)
- [Informatica Data Services Architecture, 13](#)
- [Informatica Analyst, 14](#)
- [Informatica Developer, 16](#)
- [Informatica Administrator, 19](#)

Informatica Data Services Overview

Informatica Data Services makes data available through a virtual database or a web service. A virtual database provides access to data from heterogeneous sources without loading data to a physical target. A web services makes data available over the internet. A web service client can connect to a web service to access, transform, and deliver data.

Business analysts and developers can collaborate to determine the business logic used to transform data and make it available in a virtual database. Developers can also evaluate the quality of data, create a model of the data to establish uniformity within the enterprise, and transform the data based on business needs before exposing the data through a virtual database or a web service.

You can use Informatica Data Services to complete the following projects:

- Create a prototype of a data warehouse. Before you create a data warehouse, you can use Informatica Data Services to create a virtual data warehouse. You can verify the data requirements and quickly validate the prototype without the time-consuming task of implementing a physical data warehouse. After you validate the prototype, you can reuse the same logic to create the physical data warehouse in PowerCenter®.
- Create a prototype of changes to a data warehouse. When a data warehouse requires changes, you can create a prototype. You can validate the changes before you make them to a physical data warehouse.
- Create a virtual database that allows you to make more data available quickly to enable accurate reporting.

Data Services Tasks

You can use Informatica Data Services to perform different tasks based on organization requirements.

You can perform the following tasks:

- Define business logic that makes data available in a virtual table.
- Analyze the structure and quality of data.
- Create a single view of data.
- Expose heterogeneous data as a virtual database that end users can query.
- Provide a web service that web service clients can query.

Define Business Logic

Business analysts can create mapping specifications to define the business logic that makes data available in a virtual table. A mapping specification can also describe the transformation of data.

Business analysts and developers can collaborate on the development of mapping specifications. Business analysts can use Informatica Analyst (the Analyst tool) to create mapping specifications. After the mapping specifications are complete, developers can use Informatica Developer (the Developer tool) to export them as virtual tables that analysts and developers can run SQL queries against. Developers can also use the Developer tool to export a mapping specification to a PowerCenter mapping to physically transform and move the source data to a target.

Example

A business analyst and a developer want to collaborate on the development of a virtual table that combines data from two data sources. The business analyst creates a mapping specification based on the two data sources in the Analyst tool. The developer views and edits the business logic in the mapping specification in the Developer tool. The business analyst exports the mapping specification logic as a virtual table for end users to run SQL queries against.

Analyze the Structure and Quality of Data

Developers can run column profiles to analyze the structure and quality of the data to ensure that the data is accurate, valid, and complete. A column profile analyzes the characteristics of columns in a data set, such as character patterns and the frequency of values, null values, and blank values.

Developers can use the Developer tool to create column profiles at different stages in the data integration process. Developers can run a column profile at each stage where they want to analyze the data. They can also create a scorecard on a column profile to periodically review data quality. A scorecard is a graphical representation of the quality measurements in a profile.

Example

A developer wants to combine customer address information from relational tables in two data sources. Before they combine the data, they run a column profile on the State columns in both tables to verify that the tables use the same values for states.

Create a Single View of Data

Developers can create a logical data object model that provides a single view to access data from disparate data sources. A logical data object model describes data in an organization and the relationship between the data.

A logical data object model contains logical data objects and defines relationships between them. Logical data objects represent business entities such as customers, orders, accounts, and products. Each logical data object maps to data in underlying data sources.

Developers can expose a logical data object as a virtual table to allow business analysts to build reports based on the virtual table instead of building reports for each data source. The reports can aggregate information from multiple data sources. Developers can also make a logical data object the source of a web service operation to allow end users to access the logical data object over the Web.

Developers can export the logic used to build a logical data object to a PowerCenter maplet to physically move the data from the data source to a target.

Example

A developer works for an organization that acquires another organization. To create a standardized model of customer information, employee information, and order information between the two organizations, the developer creates a logical data object model. The logical data object model contains logical data objects for customers, employees, and orders. The developer runs column profiles on the logical data objects to verify that the data is consistent across both companies. The developer exposes each logical data object as a virtual table. A business analyst builds business intelligence reports based on the customers, employees, and orders virtual tables.

Expose Heterogeneous Data as a Virtual Database

Developers can expose heterogeneous data as a virtual database that end users can query. End users can run SQL queries against the virtual database as if the data were in the same database.

To expose data as a virtual database, a developer can use the Developer tool to create an SQL data service. An SQL data service is a virtual database that end users can query using third-party client tools, like business intelligence tools. A developer can build an SQL data service based on one or more physical data sources, logical data objects, or both. A developer builds an SQL data service based on a logical data object when they want to pull data from multiple, heterogeneous data sources, such as relational databases and flat files, or from data sources that reside in different locations. They can export a mapping specification to a virtual table. They can also run column profiles on a virtual table in an SQL data service.

Example

A developer wants to add data to business intelligence reports, but some of the data does not exist in the data warehouse. Before they create the logic to physically load the new data into the data warehouse, they use an SQL data service to create a prototype of the changes. They create an SQL data service to combine the additional data with the data in the data warehouse. They build a business intelligence report based on the virtual table in the SQL data service.

Provide a Web Service

A web service provides access to data integration functionality. Provide a web service to enable a web service client to access, transform, or deliver data.

An external application or a Web Service Consumer transformation can connect to a web service as a web service client.

Example

A web service client sends a SOAP request to run a web service operation called `getCustomerOrders`. The web service client passes a customer ID in the request. The web service retrieves the customer and the order information. The web service returns the information to the client in a SOAP response.

Data Services Example

You work for an organization that needs to add data from a flat file to the data warehouse to generate a business intelligence report. You use Informatica Data Services to prototype the changes to the data warehouse before integrating the flat file data into the data warehouse.

You create a logical data object model to prototype the modified data warehouse. You build an SQL data service on the logical data object model to enable a business user to query the data in the logical data object model. The business user runs queries to test the changes and provides feedback. After incorporating feedback from the business user, you use the logic defined for the logical data object model to generate a PowerCenter mapping that integrates the flat file data into the data warehouse.

To prototype the changes to the data warehouse, you perform the following steps:

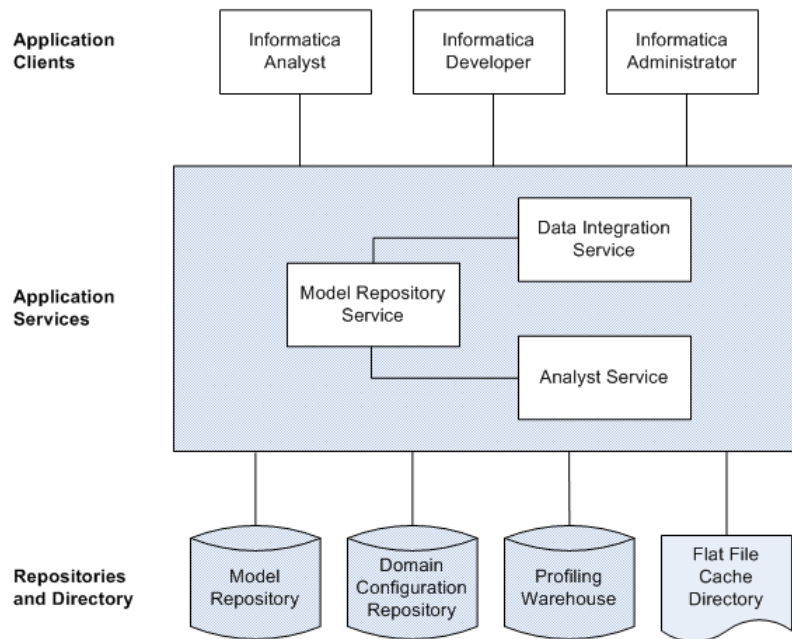
1. Create a flat file data object based on the flat file.
2. Create a relational data object based on the data warehouse table.
3. Create a logical data object model based on the data in the data objects.
4. Create an SQL data service on the logical data object model.
5. Create an application and deploy the SQL data service to the application.
6. From a database client tool, the business user runs SQL queries against the virtual database to verify that it contains all data required for the business intelligence report.

To integrate the flat file data into the data warehouse, you can export the logical data object model as a PowerCenter mapping.

Informatica Data Services Architecture

The Informatica Data Services application clients, application services, and repositories are components that run within the Informatica domain. The Informatica domain is the fundamental administrative unit in Informatica.

The following figure shows the Informatica Data Services components that run within the Informatica domain:



Application Clients

A group of clients that you use to access underlying Informatica functionality. Application clients send requests to the Service Manager or application services. The Service Manager runs the application services and performs domain functions including authentication, authorization, and logging.

The Informatica domain includes the following application clients for Informatica Data Services:

Informatica Analyst

Informatica Analyst (the Analyst tool) is a web-based application client that analysts can use to perform data integration and data federation tasks in an enterprise. Use the Analyst tool to collaborate with developers on data integration and data federation solutions.

Informatica Developer

Informatica Developer (the Developer tool) is an application client that developers can use to design and implement data integration and data federation solutions.

Informatica Administrator

Informatica Administrator (the Administrator tool) is an application client that consolidates the administrative tasks for domain objects such as services, connections, and licenses. Administrators manage the domain and the security of the domain through the Administrator tool.

Application Services

A group of services that represent server-based functionality. An Informatica domain can contain a subset of application services. Administrators configure the application services that are required by the application clients.

The Informatica domain includes the following application services for Informatica Data Services:

Data Integration Service

The Data Integration Service is an application service that runs data integration jobs for the Developer tool and external clients. Data integration jobs include previewing data and running profiles and mappings.

Model Repository Service

The Model Repository Service is an application service that manages the Model repository.

Analyst Service

The Analyst Service is an application service that runs the Analyst tool in the Informatica domain. The Analyst Service manages the connections between service components and the users that have access to the Analyst tool.

Repositories and Directory

Repositories are a group of relational databases that store metadata about objects and processes required to handle user requests from application clients. The directory stores flat files that can be used as sources in the Analyst tool and the Developer tool.

The Informatica domain includes the following databases and directory for Informatica Data Services:

Model repository

The Model repository is a relational database that stores the metadata for projects. The Model repository also stores run-time and configuration information for applications that are deployed to a Data Integration Service.

Domain configuration repository

The domain configuration repository is a set of domain metadata tables stored in a relational database. Each time an administrator makes a change to the domain, the Service Manager writes the change to the domain configuration repository.

Profile warehouse

The profile warehouse is a relational database that the Data Integration Services uses to store profile results.

Flat file cache directory

The flat file cache directory stores flat files that can be used as sources in the Analyst tool. The Analyst Service manages the connection to the flat file cache directory.

Informatica Analyst

Informatica Analyst (the Analyst tool) is a web-based client tool that is available to multiple Informatica products and is used by business users to collaborate on projects within an organization. For example, business analysts can use the Analyst tool to collaborate on data integration projects in an organization.

Use the Analyst tool to define data integration logic and collaborate on projects to accelerate project delivery. Use assets such as flat file data objects, table data objects, or mapping specifications in the Analyst tool to

define the data integration logic for a project. An asset is a type of object that you use to support business operations within the enterprise.

Using Informatica Analyst

Use the Analyst tool to analyze, cleanse, define, and standardize data in an enterprise.

You can use the Analyst tool to complete the following tasks:

Create a data object.

Import metadata to create a flat file data object or table data object. Create flat file or table data objects for sources, lookups, and targets that you want to use in a mapping specification or profile.

Run a profile.

Run a profile to analyze the structure and content of your data, and to determine the quality of your data.

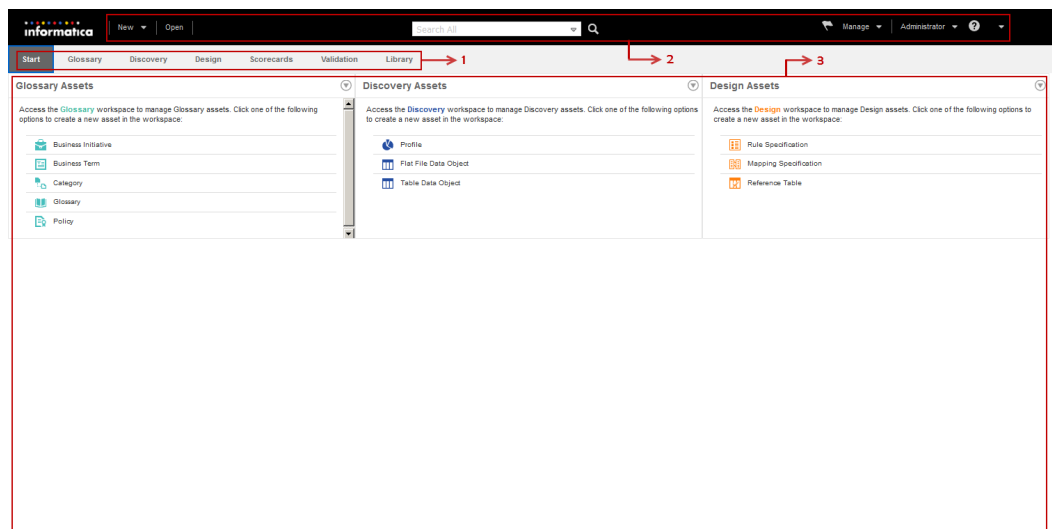
Create a mapping specification.

Use a mapping specification to define business logic that populates a target with data. You can also share the logic or results of the mapping specification with other analysts and developers.

Informatica Analyst User Interface

The Analyst tool user interface consists of multiple workspaces that you use to create data integration and data federation solutions. A workspace is a web page that you can access based on license and privilege.

The following image shows the Analyst tool user interface:



1. Workspace access panel
2. Header area
3. Workspace tabs

The Analyst tool opens on the **Start** workspace. The **Start** workspace lists the workspaces that you have the license and privilege to use through workspace access panels.

The Analyst tool displays the following header menus:

New

Create assets in the **Glossary**, **Discover**, and **Design** workspaces.

Open

Open the **Library** workspace.

Notifications alert

View notifications for Glossary assets.

Manage

Open temporary workspaces and **Notifications**. You can open the **Connections**, **Data Domains**, **Job Status**, **Projects**, and **Business Glossary Security** workspaces.

User name

Set user preferences to change the password and to log out of the Analyst tool.

Help

Access help in the current workspace.

The Analyst tool contains the following permanent workspaces:

Start

Access other workspaces that you have the license to access through the workspace access panels. If you have the license to perform exception management, your tasks appear on the **My Tasks** panel of the workspace.

Glossary

Define and describe business concepts that are important to your organization. You can create and manage business terms, categories, glossaries, and policies.

Discovery

Analyze the quality of data and metadata in source systems. You can create and manage profiles, flat file data objects, and table data objects.

Design

Design business logic that helps analysts and developers collaborate. You can create and manage mapping specifications, reference tables, and rule definitions.

Scorecards

Open, edit, and run scorecards that you created from profile results. You can add metrics, drill down on columns, add scorecard filters, and view trend charts for a scorecard.

Informatica Developer

Informatica Developer (the Developer tool) is an application client that you can use to design and implement data integration and data federation solutions.

You can use the Developer tool to import metadata, create connections, and create logical data objects. You can use the Developer tool to create SQL data services and web services. You can also use the Developer tool to create and run profiles and mappings.

Using Informatica Developer

Use the Developer tool to complete the tasks required to implement data integration and data federation solutions.

You can use the Developer tool to complete the following tasks:

Create a physical data object.

Import metadata to create a physical data object or manually create the physical data object. Create physical data objects for sources, lookups, and targets that you want to use in a mapping, profile, or mapping specification.

Define logical views of data.

A logical view of data describes the structure and use of data in an enterprise. You can create a logical data object model that shows the types of data your enterprise uses and how that data is structured. Then, you can create a mapping that links objects in a logical model to data sources or targets.

Run a column profile.

Run column profiles to discover the number of unique values and null values in each column. You can also run column profiles to view patterns of data in each column and the frequencies with which these values occur.

Create an SQL data service.

An SQL data service is a virtual database that end users can query. Create an SQL data service so that end users can run SQL queries against the virtual tables through a third-party client tool.

Create a web service.

A web service provides access to data integration functionality. Create a web service so that a web service client can connect to a web service to access, transform, or deliver data.

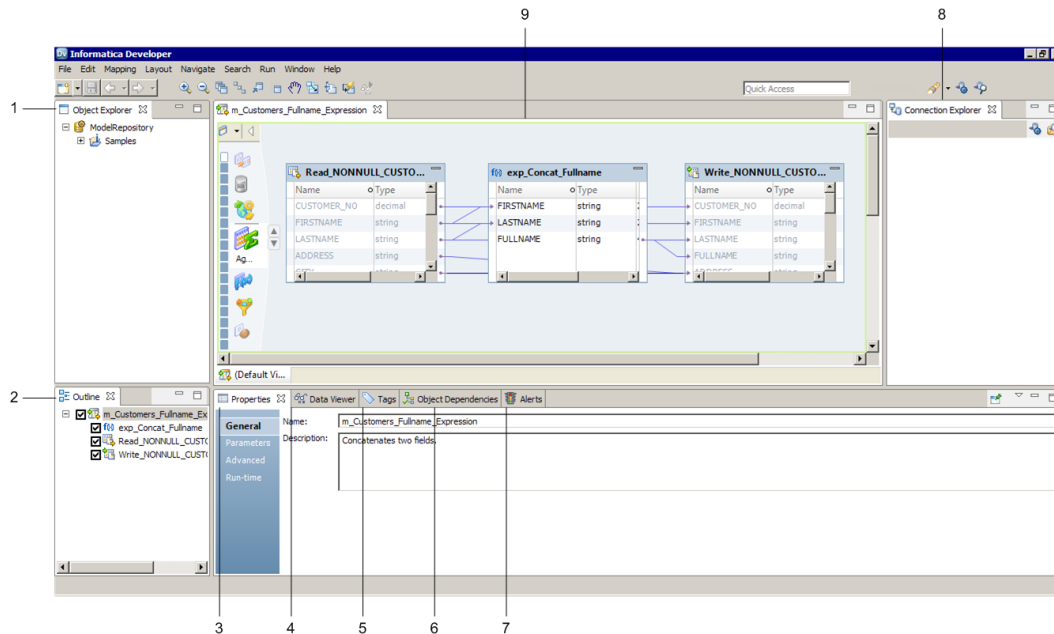
Develop mappings.

Develop mappings to run data integration jobs.

Informatica Developer User Interface

The Developer tool user interface consists of a workbench with multiple views that you use to create data integration and data federation solutions.

The following image shows the Developer tool user interface:



1. Object Explorer view
2. Outline view
3. Properties view
4. Data Viewer view
5. Tags view
6. Object Dependencies view
7. Alerts view
8. Connection Explorer view
9. Editor

The Developer tool can display the following views:

Object Explorer view

Displays projects, folders, and the objects within the projects and folders.

Connection Explorer view

Displays connections to relational databases.

Outline view

Displays objects that are dependent on an object selected in the **Object Explorer** view.

Cheat Sheets view

Displays the cheat sheet that you open. To open a cheat sheet, click **Help > Cheat Sheets** and select a cheat sheet.

Data Viewer view

Displays source data, profile results, and previews the output of a transformation. You can also preview web service messages or run an SQL query from the Data Viewer view.

Object Dependencies view

Displays object dependencies when you view, modify, or delete an object.

Alerts view

Displays connection status alerts.

Validation Log view

Displays object validation errors.

The Developer tool can display other views also. You can hide views and move views to another location in the Developer tool workbench. Click **Window > Show View** to select the views you want to display.

Informatica Administrator

Informatica Administrator (the Administrator tool) is an application client that consolidates the administrative tasks for domain objects such as services, connections, and licenses.

You manage the domain and the security of the domain through the Administrator tool.

Using Informatica Administrator

Informatica Administrator (the Administrator tool) is the tool you use to manage and monitor the Informatica domain and Informatica security.

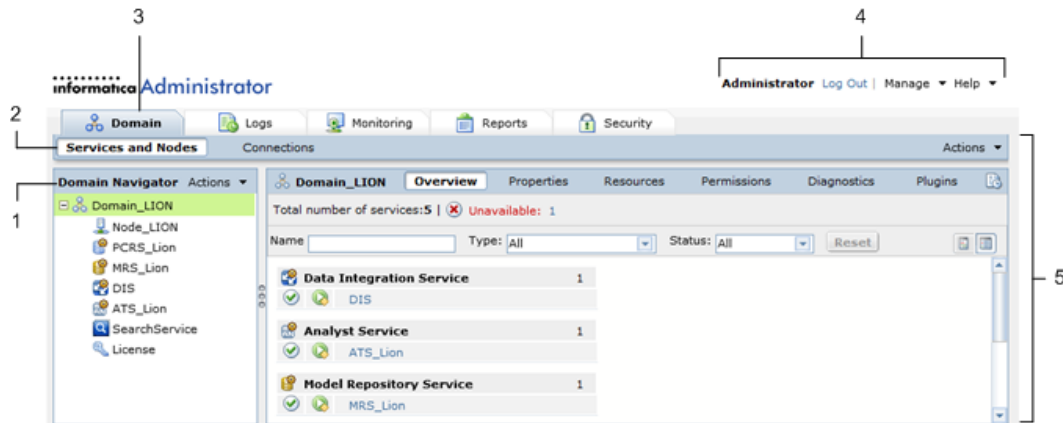
Use the Administrator tool to complete the following types of tasks:

- Domain administrative tasks. Manage logs, domain objects, user permissions, and domain reports. Generate and upload node diagnostics. Monitor Data Integration Service jobs and applications. Domain objects include application services, nodes, grids, folders, database connections, operating system profiles, and licenses.
- Security administrative tasks. Manage users, groups, roles, and privileges.

Informatica Administrator User Interface

The Administrator tool is an application that you use to manage the Informatica domain and the security of the Informatica domain. The Administrator tool interface contains tabs, header items, views, a navigator, and a contents panel.

The following figure shows the Administrator tool:



1. Navigator
2. View in the tab
3. Tab
4. Header area
5. Contents panel

The tabs and views that are available in the Administrator tool differ based on your product license and user permissions. The Navigator displays a hierarchy of objects. The types of objects in the Navigator differ based on the tab that you select. The contents panel displays details about the object that you select in the Navigator.

The Administrator tool has the following tabs:

- **Manage.** View and edit the properties of the domain and objects within the domain.
- **Monitor.** View the status of profile jobs, scorecard jobs, preview jobs, mapping jobs, SQL data services, web services, and workflows for each Data Integration Service.
- **Monitor.** View the status of profile jobs, preview jobs, mapping jobs, SQL data services, and web services for each Data Integration Service.
- **Logs.** View log events for the domain and services within the domain.
- **Reports.** Run a Web Services Report or License Management Report.
- **Security.** Manage users, groups, roles, and privileges.
- **Cloud.** View information about your Informatica Cloud® organization.

The Administrator tool has the following header items:

- **Log out.** Log out of the Administrator tool.
- **Manage.** Manage your account.
- **Help.** Access help for the current tab and determine the Informatica version.

CHAPTER 2

Introduction to the Tutorial

This chapter includes the following topics:

- [Tutorial Overview, 21](#)
- [Tutorial Lessons, 22](#)
- [Tutorial Files, 23](#)
- [Tutorial Prerequisites, 24](#)

Tutorial Overview

The Informatica Data Services tutorial includes multiple lessons. Each lesson contains typical, related tasks that you perform in Informatica Developer.

Each lesson contains the following elements:

Concepts

Each lesson describes the concepts illustrated in the lesson.

Objectives

Each lesson contains objectives that list the tasks included in the lesson and describe the skill level required for the lesson.

Prerequisites

Each lesson lists the prerequisites that you must complete before you start the lesson.

Note: The tutorial also has tutorial prerequisites that you must complete before you start any lesson.

Timing

Each lesson specifies the estimated amount of time required to complete all tasks in the lesson.

Tasks

Each lesson consists of tasks. Perform the tasks in the order specified. The tasks help you understand the lesson concepts and learn how to use the Developer tool to accomplish the tasks.

Each task consists of steps. The steps provide detailed instructions about how to perform the task.

Data Results

Some lessons contain instructions about how to preview the data. Preview the data to verify that the output data looks correct.

Tips

Each lesson provides tips about using the Developer tool. The tips are related to the tasks included in the corresponding lesson.

After completing the tutorial, you should be able to perform the following high-level tasks:

- Connect the Developer tool to a domain and Model repository.
- Select the default Data Integration Service to preview data and run mappings.
- Create folders and projects to organize objects that you create in a Model repository.
- Import physical data objects.
- Create a logical data object read mapping.
- Create and deploy an SQL data service.
- Create and deploy a web service.
- Export a mapping from a Model repository to a PowerCenter repository.

Tutorial Lessons

The tutorial consists of multiple lessons. Some lessons are prerequisites for subsequent lessons.

In this tutorial, you complete the following lessons:

Set up Informatica Developer.

You start and set up the Developer tool. To set up the Developer tool, you connect to the Model repository, and create a project and folder to store your work. You also select the default Data Integration Service to preview data and run mappings.

Import physical data objects.

You import the Boston_Customers.csv and LA_Customers.csv tutorial flat file as physical data objects.

Create a logical view of data.

You import a logical data object model that contains the Customer and Order logical data objects. You also create a logical data object read mapping with the Customer logical data object as the mapping output. You run the mapping to view customer data from multiple sources.

Create a virtual view of data.

You create an SQL data service to define a virtual database that contains customer data. You preview the virtual data. You create an application that contains the SQL data service, and deploy the application to the Data Integration Service.

Create a web service.

You create a web service that provides access to customer data. You view and validate the operation mapping. You create an application that contains the web service, and deploy the application to the Data Integration Service.

Export a mapping from the Model repository to a PowerCenter repository.

You export a logical data object read mapping to a PowerCenter repository.

Tutorial Files

The tutorial files contain sample customer data and a logical data object model XML schema. Some lessons require the tutorial files to build objects.

The tutorial requires the following tutorial files:

- Boston_Customers.csv
- LA_customers.csv
- Customer_Order.xsd

The following table provides the machines and directories where you can find the tutorial files:

Machine	Directory
Machine that runs the Developer tool	<Informatica installation directory>\clients \DeveloperClient\Tutorials
Machine that runs the Data Integration Service	<Informatica installation directory>\server \Tutorials

Boston_Customers.csv tutorial file

The Boston_Customers.csv file contains a list of customers from the Boston region. In a lesson, you import the Boston_Customers.csv file into Informatica Developer to create a physical data object.

The following table shows sample data from the Boston_Customers.csv tutorial file:

CustomerID	Customer Region	CustomerTier	Lastname	Firstname
10110102	Boston	Bronze	Brosseau	Derrick
10110105	Boston		Anderson	Lauren
10110106	Boston	Emerald	Boonstra	Paul

LA_Customers.csv tutorial file

The LA_Customers.csv file contains a list of customers from the Los Angeles region. In a lesson, you import the LA_Customers.csv file into Informatica Developer to create a physical data object.

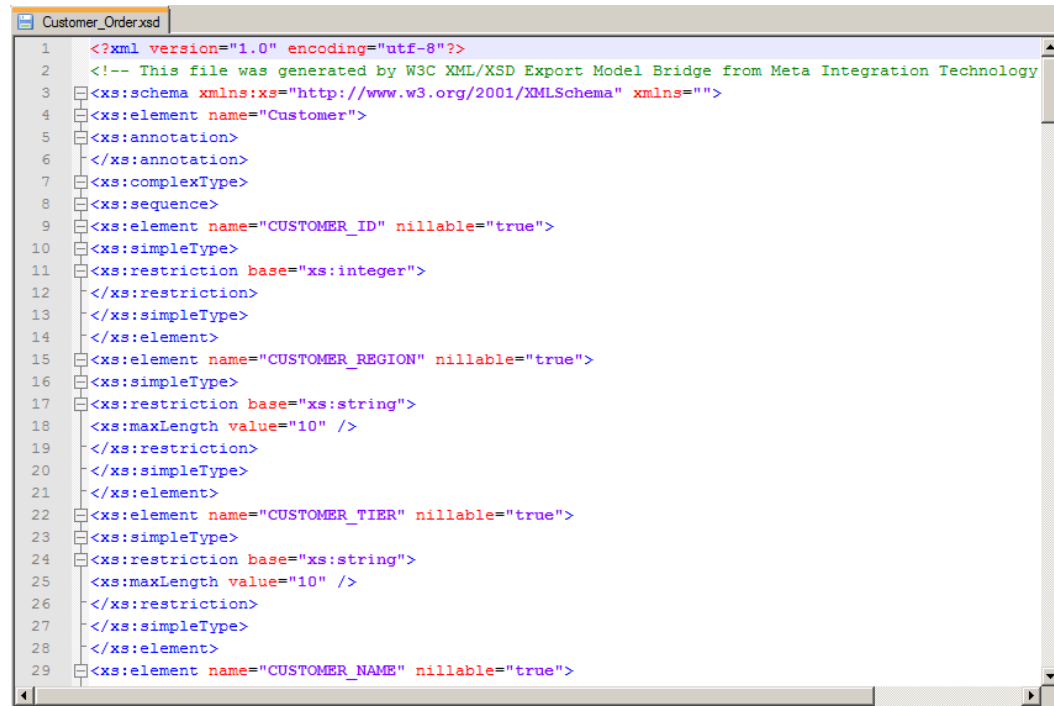
The following table shows sample data from the LA_Customers.csv tutorial file:

CustomerID	Customer Region	CustomerTier	FullName
10110095	Los Angeles	Diamond	Barb Bergeron
10110131	Los Angeles	Diamond	Sandra Tucker
10110147	Los Angeles	2	Robert Blayney

Customer_Order.xsd tutorial file

The Customer_Order.xsd file contains an XML schema definition of the Customer_Order logical data object model. In a lesson, you import the Customer_Order.xsd file into the Developer tool to create a logical data object model.

The following figure shows a sample of the Customer_Order.xsd tutorial file:



Tutorial Prerequisites

Before you can begin any lesson, you must complete the tutorial prerequisites.

Complete the following tutorial prerequisites:

1. Verify that the Informatica domain, Model Repository Service, and Data Integration Service are running.
2. Verify that the tutorial files are in the following directory on the machine that runs the Developer tool:
`<Informatica installation directory>\clients\DeveloperClient\Tutorials`
3. Verify that the tutorial files are in the following directory on the machine that runs the Data Integration Service:
`<Informatica installation directory>\server\Tutorials`

CHAPTER 3

Setting Up Informatica Developer

This chapter includes the following topics:

- [Setting Up Informatica Developer Overview, 25](#)
- [Step 1. Start Informatica Developer, 26](#)
- [Step 2. Connect to the Repository, 27](#)
- [Step 3. Create a Project, 31](#)
- [Step 4. Create a Folder, 32](#)
- [Step 5. Select the Default Data Integration Service, 33](#)
- [Informatica Developer Tips, 34](#)

Setting Up Informatica Developer Overview

In this lesson, you start and set up the Developer tool. You get information from the domain administrator to connect the Developer tool to the Informatica domain.

To set up the Developer tool, you connect to the Model repository and create a project and folder to store your work. You select the default Data Integration Service to preview data and run mappings.

Lesson Concepts

The Informatica domain is a collection of services that perform data integration jobs and monitoring jobs. You manage the domain through the Administrator tool.

The Model Repository Service manages the Model repository. The Model repository is a relational database that stores the metadata for projects and folders. A project stores objects that you create in the Developer tool. A project can also contain folders that store related objects that are part of the same business requirement.

The Data Integration Service performs data integration jobs for the Developer tool. Data integration jobs include previewing data, and running profiles, mappings, and workflows.

Lesson Objectives

In this lesson, you complete the following beginner-level tasks:

- Start the Developer tool.
- Configure the Developer tool to connect to the Model repository.
- Create a project to store the objects that you create in the Developer tool.
- Create a folder in the project that can store related objects.

- Select the default Data Integration Service to preview data and to run mappings in the Developer tool.

Lesson Prerequisites

Before you start this lesson, complete the following prerequisites:

- Ask a domain administrator to verify that the Model Repository Service and Data Integration Service are running in the domain.
- Get the following information from a domain administrator:
 - Domain name, host name, and port number to connect to a domain.
 - User name and password to access the Model repository.
- Install the Informatica clients, which includes the Developer tool.

Lesson Timing

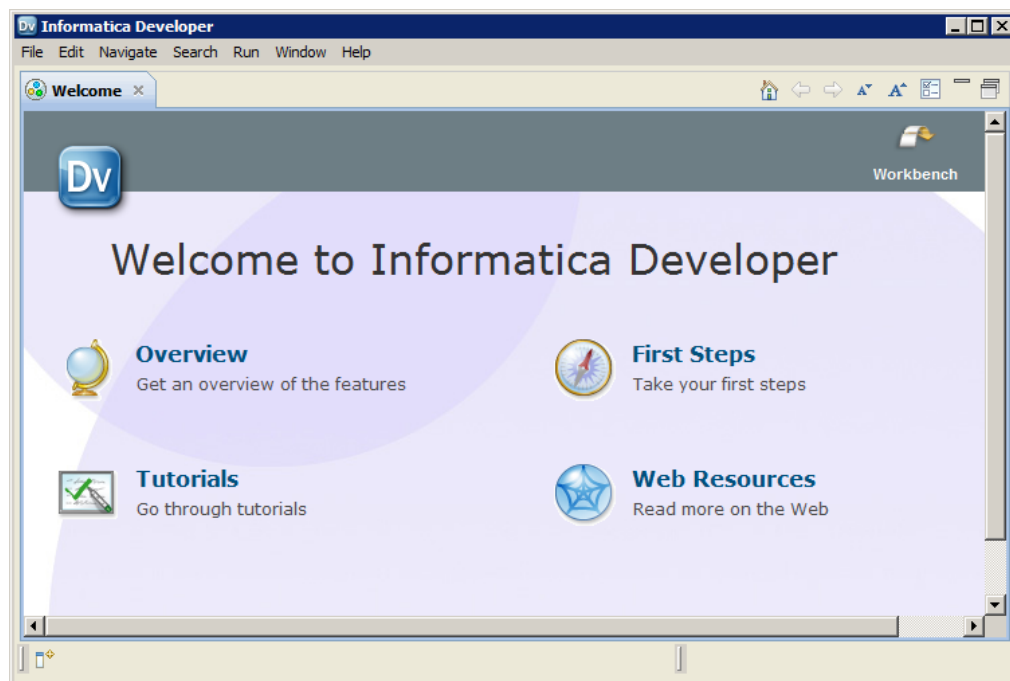
Set aside 5 to 10 minutes to complete the tasks in this lesson.

Step 1. Start Informatica Developer

Start the Developer tool to begin the lesson.

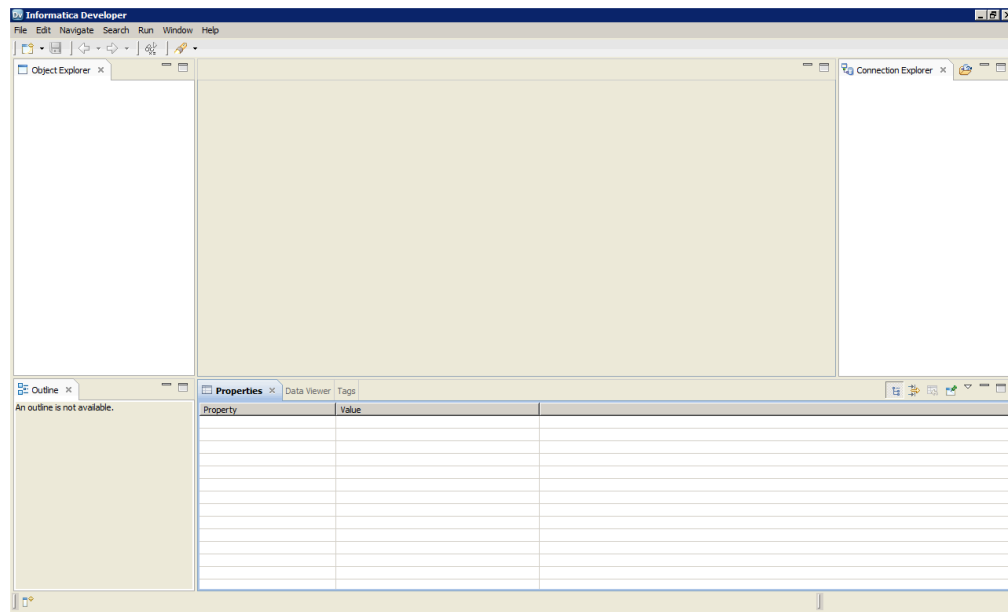
1. From the Windows Start menu, click **Informatica <version> > Client > Developer Client > Launch Informatica Developer**.

The **Welcome** page of the Developer tool appears. If you have started the Developer tool before, the Developer tool opens to the **Workbench**.



2. In the top, right corner, click the **Workbench** link.

The Developer tool **Workbench** appears.



Step 2. Connect to the Repository

Connect to the Model repository to create, view, and manage metadata.

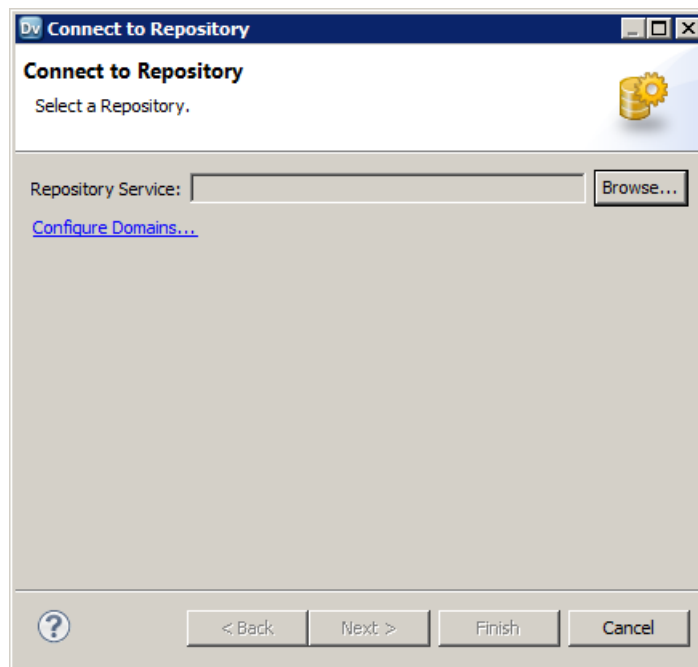
Adding the Domain and Repository

You must add the domain and Model repository the first time that you set up the Developer tool.

When you add the domain, use the domain name, host name, and port number that you got from the domain administrator.

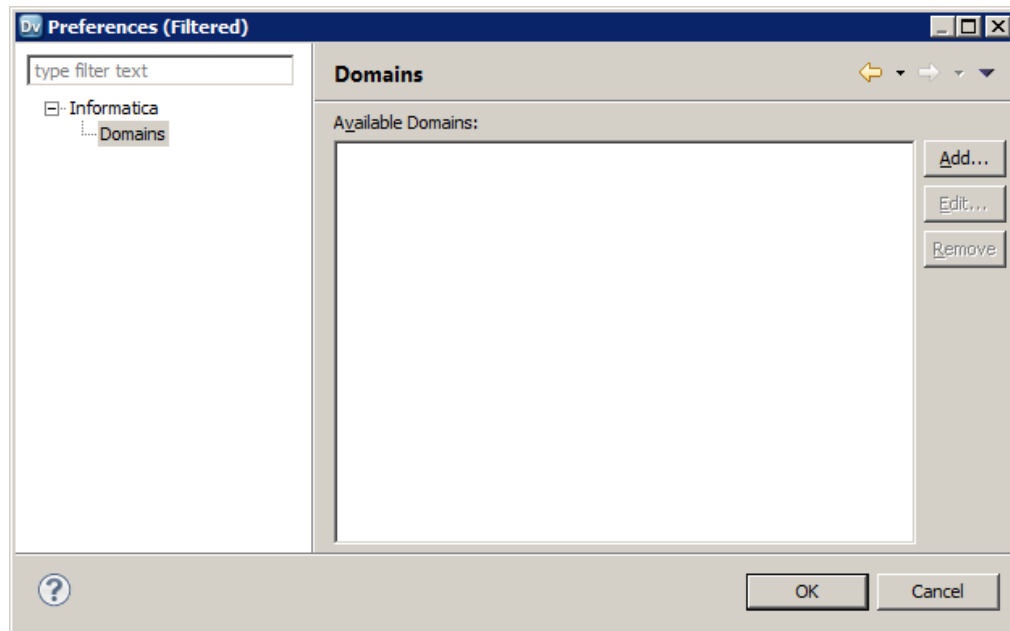
1. From the Developer tool menu, click **File > Connect to Repository**.

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



2. Click **Configure Domains**.

The **Preferences** dialog box appears.



3. Click **Add**.

The **New Domain** dialog box appears.

4. Enter the domain name, host name, and port number for the domain.

The following table lists the default values for the domain:

Property	Default Value	Example
Domain name	Domain_<machine name>	Domain_SalesServer
Host name	<machine name>	SalesServer
Port number	6005	6005

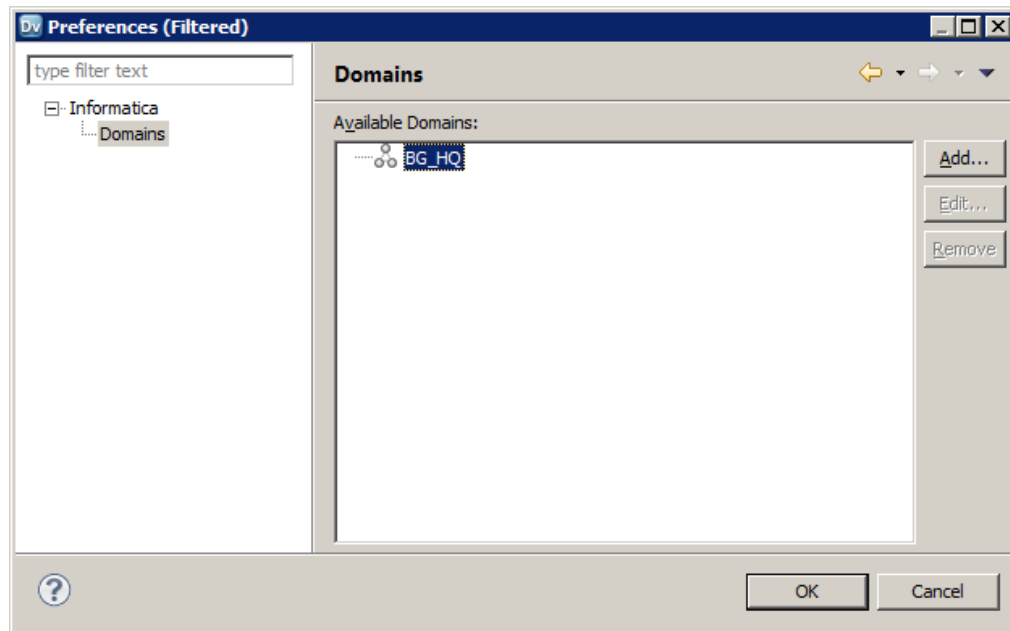
5. Click **Test Connection** to test the connection.

The Developer tool displays a message that states whether the connection is successful.

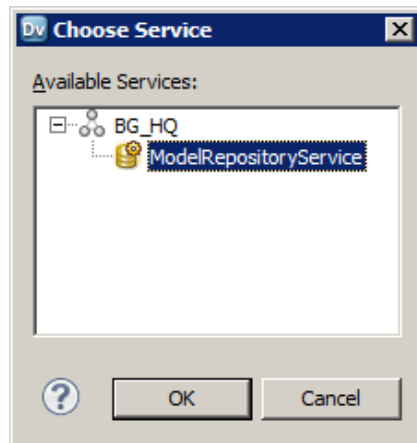
6. Click **OK**.

7. Click **Finish**.

The domain appears in the **Available Domains** panel of the **Preferences** dialog box.



8. Click **OK**.
The **Connect to Repository** dialog box appears.
9. Click **Browse** to select the Model Repository Service associated with the Model repository.
The **Choose Service** dialog box appears.
10. Expand the domain and select the Model Repository Service.
The following figure shows the selected Model Repository Service:



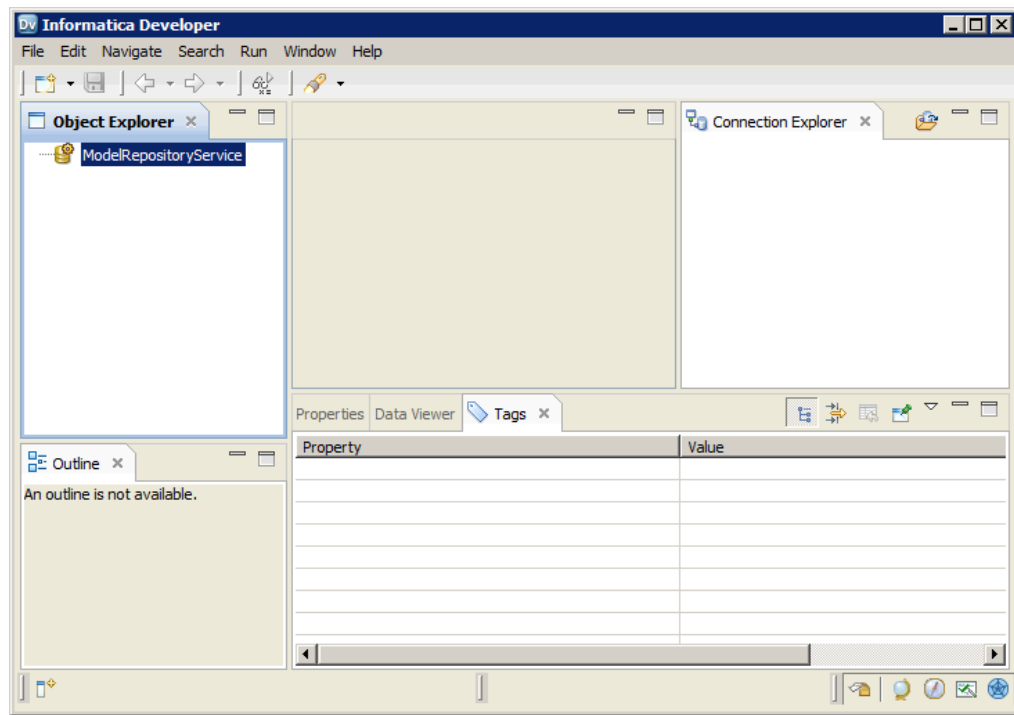
11. Click **OK** and then click **Next**.
The **Login** window appears.
12. Enter the user name and password provided to you.
13. Click **Finish**.
The Developer tool connects to the Model repository.

Connecting to the Model Repository

You can connect to the repository from the **Object Explorer** view.

1. In the **Object Explorer** view, right-click the Model Repository Service associated with the Model repository to which you want to connect.
 2. Select **Connect**.
- The **Connect to Domain** dialog box appears.
3. Enter the user name and password provided to you.
 4. Click **OK**.

The Developer tool connects to the Model repository.



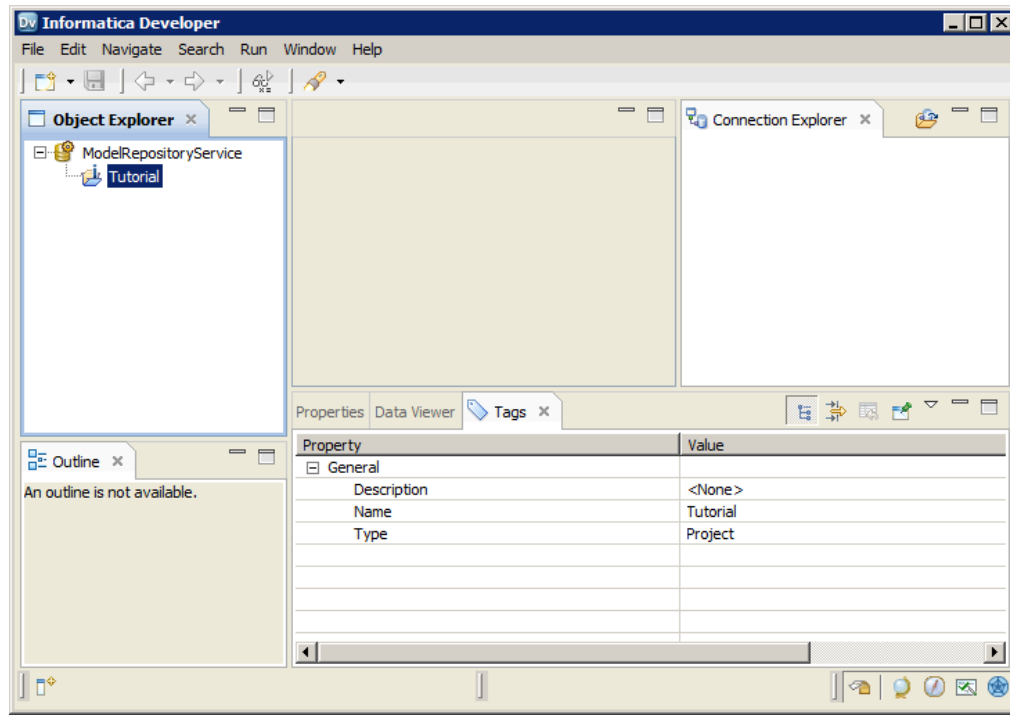
Step 3. Create a Project

Create a project in the Model repository to store objects that you create in the Developer tool. You can create one project for all lessons in this guide.

1. From the Developer tool menu, click **File > New > Project**.
- The **New Project** dialog box appears.
2. Enter Tutorial for the project name.

3. Click **Finish**.

The Developer tool adds the Tutorial project under the Model Repository Service in the **Object Explorer** view.



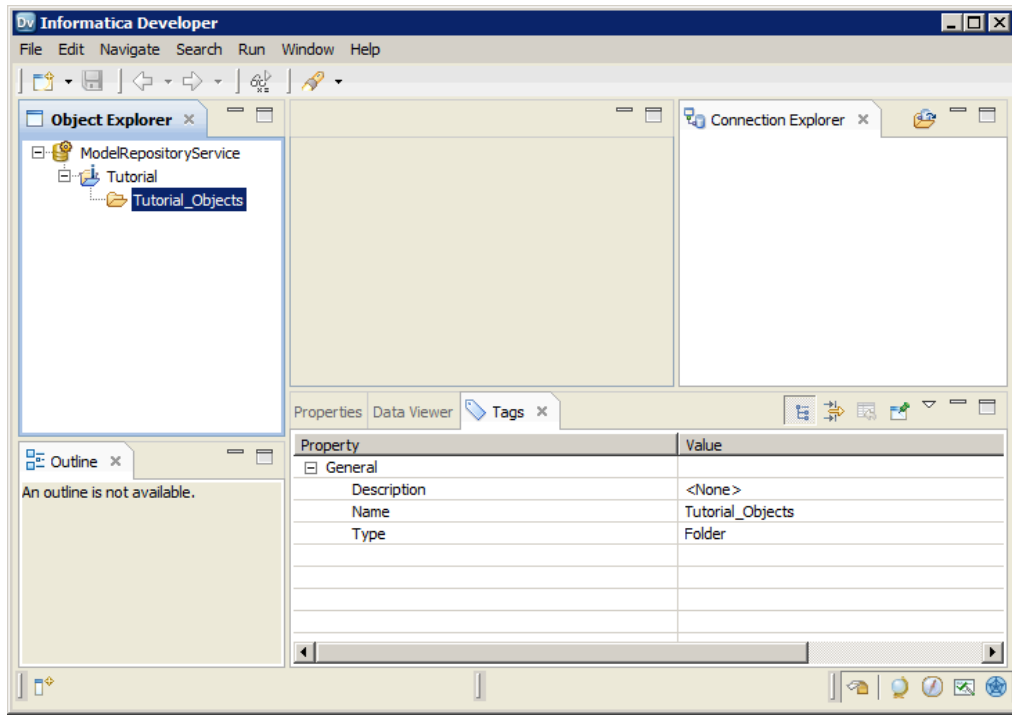
Step 4. Create a Folder

Create a folder to store related objects. You can also store objects in projects. However, folders are useful to organize and store many related objects. You can create one folder for all lessons in this guide.

1. In the **Object Explorer** view, select the Tutorial project.
2. Right-click and select **New > Folder**.
The **New Folder** dialog box appears.
3. Verify that the location indicates the Tutorial project, and enter Tutorial_Objects for the folder name.

4. Click **Finish**.

The Developer tool adds the Tutorial_Objects folder under the Tutorial project in the **Object Explorer** view.

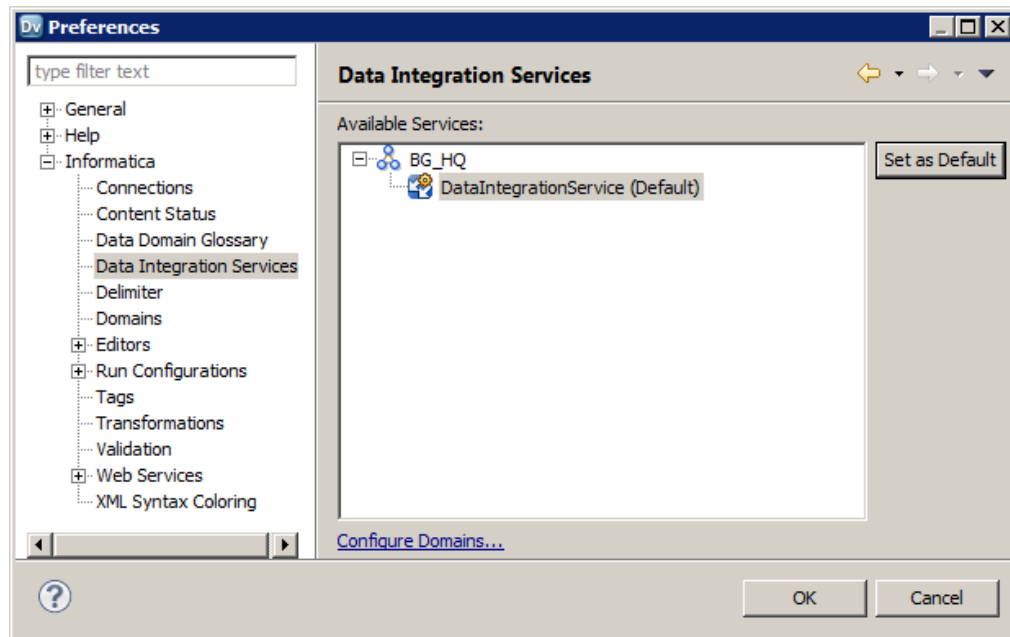


Step 5. Select the Default Data Integration Service

Select the default Data Integration Service that the Developer tool uses to preview data and run mappings.

1. From the Developer tool menu, click **Window > Preferences**.
The **Preferences** dialog box appears.
2. Select **Informatica > Data Integration Services**.
3. Expand the domain.
4. Select the Data Integration Service and click **Set as Default**.

The following figure shows the selected default Data Integration Service:



5. Click **OK**.

Informatica Developer Tips

Use the following tips to help you use the Developer tool more effectively.

Reset the Developer tool perspective to the default values.

If you hide views or move views to another location in the Developer tool workbench, you can reset the Developer tool perspective to the default values. Click **Window > Reset Perspective**.

Use keyboard shortcuts.

You can use keyboard shortcuts to complete tasks in the Developer tool. To view the keyboard shortcuts, click **Help > Key Assist**.

CHAPTER 4

Importing Physical Data Objects

This chapter includes the following topics:

- [Importing Physical Data Objects Overview, 35](#)
- [Step 1. Import the Boston_Customers Flat File Data Object, 36](#)
- [Step 2. Import the LA_Customers Flat File Data Object, 42](#)
- [Informatica Developer Tips, 43](#)

Importing Physical Data Objects Overview

Import the Boston_Customers.csv and LA_Customers.csv tutorial flat files as physical data objects.

Lesson Concepts

A physical data object is a Model repository object that represents a flat file or relational database table. You can import a flat file or relational database table as a physical data object to use as a source, target, or lookup in a mapping.

Lesson Objectives

In this lesson, you complete the following beginner-level tasks:

- Import the Boston_Customers flat file as a physical data object.
- Import the LA_Customers flat file as a physical data object.

Lesson Prerequisites

Before you start this lesson, complete the following prerequisites:

- Set up Informatica Developer. For more information, see [“Setting Up Informatica Developer Overview” on page 25](#).
- Verify that the Boston_Customers.csv and LA_Customers.csv tutorial files are in the following directory on the Developer tool machine:

```
<Informatica installation directory>\clients\DeveloperClient\Tutorials
```

- Verify that the Boston_Customers.csv and LA_Customers.csv tutorial files are also in the following directory on every machine that runs the Data Integration Service:

```
<Informatica installation directory>\server\Tutorials
```

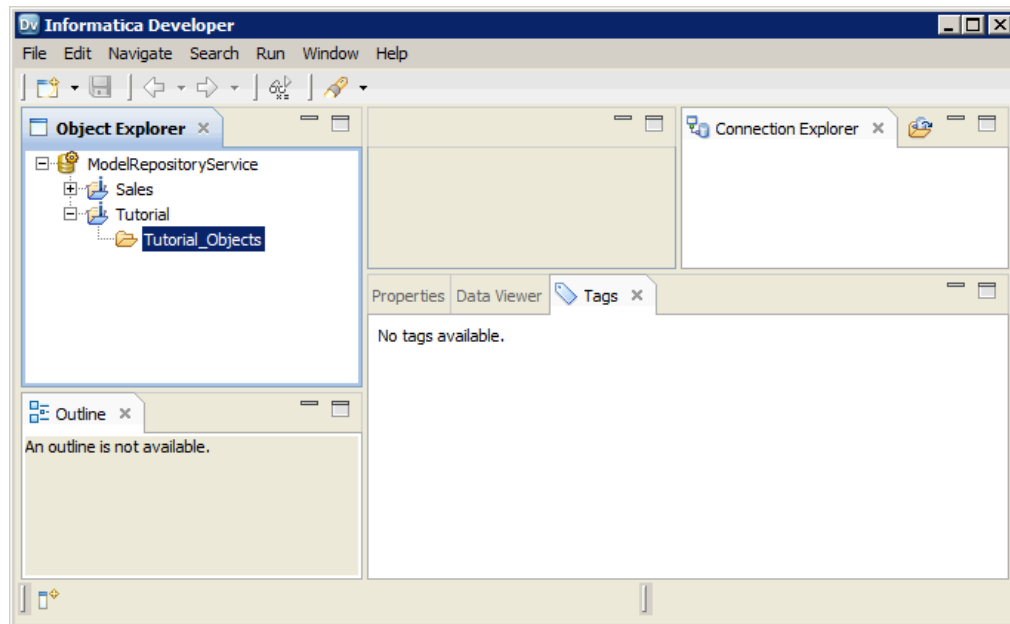
Lesson Timing

Set aside 5 to 10 minutes to complete the tasks in this lesson.

Step 1. Import the Boston_Customers Flat File Data Object

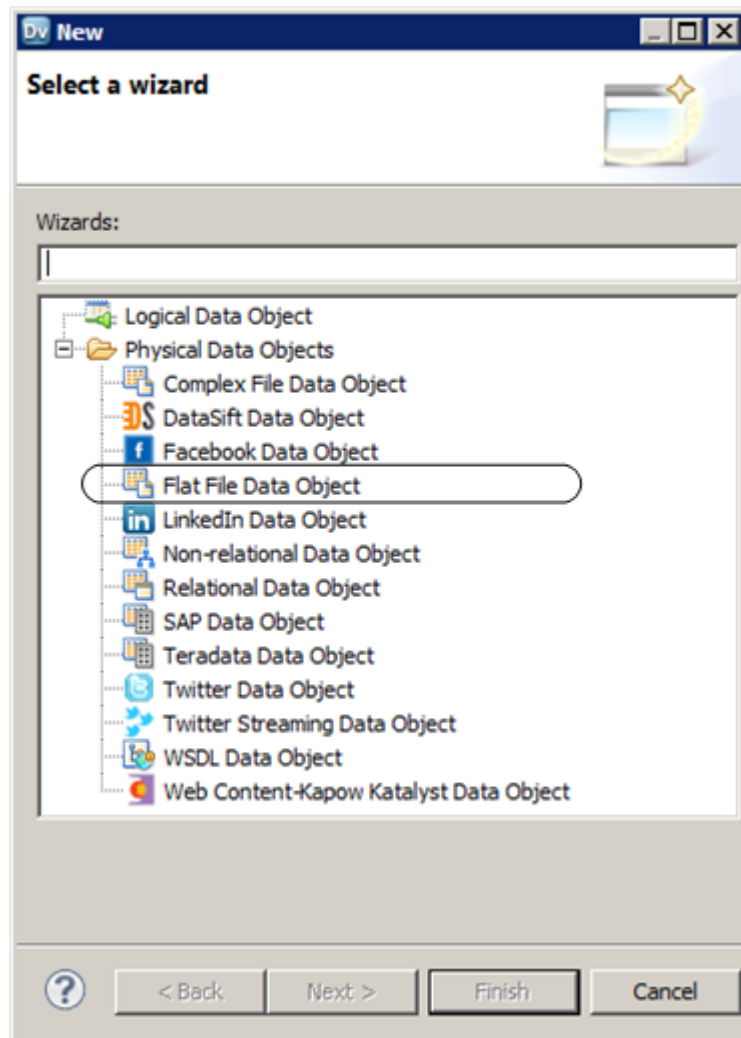
In this task, you import the Boston_Customers.csv flat file as a physical data object. The flat file contains data about customers from a Boston office.

1. In the **Object Explorer** view, select the Tutorial_Objects folder.



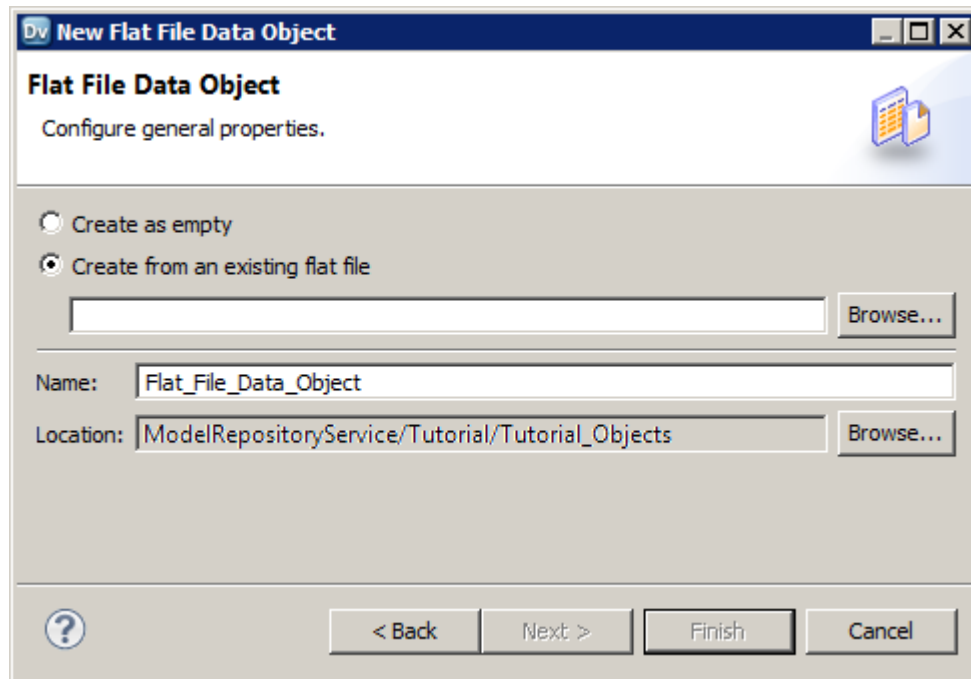
2. Right-click the Tutorial_Objects folder and select **New > Data Object**.

The **New** dialog box appears.



3. Select **Physical Data Objects** > **Flat File Data Object** and click **Next**.

The **New Flat File Data Object** dialog box appears.



4. Select **Create from an existing flat file**.
5. Click **Browse** and navigate to Boston_Customers.csv in the following directory on the Developer tool machine: `<Informatica installation directory>\clients\DeveloperClient\Tutorials`
6. Click **Open**.
The wizard names the data object "Boston_Customers."
7. Click **Next**.
8. Verify that the code page is set to **MS Windows Latin 1 (ANSI), superset of Latin 1** and the format is set to **Delimited**.

The **New Flat File Data Object** dialog box shows the default code page, the format, and a preview of the flat file data.

New Flat File Data Object

Flat File Data Object
Configure code page and format.

Code page: MS Windows Latin 1 (ANSI), superset of Latin 1

Format

☒ Delimited (fields separated by delimiters)

☐ Fixed-width (fields aligned in columns)

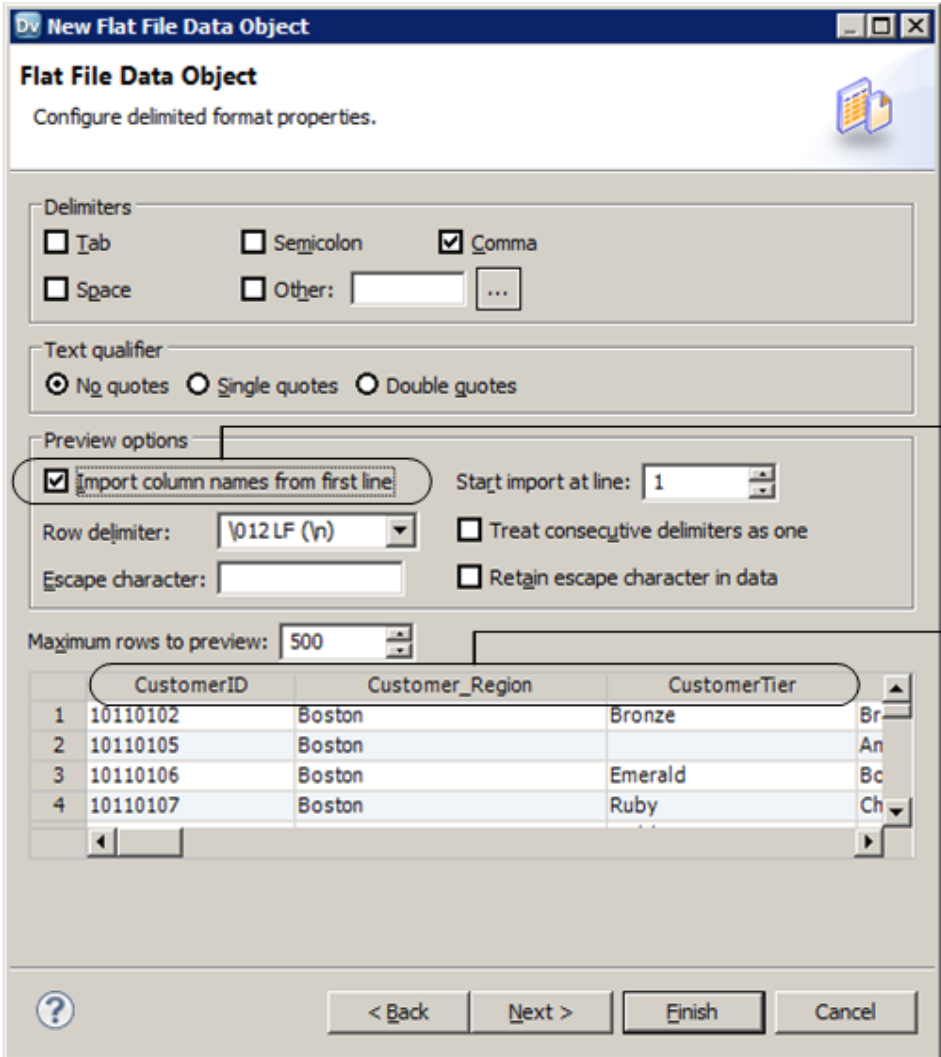
Maximum rows to preview: 500

	Field1	Field2	Field3
1	CustomerID	Customer Region	CustomerTier
2	10110102	Boston	Bronze
3	10110105	Boston	
4	10110106	Boston	Emerald
5	10110107	Boston	Ruby
6	10110109	Boston	Gold
7	10110111	Boston	2
8	10110117	Boston	2

? < Back Next > Finish Cancel

9. Click **Next**.
10. Select **Import column names from first line**.

The **New Flat File Data Object** dialog box shows the column names in the preview of the flat file data.



The dialog box is titled "New Flat File Data Object" and "Flat File Data Object". It contains several sections: "Delimiters" with checkboxes for Tab, Semicolon, Comma (checked), and Space; "Text qualifier" with radio buttons for No quotes (selected), Single quotes, and Double quotes; "Preview options" with a checked checkbox for "Import column names from first line", a "Start import at line" spinner set to 1, a "Row delimiter" dropdown set to "\012 LF (\n)", and checkboxes for "Treat consecutive delimiters as one" and "Retain escape character in data"; and a "Maximum rows to preview" spinner set to 500. Below these is a table preview with columns CustomerID, Customer_Region, and CustomerTier, showing four rows of data. At the bottom are buttons for Back, Next, Finish, and Cancel. Two callout lines point to the "Import column names from first line" checkbox (labeled 1) and the table preview (labeled 2).

1

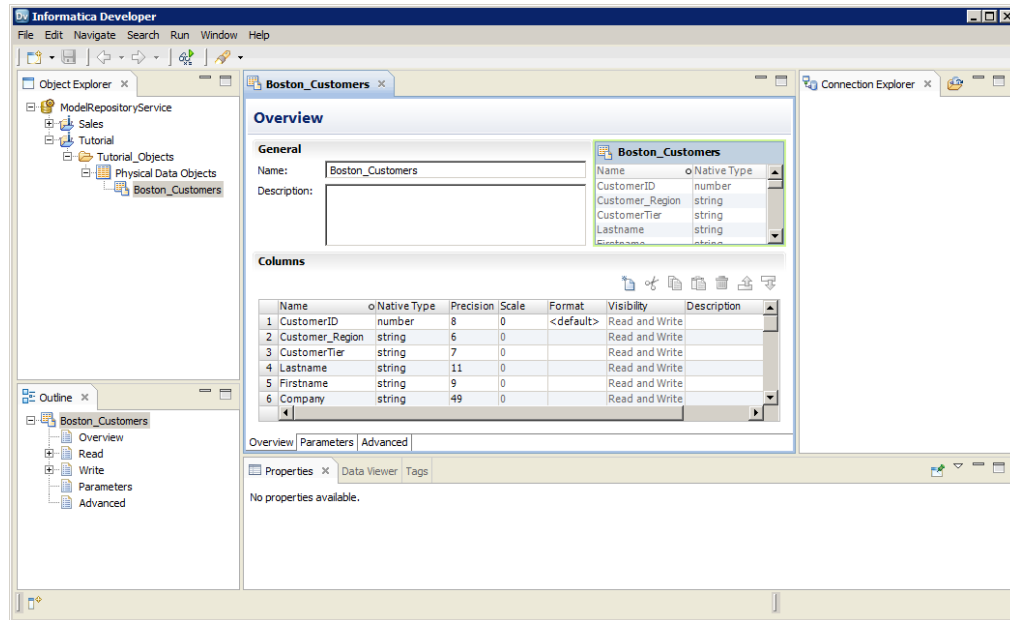
2

	CustomerID	Customer_Region	CustomerTier
1	10110102	Boston	Bronze
2	10110105	Boston	An
3	10110106	Boston	Emerald
4	10110107	Boston	Ruby

1. The **Import column names from first line** option
2. Column names

11. Click **Finish**.

The Boston_Customers physical data object appears under the Physical Data Objects folder in the Tutorial_Objects folder. The **Overview** view displays the file content and it is open in the editor.



12. Click the **Advanced** view.

The **Advanced** view shows properties for the physical data object.

13. In the **Advanced** view, scroll to the **Run-time: Read** section.
14. In the **Run-time: Read** section, set **Source file directory** to the following directory on the Data Integration Service machine: `<Informatica installation directory>\server\Tutorials`

The Data Integration Service searches for the source file in the server directory on the machine that runs the Data Integration Service. The server installation contains a copy of the tutorial files. The Data Integration Service cannot read files from the client installation directory unless you change access permissions on the source file and directory.

The following figure shows a sample source file directory:

Advanced	
Name	Value
Runtime : Read	
Input type	File
Source type	Direct
Source file name	Boston_Customers.csv
Source file directory	\\MyMachine\Informatica\10.0\server\Tutorials

Note: The Developer tool machine must have access to the source file directory on the machine that runs the Data Integration Service. If the Developer tool cannot access the source file directory, the Developer tool cannot preview data in the source file or run mappings that access data in the source file. If you run multiple Data Integration Services, there is a separate source file directory for each Data Integration Service.

15. Click the **Data Viewer** view.
16. In the **Data Viewer** view, click **Run**.

The Data Integration Service reads the data from the Boston_Customers file and shows the results in the Output window.

17. Click **File > Save** to save the Boston_Customers physical data object.

Step 2. Import the LA_Customers Flat File Data Object

In this task, you import the LA_Customers.csv flat file as a physical data object. The flat file contains data about customers from a Los Angeles office.

1. In the **Object Explorer** view, select the Tutorial_Objects folder.
2. Right-click the Tutorial_Objects folder and select **New > Data Object**.
The **New** dialog box appears.
3. Select **Physical Data Objects > Flat File Data Object** and click **Next**.
The **New Flat File Data Object** dialog box appears.
4. Select **Create from an existing flat file**.
5. Click **Browse** and navigate to LA_Customers.csv in the following directory on the Developer tool machine: `<Informatica installation directory>\clients\DeveloperClient\Tutorials`
6. Click **Open**.
The wizard names the data object "LA_Customers."
7. Click **Next**.
8. Verify that the code page is MS Windows Latin 1 (ANSI), superset of Latin 1 and the format is delimited.
9. Click **Next**.
10. Select **Import column names from first line**.
The **New Flat File Data Object** dialog box shows the column names in the preview of the flat file data.
11. Click **Finish**.
The LA_Customers physical data object appears under the Physical Data Objects folder in the Tutorial_Objects folder. The **Overview** view displays the file content and it is open in the editor.
12. Click the **Advanced** view.
The **Advanced** view shows properties for the physical data object.
13. In the **Advanced** view, scroll to the **Run-time: Read** section.
14. In the **Run-time: Read** section, set **Source file directory** to the following directory on the Data Integration Service machine: `<Informatica installation directory>\server\Tutorials`
When processing the data, the Data Integration Service searches for the source file in the server directory on the machine that runs the Data Integration Service. The Data Integration Service cannot read the tutorial files from the client installation directory unless you change the access permissions for the source file and directory.
15. Click the **Data Viewer** view.
16. In the **Data Viewer** view, click **Run**.
The Data Integration Service reads the data from the LA_Customers file and shows the results in the Output window.

17. Click **File** > **Save** to save the LA_Customers physical data object.

Informatica Developer Tips

Use the following tip when you review source data.

Export data.

You can export the data that displays in the **Data Viewer** view to a tab-delimited flat file, such as a TXT or CSV file. Export data when you want to create a local copy of the data. To export the data, right-click a row of data in the **Data Viewer** view and select **Export Data**.

CHAPTER 5

Creating a Logical View of Data

This chapter includes the following topics:

- [Creating a Logical View of Data Overview, 44](#)
- [Step 1. Import a Logical Data Object Model, 46](#)
- [Step 2. Create a Logical Data Object Mapping, 49](#)
- [Step 3. Run the Logical Data Object Mapping, 57](#)
- [Informatica Developer Tips, 58](#)

Creating a Logical View of Data Overview

Create a logical view of data to provide a single view of physical data that might come from multiple sources and different source types, such as flat files and databases.

Lesson Concepts

A logical view of data is a data model in an enterprise.

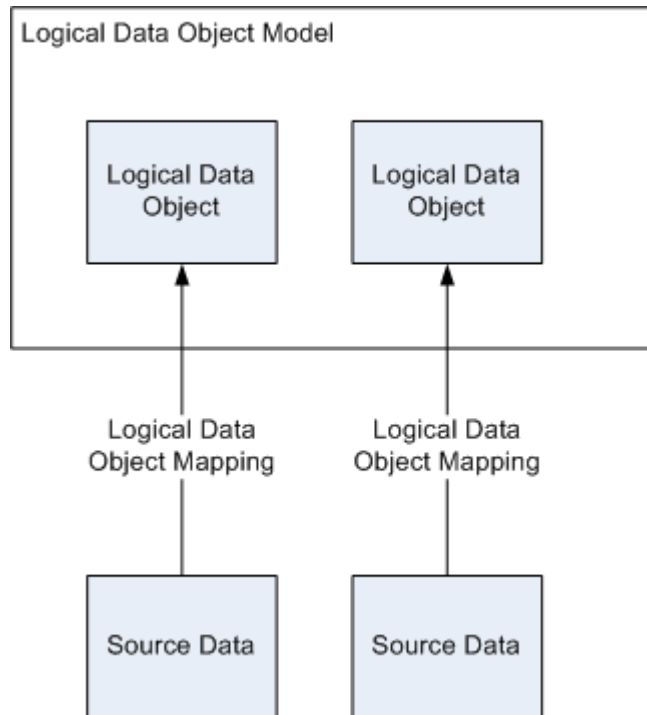
To develop a single view of data, define a logical data object model. A logical data object model describes data in an organization and the relationship between the data. You can use a data modeling tool, such as Erwin, to create the logical data object model. Or, you can manually create the model.

A logical data object model contains logical data objects. A logical data object is an object that describes a logical entity in an organization, such as a customer or an order. It has attributes and keys, and it describes relationships between attributes.

The logical data object model describes the relationship between logical data objects. For example, a logical data model defines a relationship between the Order ID attribute of the Order logical data object and the Customer ID attribute of the Customer logical data object. The model states that each order ID must be associated with a customer ID.

A logical data object mapping links a logical data object to one or more physical data objects. The mapping can include transformation objects that define the logic to transform data. For example, you can use a logical data object read mapping to access data from multiple sources and apply the output to a logical data object.

The following figure shows the components of a logical view of data:



Note: A logical data object mapping can also access data from one logical data object and apply the output to another logical data object.

After you create logical data object mappings for logical data objects in the model, you can create a data service for each logical data object mapping, and deploy the data services.

Lesson Objectives

In this lesson, you complete the following beginner-level tasks:

- Import a logical data object model that contains the Customer and Order logical data objects.
- Create a logical data object read mapping with the Customer logical data object as the mapping output. The mapping defines a single view of the customer data from the Los Angeles and Boston offices. The mapping also transforms the Boston customer data to conform to the format of the Los Angeles customer data.
- Run the mapping to view the combined customer data.

Lesson Prerequisites

Before you start this lesson, complete the following prerequisites:

- Set up Informatica Developer. For more information, see ["Setting Up Informatica Developer Overview" on page 25](#).
- Import the physical data objects. For more information, see ["Importing Physical Data Objects Overview" on page 35](#).
- Verify that the Customer_Order.xsd tutorial file is in the following directory on the Developer tool machine:
`<Informatica installation directory>\clients\DeveloperClient\Tutorials`

- Verify that the Customer_Order.xsd tutorial file is in the following directory on each machine that runs the Data Integration Service:

<Informatica installation directory>\server\Tutorials

Lesson Timing

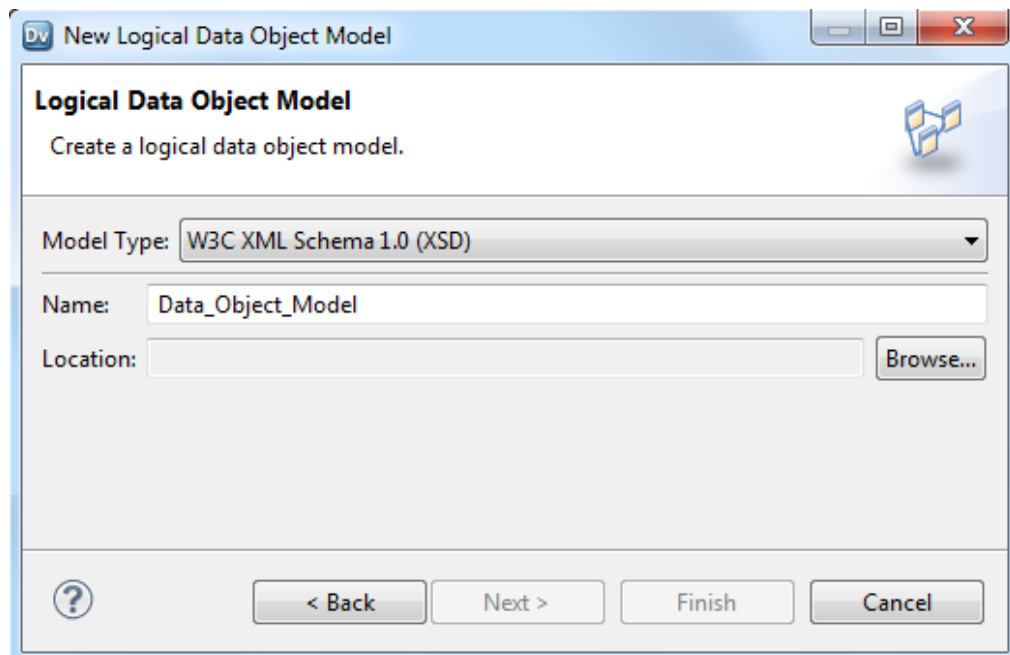
Set aside 20 minutes to complete the tasks in this lesson.

Step 1. Import a Logical Data Object Model

In this task, you import a logical data object model that contains the Customer and Order logical data objects.

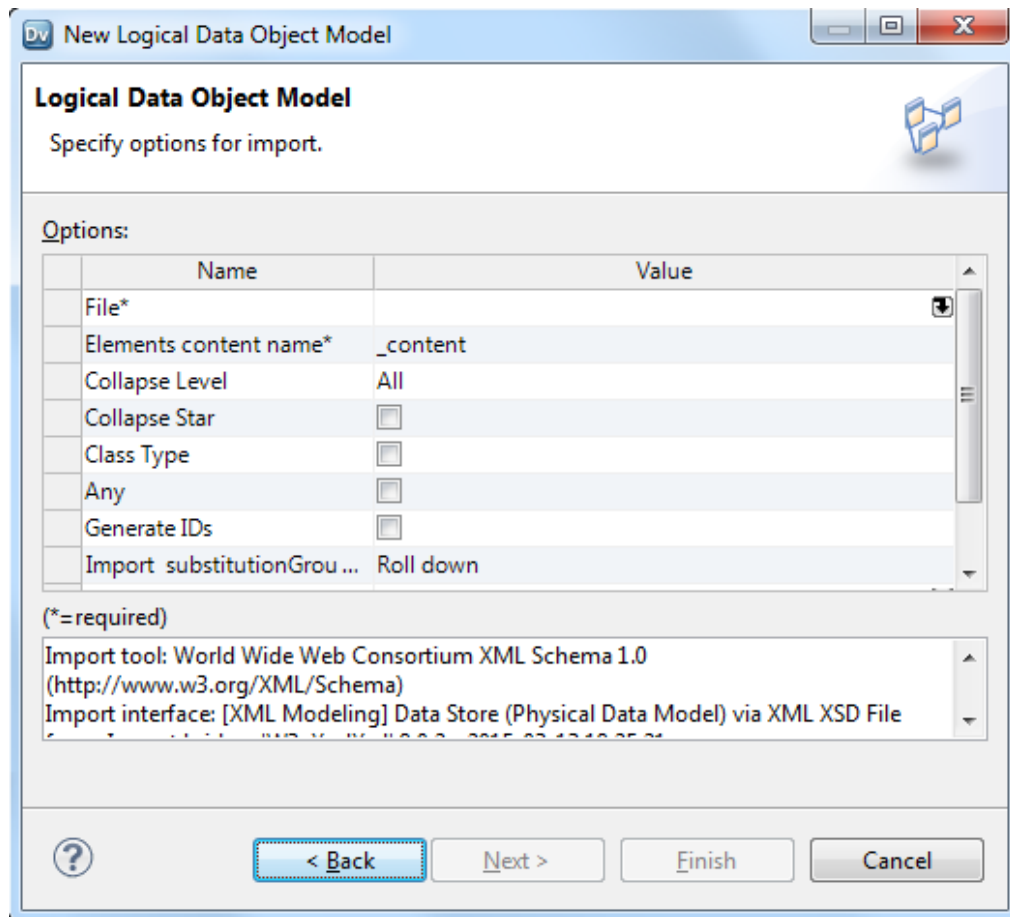
1. In the **Object Explorer** view, select the Tutorial project.
2. Click **File > New > Logical Data Object Model**.
The **New** dialog box appears.
3. Select **Logical Data Object Model from Data Model** and click **Next**.

The **New Logical Data Object Model** dialog box appears.



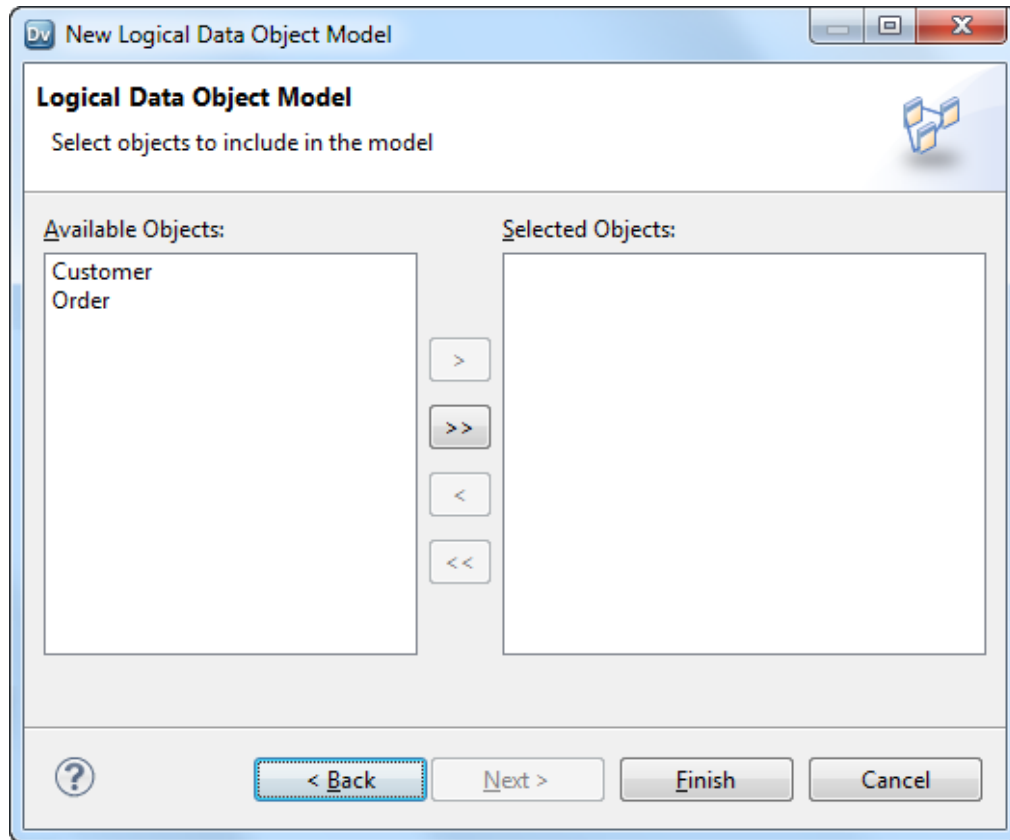
4. Select **W3C XML Schema 1.0 (XSD)** as the model type.
5. Enter Customer_Order as the logical data object model name.
6. Click **Browse** to select a location. Select the Tutorial folder, click **OK**, and then click **Next**.

The **New Logical Data Object Model** dialog box shows the import options.



7. In the Value column of the **File** property, click the **Open** button () to select an XSD file.
The **Open** dialog box appears.
8. Navigate to Customer_Order.xsd in the following directory: <Informatica installation directory>\clients\DeveloperClient\Tutorials
9. Click **Open**.
The **Open** dialog box closes. The **New Logical Data Object Model** dialog box shows the directory path and name of the model file.
10. Click **Next**.

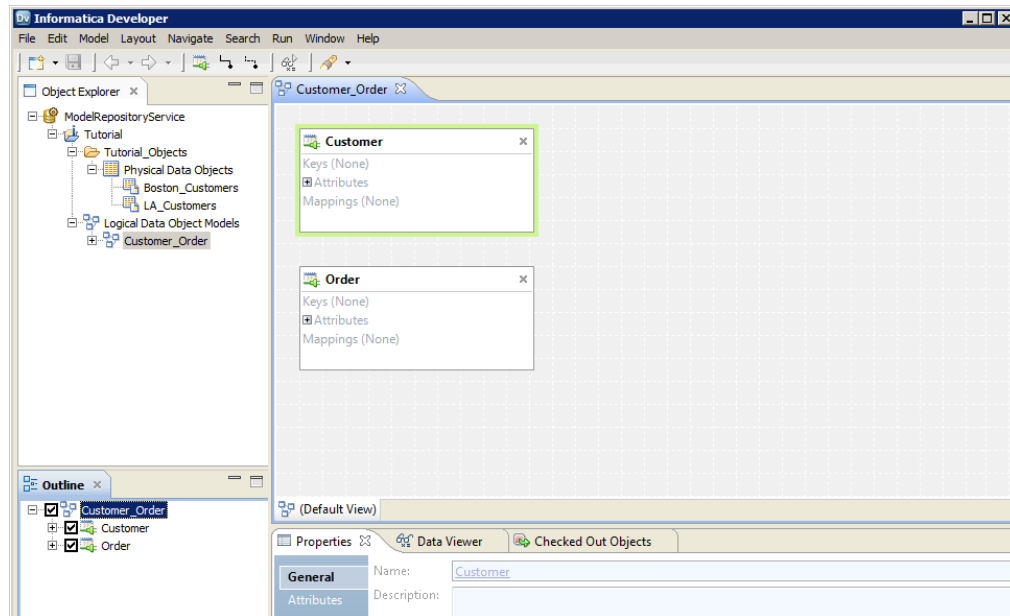
The **New Logical Data Object Model** dialog box shows the logical data objects that you can import.



11. Click the **Move all items** button to add the Customer and Order logical data objects to the logical data object model.

12. Click **Finish**.

The Customer_Order logical data object model appears in the **Object Explorer** view. The Customer and Order logical data objects appear in the editor.



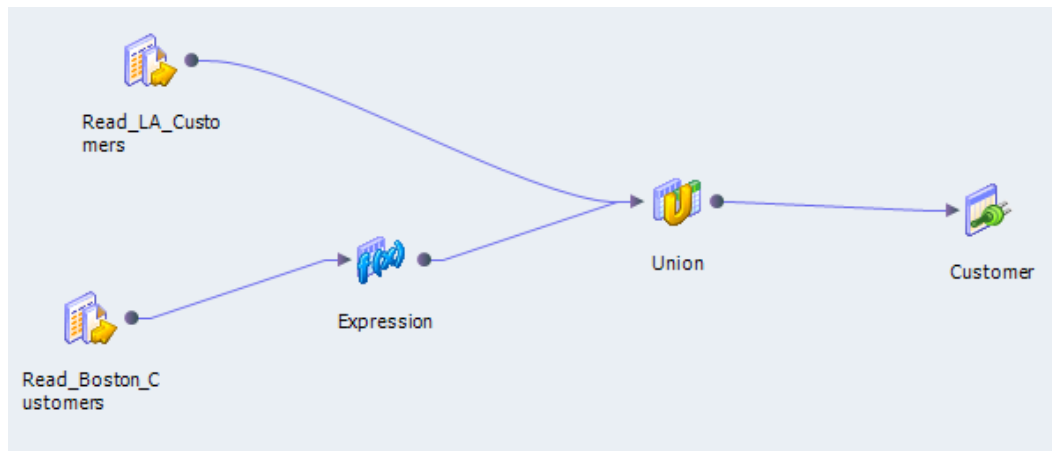
Step 2. Create a Logical Data Object Mapping

In this task, you create a logical data object read mapping that reads and transforms data from physical data objects and writes the output to a logical data object.

To create the logical data object mapping, you will complete the following tasks:

1. Add a logical data object mapping to the logical data object.
2. Add the LA_Customers and Boston_Customers physical data objects as sources in the mapping.
3. Add an Expression transformation to concatenate the first names and last names in the Boston customer data.
4. Add a Union transformation to combine the Los Angeles and Boston customer data, and then write the output to the Customer logical data object.

The following figure shows the logical data object mapping:



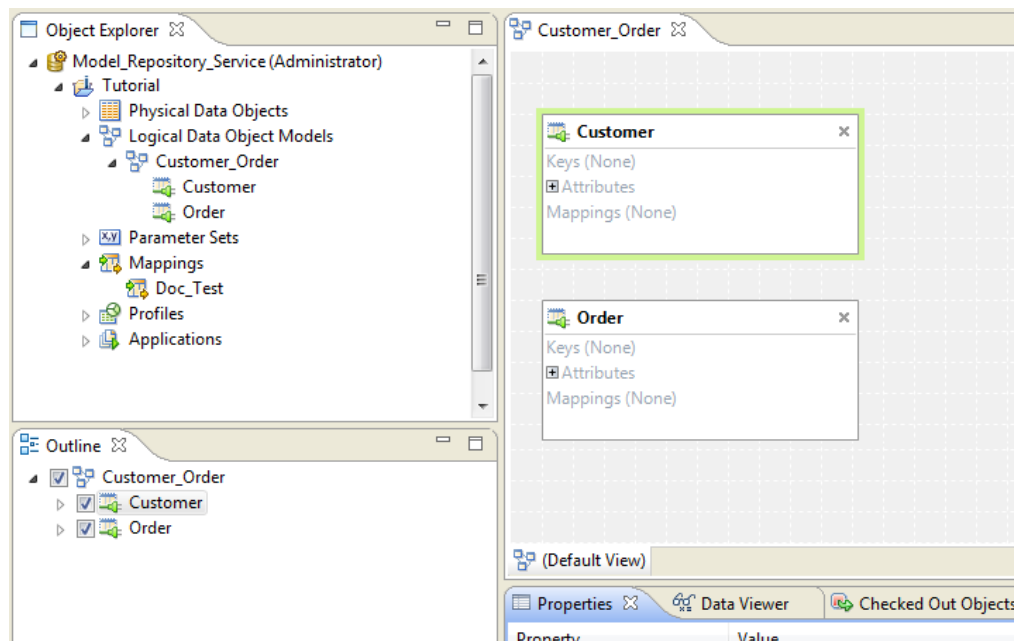
Add a Read Mapping to the Logical Data Object

Add a read mapping to allow the Customer-Order logical data object to fetch data.

1. In the **Object Explorer** view, expand the Logical Data Object Models folder in the Tutorial project.
2. Double-click the Customer_Order logical data object model.

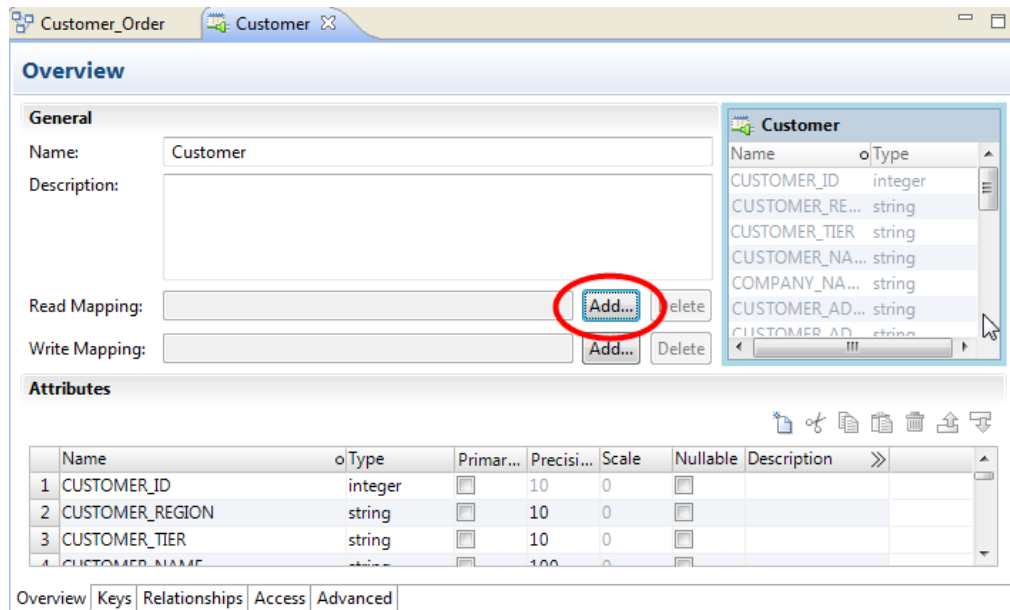
The Customer_Order logical data object model opens.

The following image shows the Customer_Order logical data object model in the editor:



3. In the Object Explorer, double-click the Customer logical data object to open it in the editor.
The logical data object editor opens. The editor contains two areas, a General area where you can add a read mapping or a write mapping, and an Attribute area where you can edit column attributes.
4. In the General area, click **Add** to add a read mapping to the Customer logical data object.

The following image shows the **Add** button:



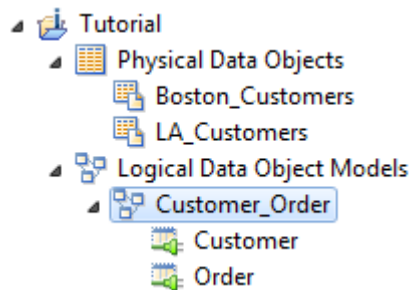
5. Accept Read Mapping as the name of the read mapping, and then click **Finish**.
6. Save the logical data object.

Adding Sources to the Mapping

Add the LA_Customers and Boston_Customers data objects to the mapping as sources.

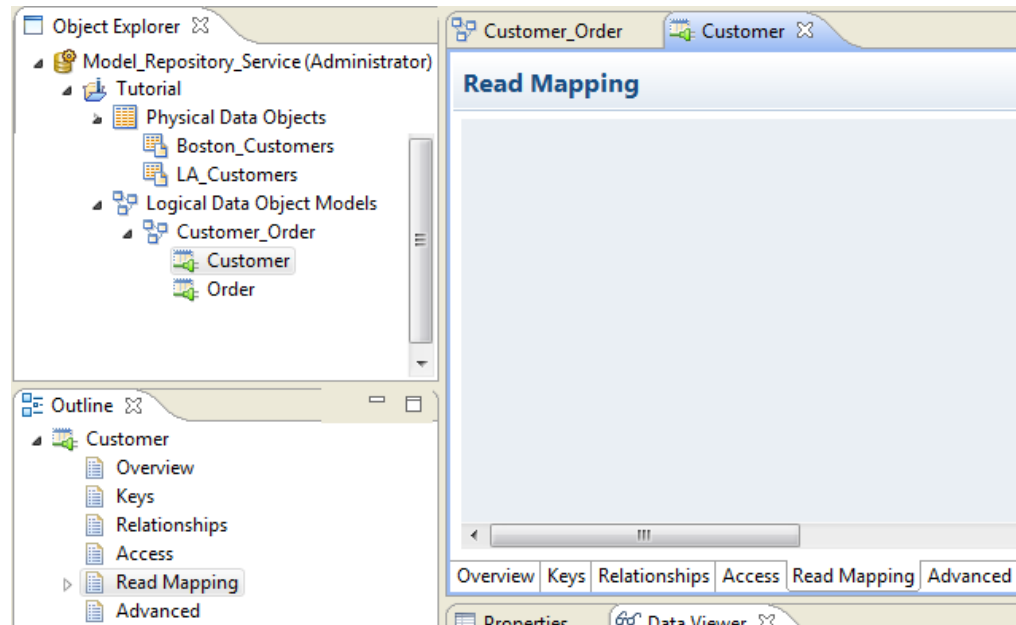
1. In the **Object Explorer** view, expand the Logical Data Object Models node and the Customer_Order logical data object model.

The following image shows the Customer_Order logical data object model and the logical data objects that it contains, Customer and Order:



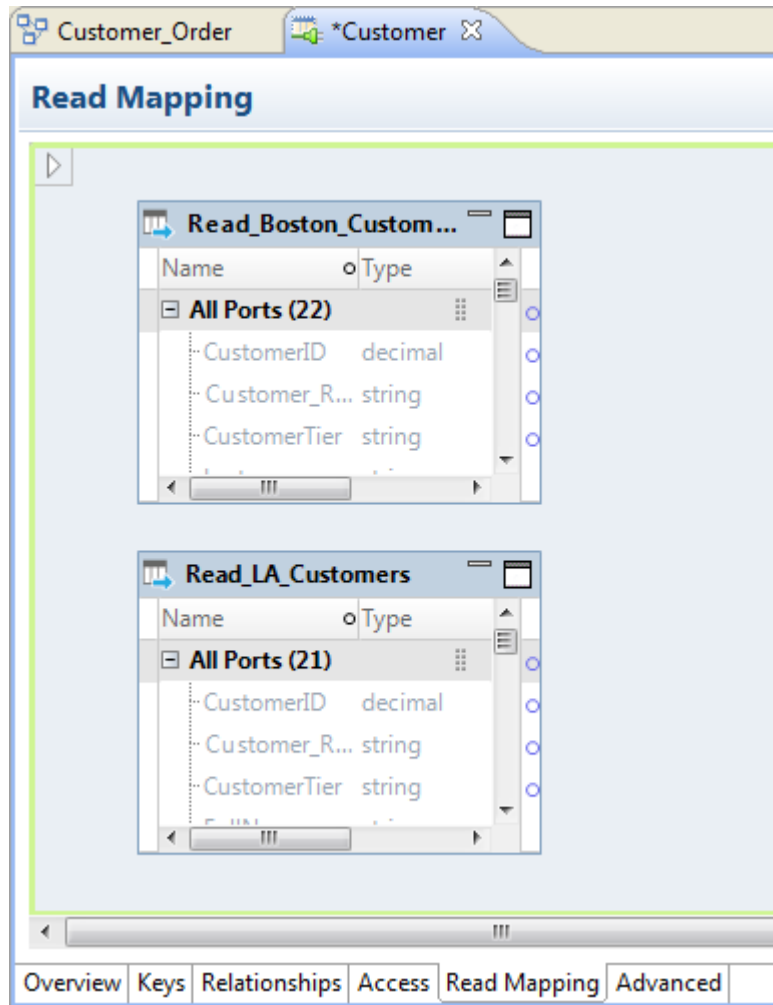
2. Right-click the Customer logical data object and select **Open**.
The Customer logical data object editor opens.
3. In the Customer logical data object editor, select the **Read Mapping** tab.
The Read Mapping editor opens.

The following image shows the Read Mapping editor:



4. In the **Object Explorer** view, expand the Physical Data Objects folder in the Tutorial project.
5. Drag LA_Customers in to the Read Mapping editor.
The **Add to Mapping** dialog box appears.
6. Verify that the Data Access Type is Read, and click **OK**.
Read_LA_Customers appears in the editor.
7. Drag Boston_Customers in to the editor.
8. Verify that the Data Access Type is Read, and click **OK**.
Read_Boston_Customers appears in the editor.

The following image shows the Customer logical data object.



9. Click **File** > **Save** to save the logical data object mapping.

Adding an Expression Transformation to the Mapping

Add an Expression transformation to concatenate the first and last names in the Boston customer data.

1. Right-click an empty area in the editor, and then select **Add Transformation**.

The **Add Transformation** dialog box appears.

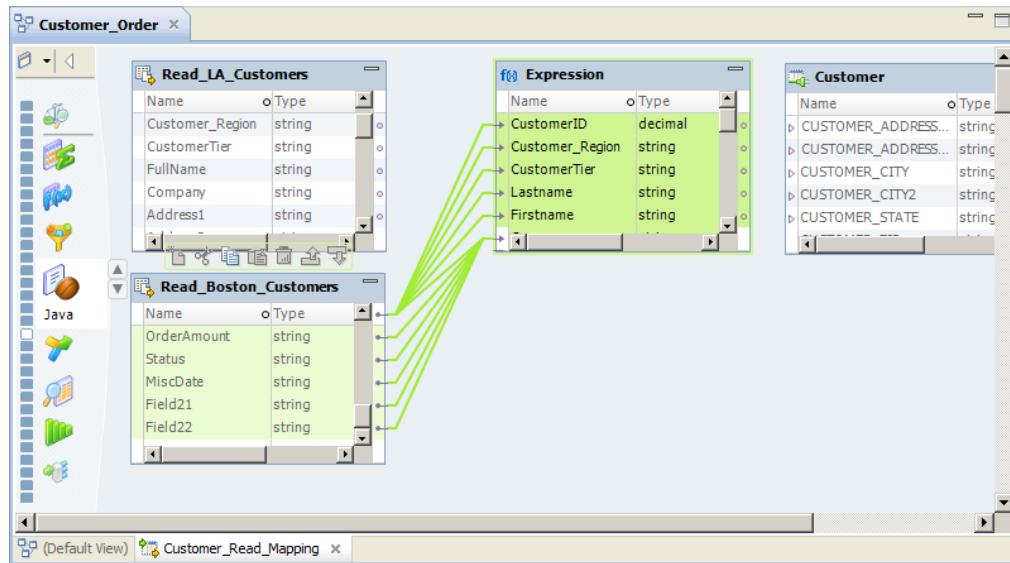
2. Select the Expression transformation, and then click **OK**.


An Expression transformation appears in the editor.

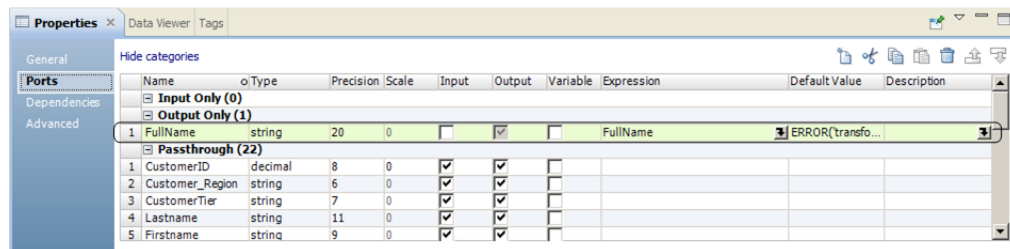
3. To create ports in the Expression transformation, select all columns in the Read_Boston_Customers source and drag them to the Expression transformation.


Tip: To select all columns in the source, right-click inside of the Read_Boston_Customers source in the editor, and then click **Select All**.

The columns appear as ports in the Expression transformation.

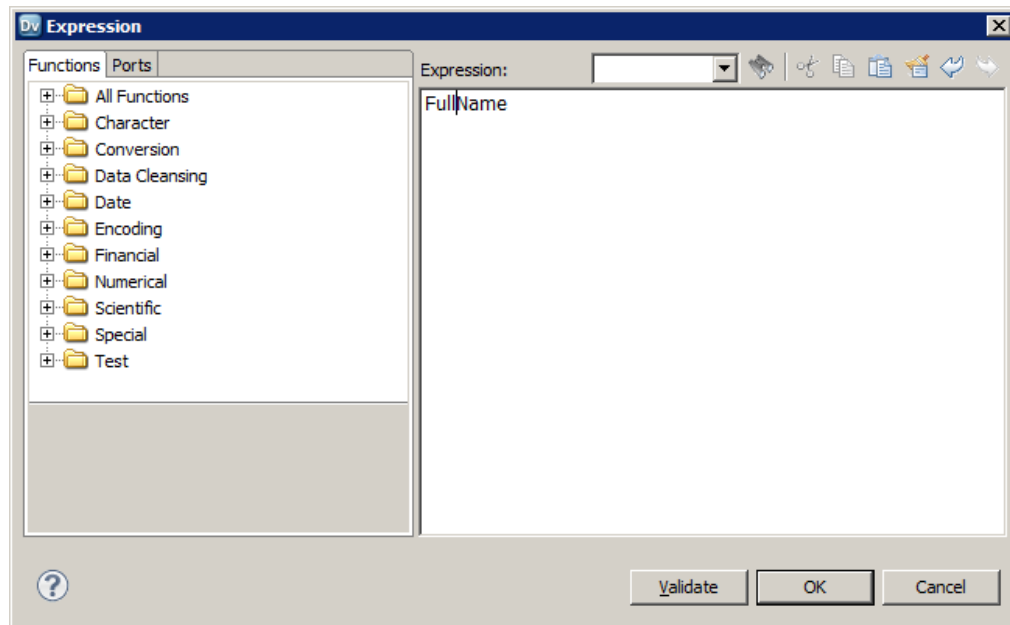


4. Select the Expression transformation.
5. Click the **Properties** view.
6. In the **Properties** view, click the **Ports** tab.
7. In the **Ports** tab, select **Output Only** and then click the **New (Insert)** button () to create an output port. A port appears.
8. Enter FullName as the port name and change the precision to 20.



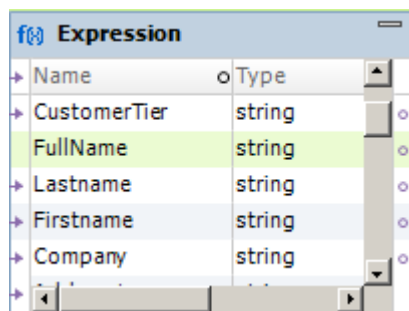
9. In the Expression column for the FullName port, click the **Open** button () to open the **Expression** editor.

The Expression editor appears.



10. Replace the existing expression in the **Expression** editor with the following expression:
`CONCAT (CONCAT (FIRSTNAME, ' '), LASTNAME)`
11. Click **Validate** to validate the expression.
12. Click **OK**.
13. Click **OK** to exit the **Expression** editor.
14. Select the Expression transformation in the editor.
15. In the Expression transformation, select the FullName port.
16. Click the **Move Up** button until you move the FullName port below the CustomerTier port.

The following figure shows the FullName port below the CustomerTier port:



You move the port to match the order of the ports in Read_LA_Customers source. The port order must match to combine data from both sources in the Union transformation.

17. Click **File > Save** to save the logical data object mapping.
18. Select the Expression transformation in the editor.
19. Click the **Data Viewer** view.

20. Click **Run** in the **Data Viewer** view to show the results of the Expression transformation.

The Data Integration Service processes the data from the Read_Boston_Customers source and the Expression transformation. The Developer tool shows the results in the Output window. The results show that the Data Integration Service concatenated the FirstName and LastName columns from the source.

Adding a Union Transformation to the Mapping

Add a Union transformation to combine the Boston customer data and Los Angeles customer data.

1. Right-click an empty area in the editor, and then select **Add Transformation**.

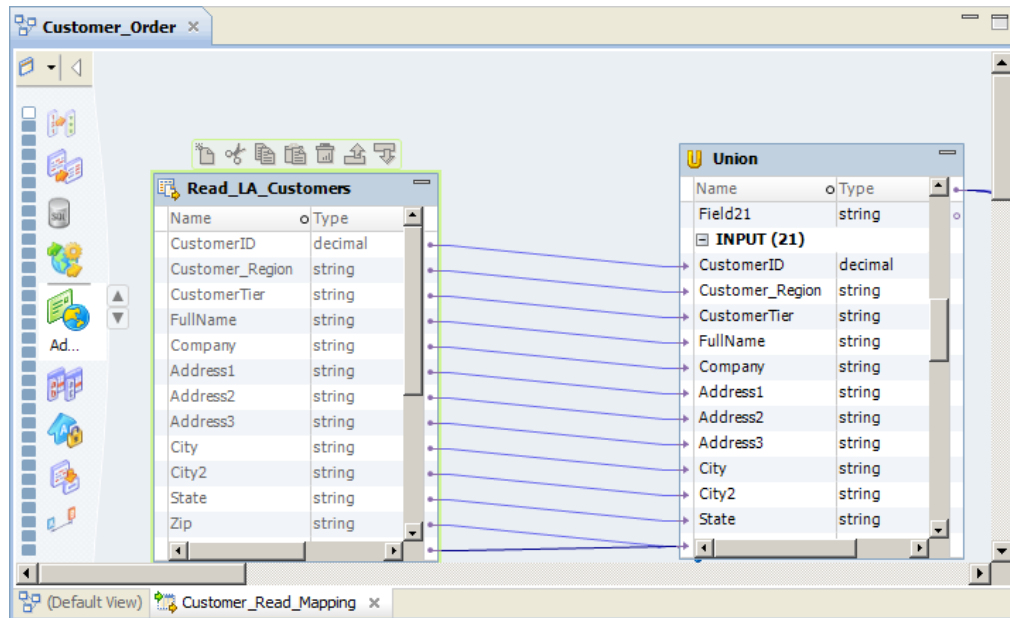
The **Add Transformation** dialog box appears.

2. Select the Union transformation, and then click **OK**.

A Union transformation appears in the editor.

3. To add the columns of the Read_LA_Customers source as ports in the Union transformation, select all columns in the Read_LA_Customers source and drag them to the Union transformation.


The ports appear in the input group and the output group of the Union transformation.



4. Select the Union transformation in the editor.

5. Select the **Properties** view.

6. Click the **Groups** tab in the **Properties** view.

7. Click the **New (Insert)** button () to add a group to the Union transformation.

You create a new group for the Read_Boston_Customers source data. The Union transformation contains a distinct input group for each source.

8. Name the group `Boston_Input`.

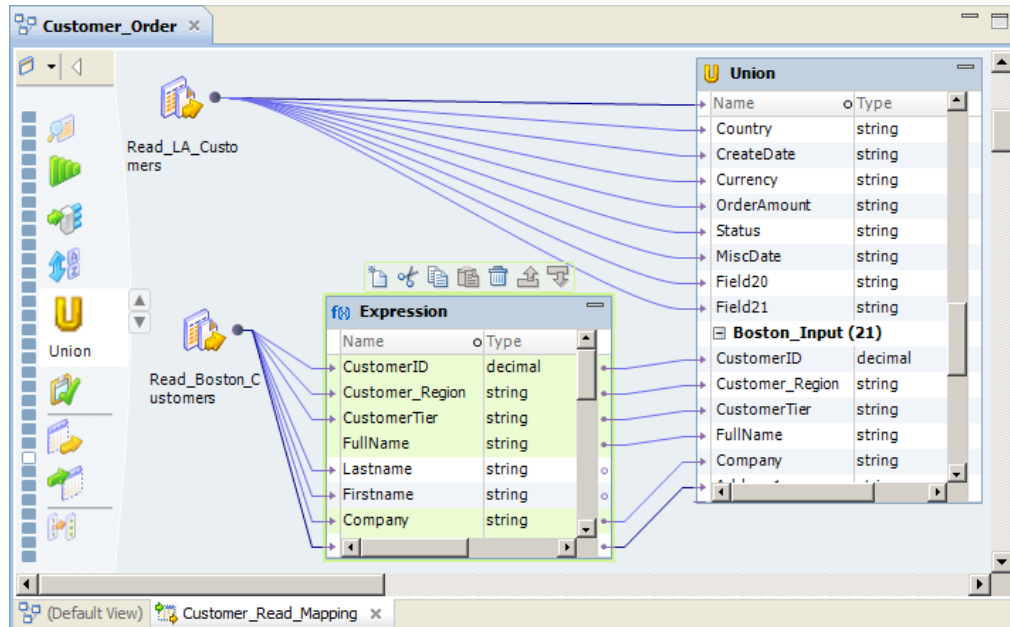
9. In the editor, scroll through the ports of the Union transformation until you see the Boston_Input group.

10. In the Expression transformation, select all of the ports except LastName and FirstName.

Tip: Select all ports in the Expression transformation, and then use CTRL-click to clear the LastName and FirstName ports from the selection.

11. Scroll to the top of the Expression transformation, and then drag the selected ports in the Expression transformation to the Boston_Input group of the Union transformation.

The ports appear in the Boston_Input group of the Union transformation. Because the Expression transformation port names match the existing output port names in the Union transformation, the Developer tool does not create output ports for the Boston customer data.



12. Select all ports in the output group of the Union transformation, except Customer_Region, and drag them to the Customer transformation.

Tip: Hold down the Shift key to select multiple columns. You might need to scroll down the list of columns to select all of them.

The Developer tool links the ports in the Union transformation to the ports in the Customer mapping output.

13. Right-click an empty area in the editor and click **Validate** to validate the mapping.

The Developer tool displays a message stating whether validation errors occurred.

14. Click **OK**.

15. Click **File > Save** to save the logical data object mapping.

Tip: Right-click an empty area in the editor and click **Arrange All** to arrange mapping objects in the editor.

Step 3. Run the Logical Data Object Mapping

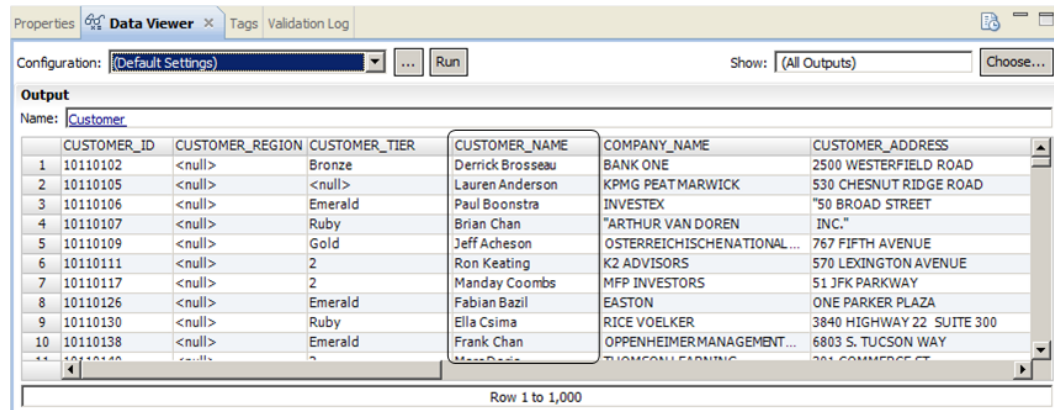
In this task, you run the logical data object mapping to create a single view of customer data from the Los Angeles and Boston offices.

- ▶ Right-click an empty area in the editor and click **Run Data Viewer** to run the mapping.

The **Data Viewer** view appears, and the Data Integration Service runs the mapping.

After the Data Integration Service runs the mapping, the Developer tool shows the data in the **Output** section of the **Data Viewer** view. The **Output** section shows the combined data from the Read_LA_Customers source

and Read_Boston_Customers source. Also, the CUSTOMER_NAME column shows the concatenated first and last names from the Read_Boston_Customers source.



	CUSTOMER_ID	CUSTOMER_REGION	CUSTOMER_TIER	CUSTOMER_NAME	COMPANY_NAME	CUSTOMER_ADDRESS
1	10110102	<null>	Bronze	Derrick Brosseau	BANK ONE	2500 WESTERFIELD ROAD
2	10110105	<null>	<null>	Lauren Anderson	KPMG PEAT MARWICK	530 CHESNUT RIDGE ROAD
3	10110106	<null>	Emerald	Paul Boonstra	INVESTEX	"50 BROAD STREET
4	10110107	<null>	Ruby	Brian Chan	"ARTHUR VAN DOREN INC."	
5	10110109	<null>	Gold	Jeff Acheson	OSTERREICHISCHENATIONAL...	767 FIFTH AVENUE
6	10110111	<null>	2	Ron Keating	K2 ADVISORS	570 LEXINGTON AVENUE
7	10110117	<null>	2	Manday Coombs	MFP INVESTORS	51 JFK PARKWAY
8	10110126	<null>	Emerald	Fabian Bazil	EASTON	ONE PARKER PLAZA
9	10110130	<null>	Ruby	Ella Csima	RICE VOELKER	3840 HIGHWAY 22 SUITE 300
10	10110138	<null>	Emerald	Frank Chan	OPPENHEIMERMANAGEMENT...	6803 S. TUCSON WAY

Informatica Developer Tips

Use the following tips when you develop mappings.

Link ports automatically.

When you link ports automatically, you can link by position or by name. When you link ports automatically by name, you can specify a prefix or suffix by which to link the ports.

To link ports automatically, select **Mapping > Auto Link**, select the to and from objects, and then select whether to link the ports by name or position. If you link ports by name, you can click **Show Advanced** to specify a prefix or suffix for port names.

Align mapping objects as icons.

You can convert the mapping objects to icons and align the icons in the editor. To align mapping objects as icons, click **Layout > Arrange All Iconic**.

Navigate mapping objects with a quick outline.

A quick outline displays objects that are dependent on a mapping object selected in the editor. You can use the quick outline to sort dependent objects by name or by type, or to search for dependent objects. Select an object in the quick outline to navigate to the object in the editor.

To display the quick outline, select the mapping or an object in the mapping and then click **Navigate > Quick Outline**.

Use the point-and-click method to add functions and ports to a port expression.

When you create an expression, you can enter the expression manually or use the point-and-click method. To minimize errors when you create an expression, select functions and ports from the point-and-click interface.

To add a function to an expression, double-click the function on the **Functions** tab. To add a port to an expression, double-click the port name in the **Ports** tab.

Add comments within the expression.

Add comments to describe the expression or to specify a valid URL to access business documentation about the expression. The Data Integration Service ignores comments when processing the expression.

To add comments within the expression, use `--` or `//` comment indicators.

CHAPTER 6

Creating a Virtual View of Data

This chapter includes the following topics:

- [Creating a Virtual View of Data Overview, 60](#)
- [Step 1. Create an SQL Data Service, 62](#)
- [Step 2. Run an SQL Query to Preview Data, 64](#)
- [Step 3. Create an Application, 65](#)
- [Step 4. Deploy the Application, 66](#)
- [Informatica Developer Tips, 67](#)

Creating a Virtual View of Data Overview

Create a virtual view of data to query data from disparate locations without loading data to a central location.

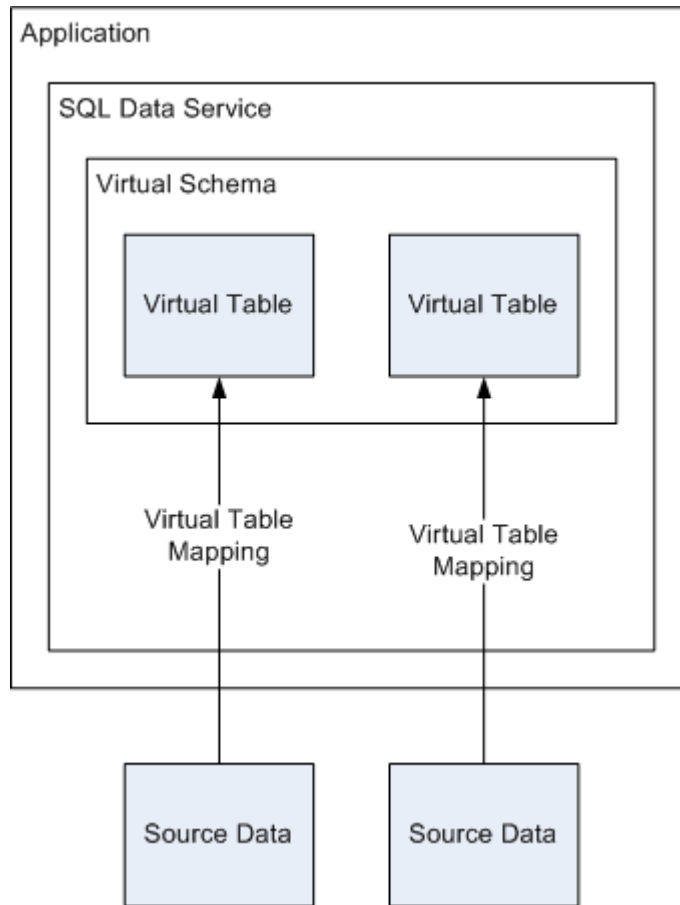
Lesson Concepts

A virtual view of data is a virtual database defined by an SQL data service that you can query as if it were a physical database.

To create a virtual database, you define an SQL data service in the Developer tool. The SQL data service must contain at least one virtual schema and virtual table. A virtual table can have a virtual table mapping that defines the data flow between the sources and the virtual table. You can create a virtual table manually, or create it from a physical or logical data object.

To run an SQL data service on a Data Integration Service, you must add the SQL data service to an application, and then deploy the application to the Data Integration Service. An application is a deployable object that can contain data objects, mappings, SQL data services, web services, and workflows. You deploy the application to a Data Integration Service to make the virtual database available for end users to query. The Data Integration Service processes end-user queries on objects included in deployed applications.

The following figure shows the components of a virtual view of data:



Lesson Objectives

In this lesson, you complete the following beginner-level tasks:

- Create an SQL data service to define a virtual database that contains customer data.
- Preview the virtual data.
- Create an application that contains the SQL data service.
- Deploy the application to a Data Integration Service.

Lesson Prerequisites

Before you start this lesson, complete the following prerequisites:

- Set up Informatica Developer. For more information, see [“Setting Up Informatica Developer Overview” on page 25](#).
- Import the physical data objects. For more information, see [“Importing Physical Data Objects Overview” on page 35](#).
- Create the Customer_Order logical data object model. For more information, see [“Creating a Logical View of Data Overview” on page 44](#).

Lesson Timing

Set aside 15 to 20 minutes to complete the tasks in this lesson.

Step 1. Create an SQL Data Service

In this task, you create an SQL data service to create a virtual database that end users can query.

1. In the **Object Explorer** view, select the Tutorial project.

2. Click **File > New > Data Service**.

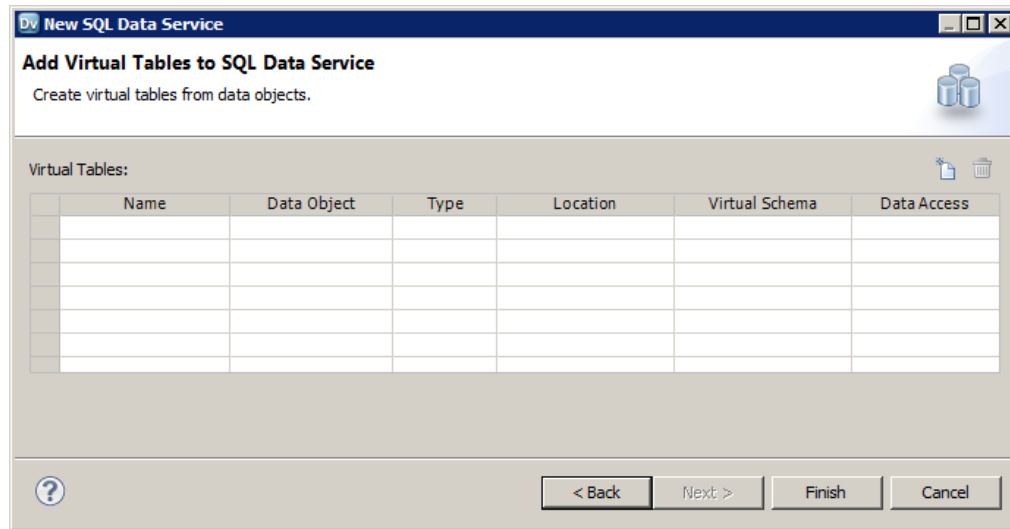
The **New** dialog box appears.

3. Select **SQL Data Service** and click **Next**.

The **New SQL Data Service** dialog box appears.

4. Enter `All_Customers` for the SQL data service name and click **Next**.

The **Add Virtual Tables To SQL Data Service** dialog box appears.



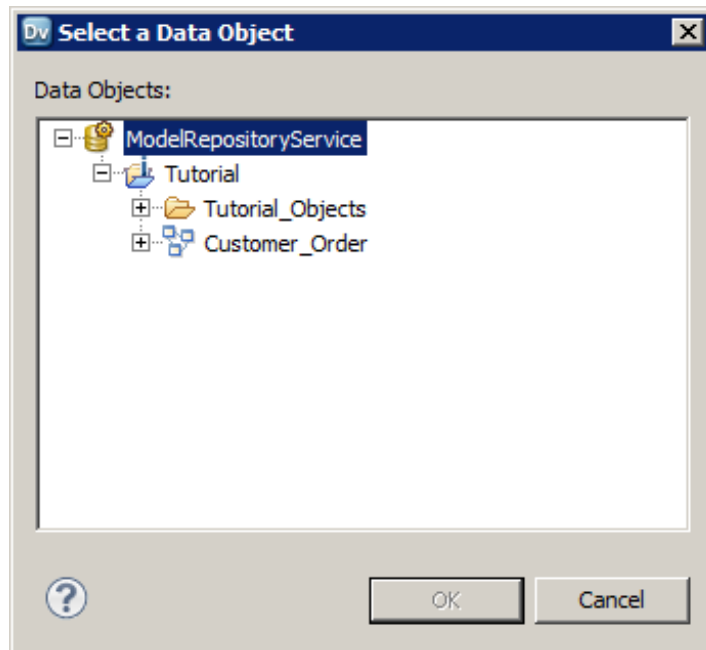
5. To create a virtual table, click the **New** button (📄).

The Developer tool adds a virtual table to the list of virtual tables.

6. Enter `Customers` for the virtual table name.

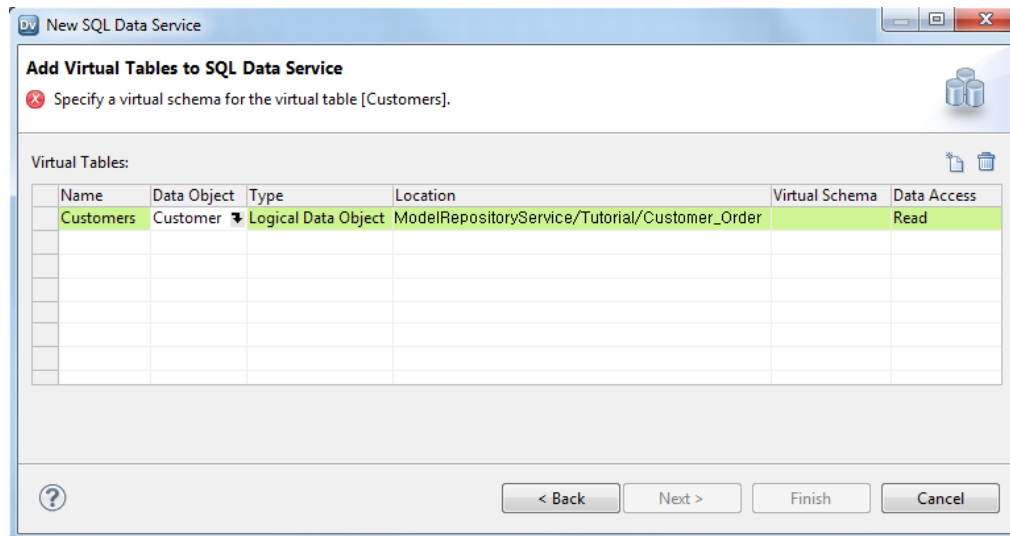
7. In the Data Object field for the virtual table, click the Open button (🔍) to add a logical data object.

The **Select a Data Object** dialog box appears.



8. In the Tutorial folder, expand the Customer_Order logical data object model, and select the Customer logical data object.
9. Click **OK**.

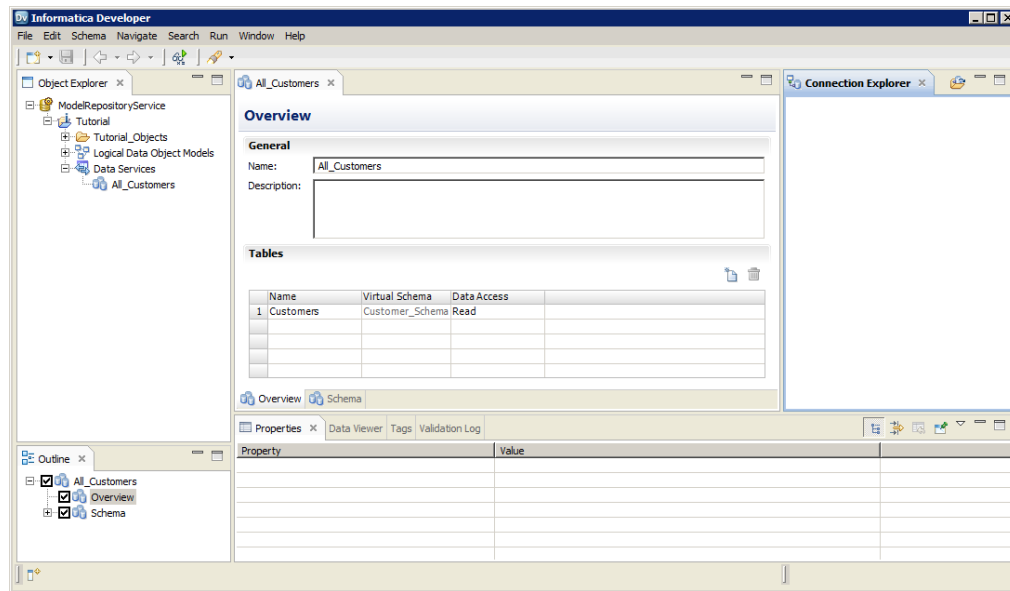
The Developer tool adds Customer as the virtual table source. It also specifies Logical Data Object as the source type and the Tutorial project as the location.



10. Enter `Customer_Schema` in the Virtual Schemas column and click **Enter**.

11. Click **Finish**.

The Developer tool creates and displays the All_Customers SQL data service.

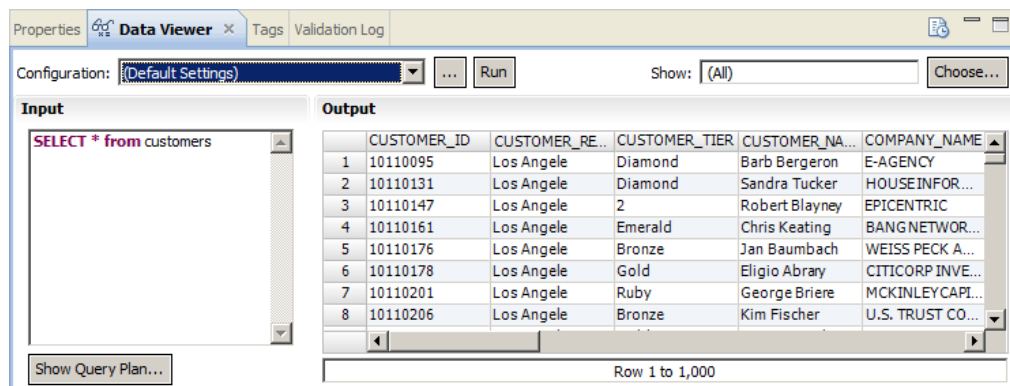


Step 2. Run an SQL Query to Preview Data

In this task, you run an SQL query against the Customers virtual table to preview the data.

1. Select the **Data Viewer** view to preview the data of the SQL data service.
2. In the **Input** section of the **Data Viewer** view, enter the following SQL statement: `SELECT * from customers`
3. Click **Run**.

The **Output** section of the **Data Viewer** view displays the combined customer data from the Los Angeles and Boston offices.



Step 3. Create an Application

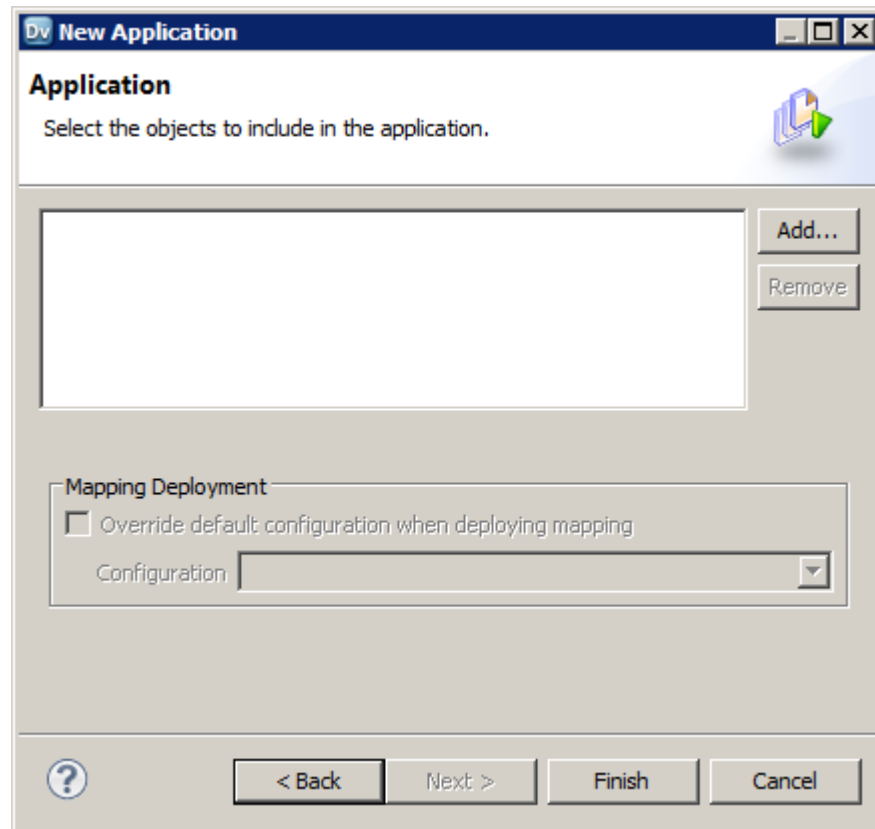
In this task, you create an application that contains the SQL data service. You must add the SQL data service to an application to deploy it to a Data Integration Service.

1. In the **Object Explorer** view, select the Tutorial project.
2. Click **File > New > Application**.

The **New Application** dialog box appears.

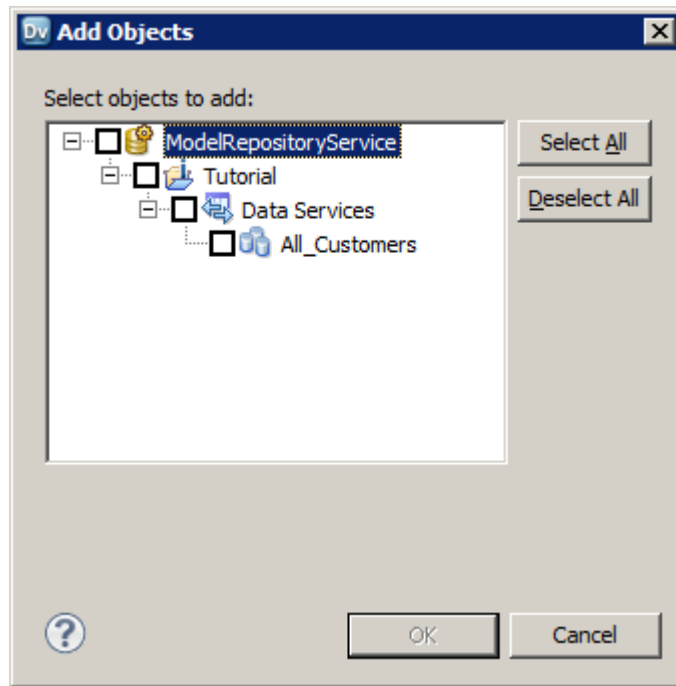
3. Enter `All_Customers_App` for the application name and click **Next**.

The New Application dialog box shows that the application contains no objects.



4. Click **Add**.

The **Add Objects** dialog box appears.



5. Select the All_Customers check box and click **OK**.

The All_Customers SQL data service appears in the New Application dialog box.

6. Click **Finish**.

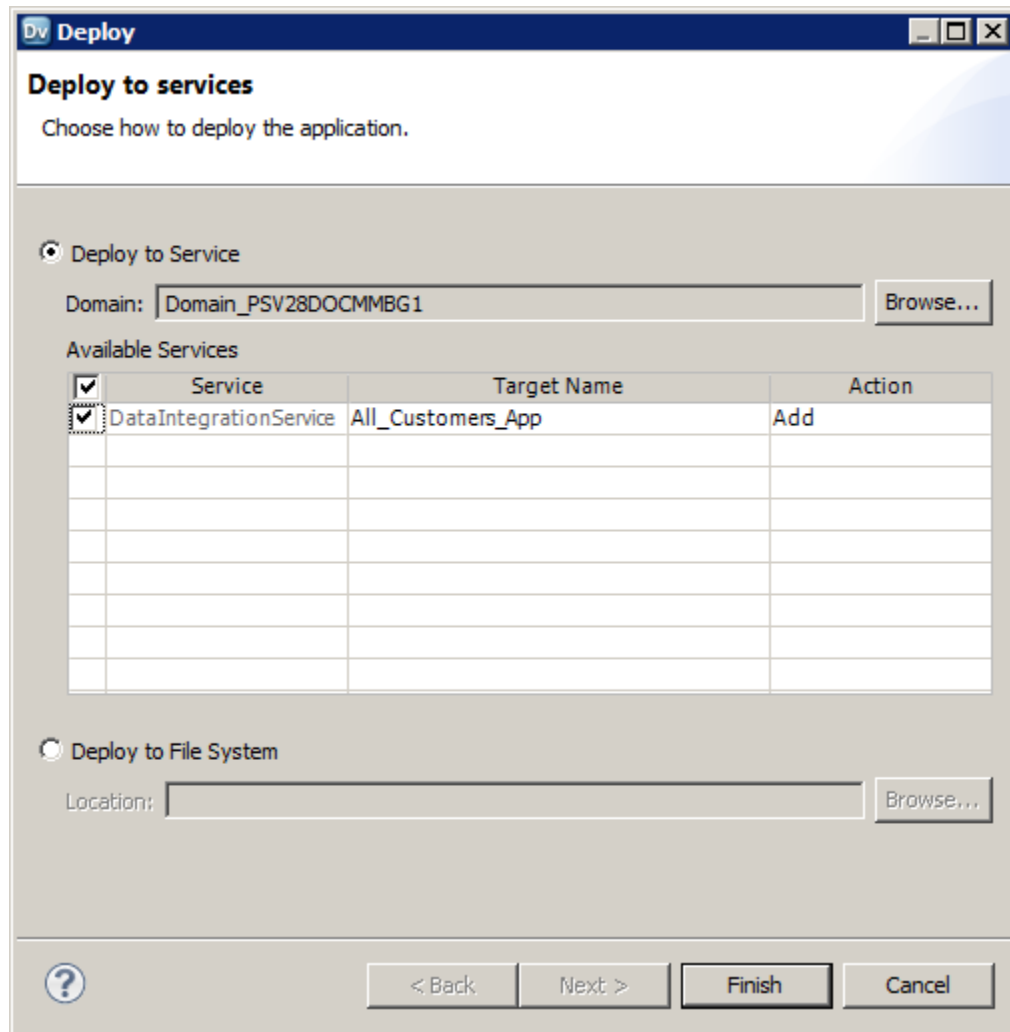
The Developer tool creates an application that contains the All_Customers SQL data service.

Step 4. Deploy the Application

In this task, you deploy the application to a Data Integration Service. You deploy the application to enable end users to query virtual tables in the SQL data service through a JDBC or ODBC client tool.

1. In the **Object Explorer** view, expand Applications in the Tutorial project.
2. Right-click the All_Customers_App application and click **Deploy**.
The **Deploy** dialog box appears.
3. Click **Browse** next to the **Domain** option.
4. Select a domain and click **OK**.
5. Select the Data Integration Service that you want to deploy the application to.

The following figure shows the selected domain and Data Integration Service:



6. Click **Finish**.
When the deployment succeeds, the **Deploy Completed** dialog box appears.
7. Click **OK**.

Informatica Developer Tips

Use the following tips when you read a virtual view of data.

View the SQL query plan to troubleshoot queries on an SQL data service.

View the SQL query plan to troubleshoot queries against a deployed SQL data service.

To run an SQL query plan, open the SQL data service in the editor, click the **Data Viewer** view, and then click **SQL Query Plan**.

Assign a tag to group objects according to their business usage.

A tag is metadata that defines an object in the Model repository based on business usage. Create tags to group objects according to their business usage.

To assign a tag to an object, create the tag, open the object in the editor, click **Edit** in the **Tags** view, and then assign the tag to the object.

Group error messages by object or object type in the Validation Log view.

You can group error messages by object or object type in the **Validation Log** view.

To group error messages, in the **Validation Log** view, select the **Menu** button () , select **Group By**, and then select **Object** or **Object Type**.

Limit the number of error messages per group that appear in the Validation Log view.

You can limit the number of error messages that appear in the **Validation Log** view.

To limit the number of error messages, click **Window > Preferences**, select **Informatica > Validation** in the **Preferences** dialog box, and then select the **Use Error Limits** check box and set the number of error messages.

CHAPTER 7

Creating a Web Service

This chapter includes the following topics:

- [Creating a Web Service Overview, 69](#)
- [Step 1. Create a Web Service, 71](#)
- [Step 2. View and Validate the Operation Mapping, 75](#)
- [Step 3. Test the Operation Mapping, 76](#)
- [Step 4. Create an Application, 77](#)
- [Step 5. Deploy the Application, 78](#)
- [Informatica Developer Tips, 79](#)

Creating a Web Service Overview

Create a web service to provide access to data integration functionality through a web service interface. For example, you can create a web service to enable customer service representatives to access customer data from a web site.

Lesson Concepts

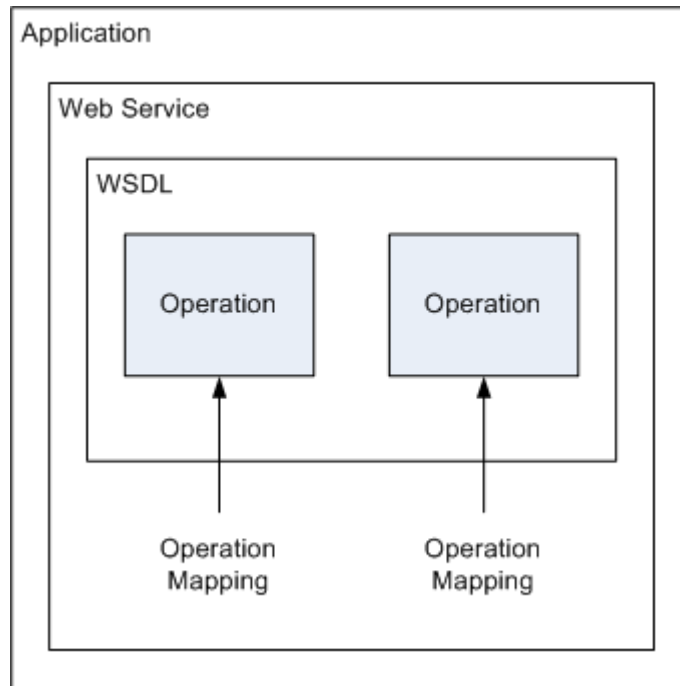
You create web services in the Developer tool. A web service can have one or more operations. Each operation defines an action that the web service client can perform when it connects to a web service. Operations of a web service are defined in a WSDL.

A WSDL is an XML schema that describes the protocols, formats, and signatures of the web service operations. It contains a description of the data to be passed to the web service so that both the sender and the receiver of the service request understand the data being exchanged.

In the Developer tool, each operation corresponds to an operation mapping. The operation mapping processes the data that it receives in the SOAP request.

To run a web service on a Data Integration Service, you must add the web service to an application, and then deploy the application to the Data Integration Service. An application is a deployable object that can contain data objects, mappings, SQL data services, web services, and workflows. You deploy the application to a Data Integration Service to make the web service available for web service clients to connect to.

The following figure shows the components of a web service:



Lesson Objectives

In this lesson, you complete the following intermediate-level tasks:

- Create a web service that provides access to customer details.
- View and validate the operation mapping.
- Test the operation mapping.
- Create an application that contains the web service.
- Deploy the application to a Data Integration Service.

Lesson Prerequisites

Before you start this lesson, complete the following prerequisites:

- Set up Informatica Developer. For more information, see [“Setting Up Informatica Developer Overview” on page 25](#).
- Import the physical data objects. For more information, see [“Importing Physical Data Objects Overview” on page 35](#).
- Create the Customer_Order logical data object model. For more information, see [“Creating a Logical View of Data Overview” on page 44](#).

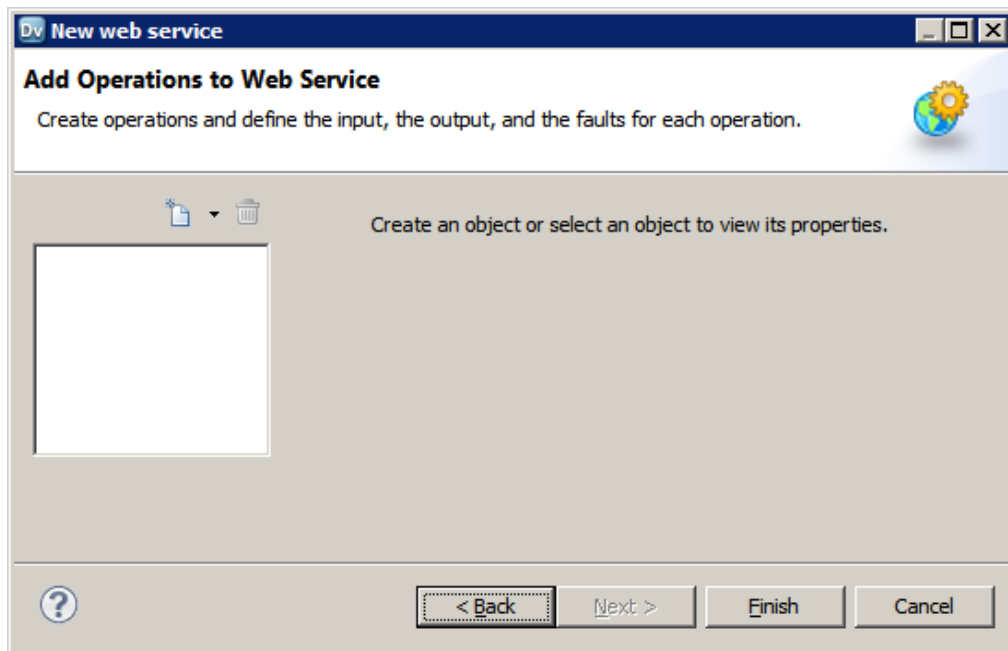
Lesson Timing

Set aside 15 to 20 minutes to complete this lesson.

Step 1. Create a Web Service

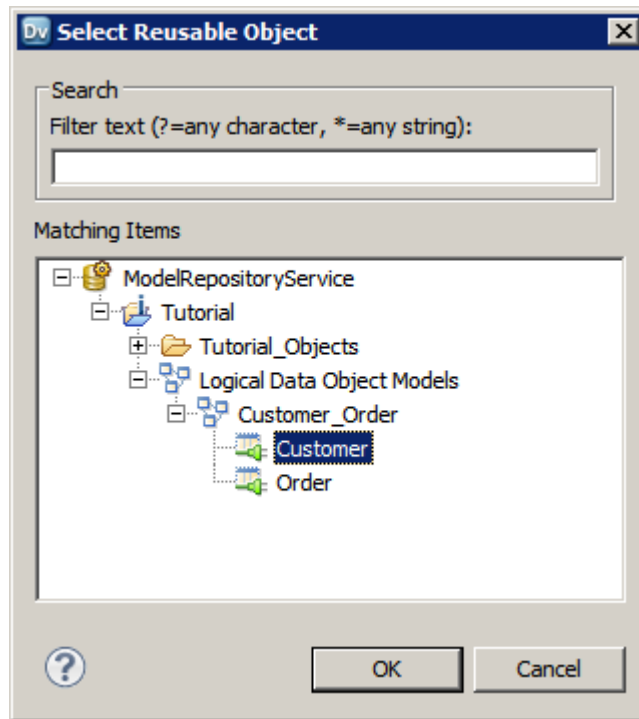
In this task, you create a web service with an operation that returns customer details based on a customer ID. You use a logical data object to create an operation.

1. In the **Object Explorer** view, select the Tutorial project.
2. Click **File > New > Data Service**.
The **New** dialog box appears.
3. Select **Web Service** and click **Next**.
The **New Web Service** dialog box appears.
4. Enter `Customer_Details` for the web service name and click **Next**.
The **Add Operations to Web Service** dialog box appears.



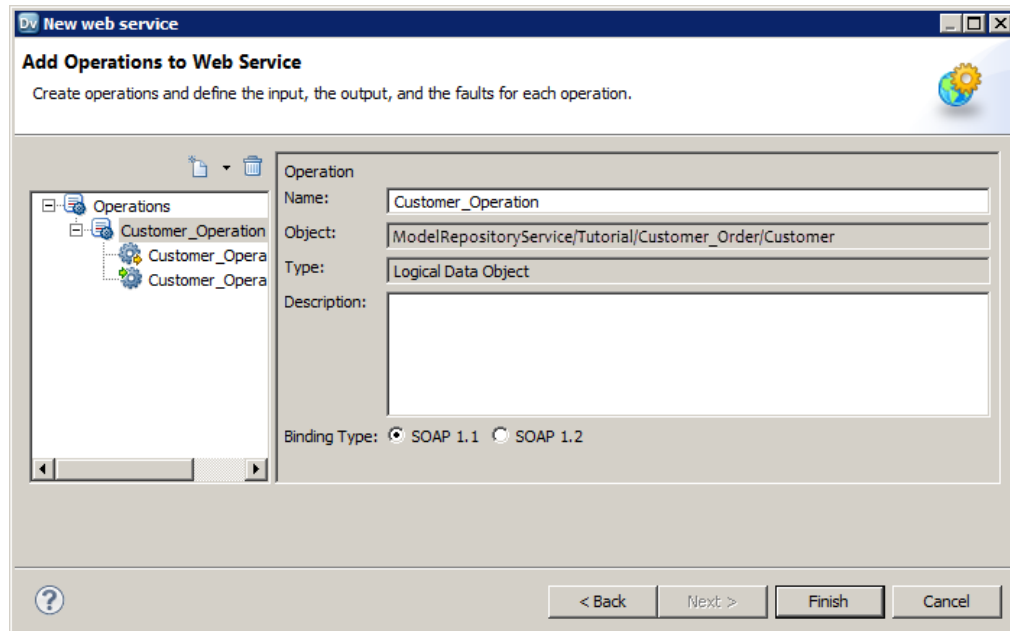
5. Click the arrow next to the **New** button and click **Operation > Create from Reusable Object**.
The **Select Reusable Object** dialog box appears.
6. Expand the Tutorial project, browse to **Logical Data Object Models > Customer_Order**, and then select **Customer**.

The following figure shows the Customer logical data object in the hierarchy of the Tutorial project:



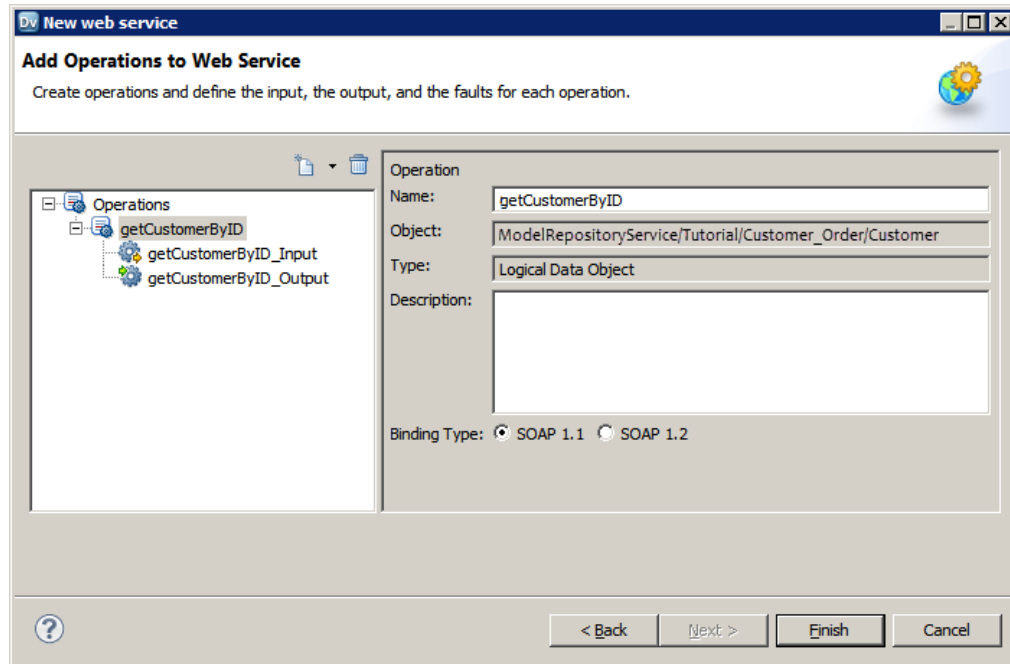
7. Click **OK**.

Operations appear in the **Add Operations to Web Service** dialog box.



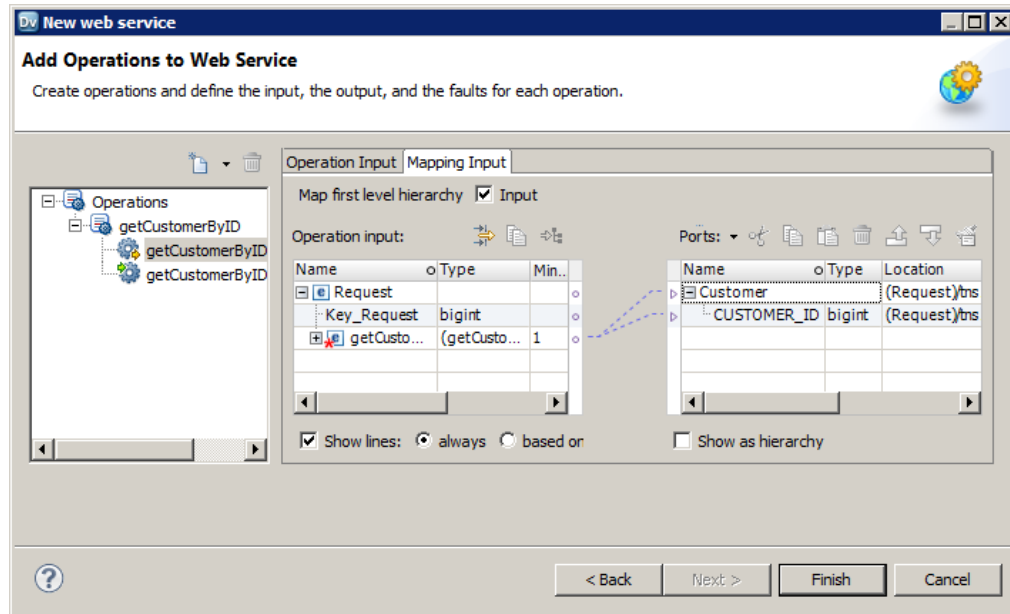
8. Enter `getCustomerByID` for the operation name.

The Developer tool uses the name that you provide for the operation to define the names for the operation input and the operation output.



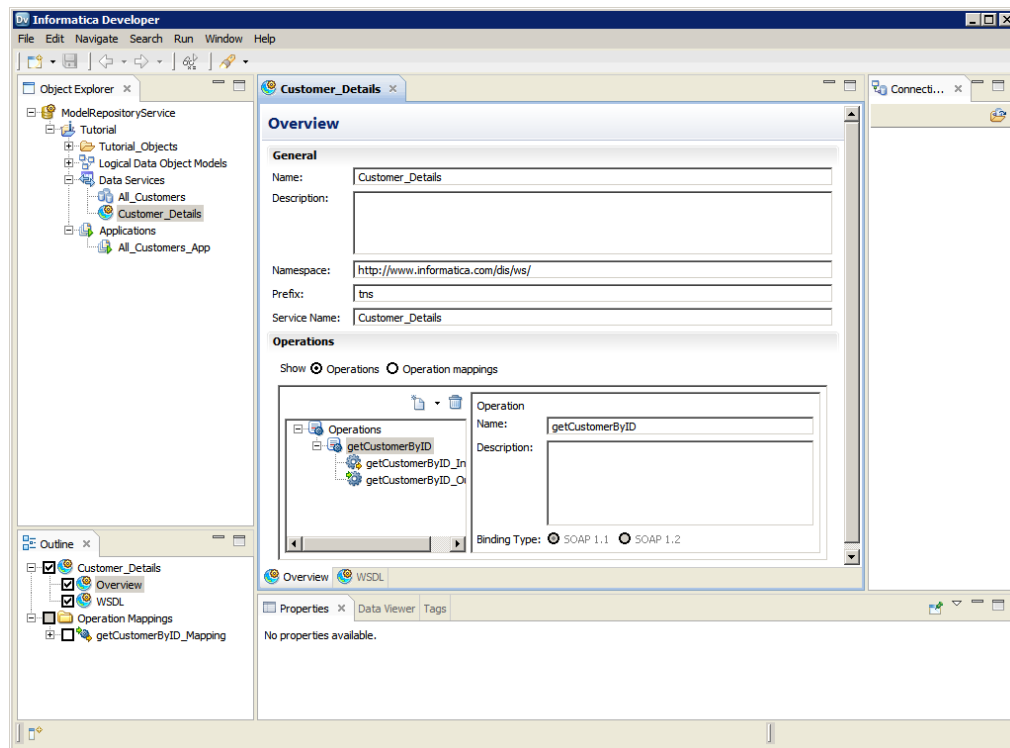
9. Click **getCustomerByID_Input** to select fields from the logical data object.
The Developer tool uses the fields that you select from the logical data object to define elements in the operation input.
10. On the **Operation Input** tab, select **CUSTOMER_ID** for the operation input to include an element for the customer ID.
Because you want the web service to return all customer details, you do not need to select fields for the operation output. By default, all fields are selected on the **Operation Output** tab.
11. To review the input of the operation mapping, select the **Mapping Input** tab.

The following figure shows the input of the operation mapping:



12. To review the output of the operation mapping, select **getCustomerByID_Output** and then select the **Mapping Output** tab.
13. Click **Finish**.

The Developer tool creates the Customer_Details web service and an operation mapping for the operation.

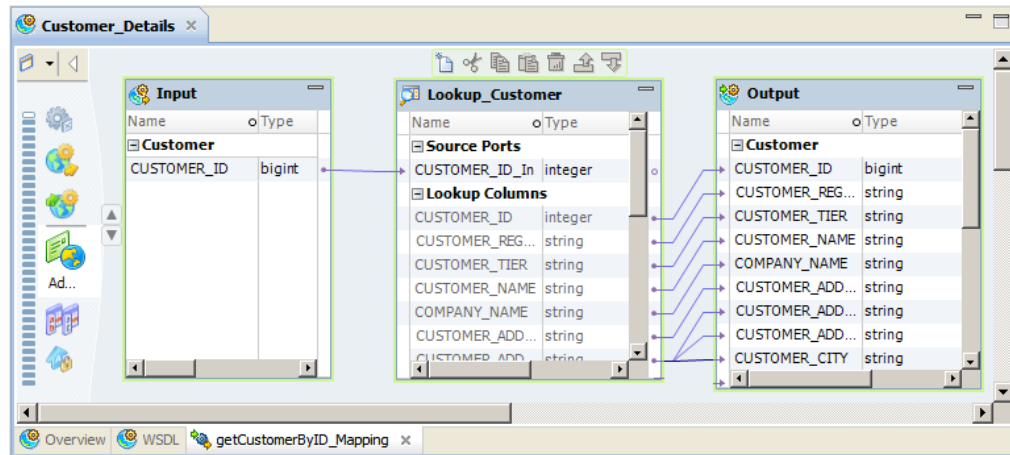


Step 2. View and Validate the Operation Mapping

In this task, you view and validate the operation mapping.

1. In the **Outline** view, expand the Operation Mappings node.
2. In the **Outline** view, select the **getCustomerByID_Mapping** check box to view the operation mapping in the editor.

In the editor, the operation mapping contains an input transformation, a Lookup transformation, and an output transformation.



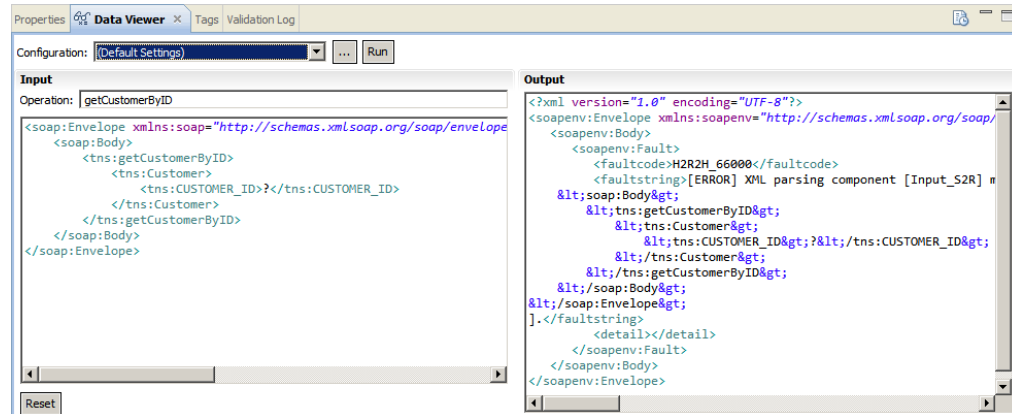
3. Right-click an empty area of the editor and click **Validate**.
The **Validation Log** view appears. The Developer tool displays a message stating whether it found any validation errors.
4. Click **File > Save** to save the web service and operation mapping.

Step 3. Test the Operation Mapping

In this task, you test the operation mapping. You provide a customer ID as input and previewing the customer details in the output. In the **Data Viewer** view, you can enter a SOAP request in the Input window and view the SOAP response in the Output window.

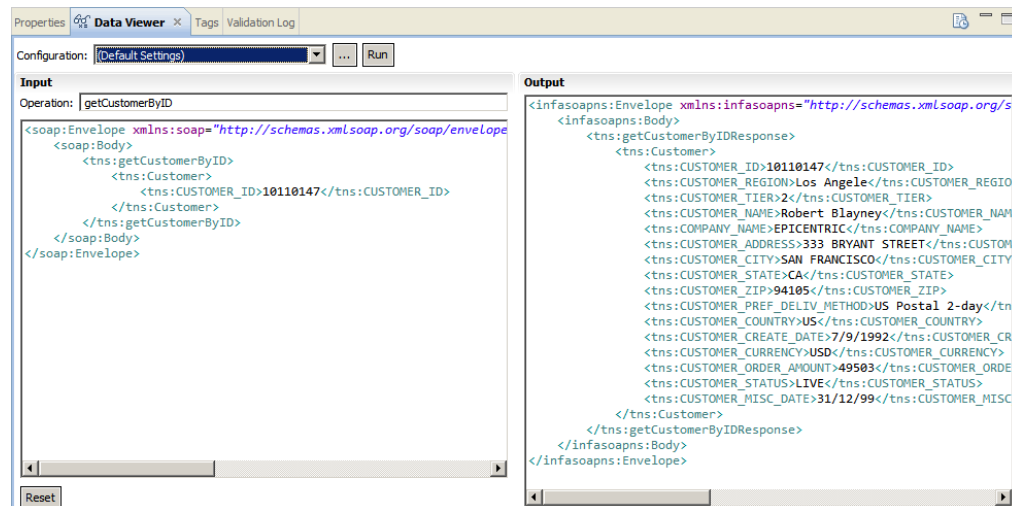
1. Right-click an empty area in the editor, and select **Run Data Viewer**.

The Data Integration Service runs the operation mapping. The operation returns an error because you did not provide the customer ID as input.



2. In the **Input** window, replace the question mark (?) with 10110147. 10110147 is a customer ID.
3. Click **Run**.

The Output window displays the SOAP response based on the customer ID you entered.



Step 4. Create an Application

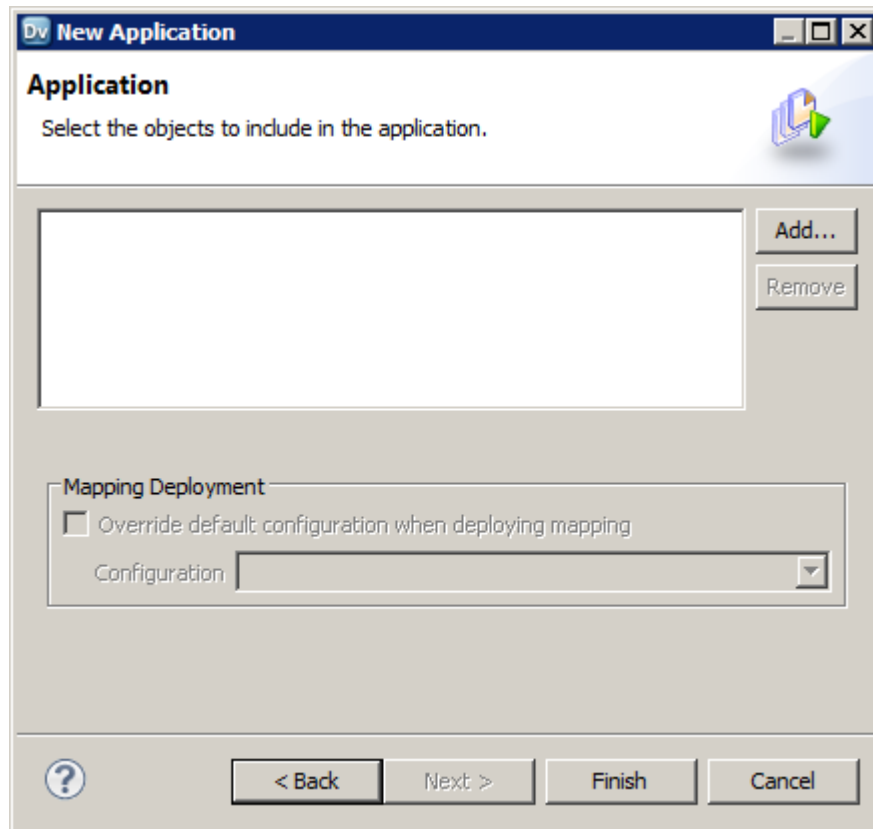
In this task, you create an application that contains the web service. You must add the web service to an application to deploy it to a Data Integration Service.

1. In the **Object Explorer** view, select the Tutorial project.
2. Click **File > New > Application**.

The **New Application** dialog box appears.

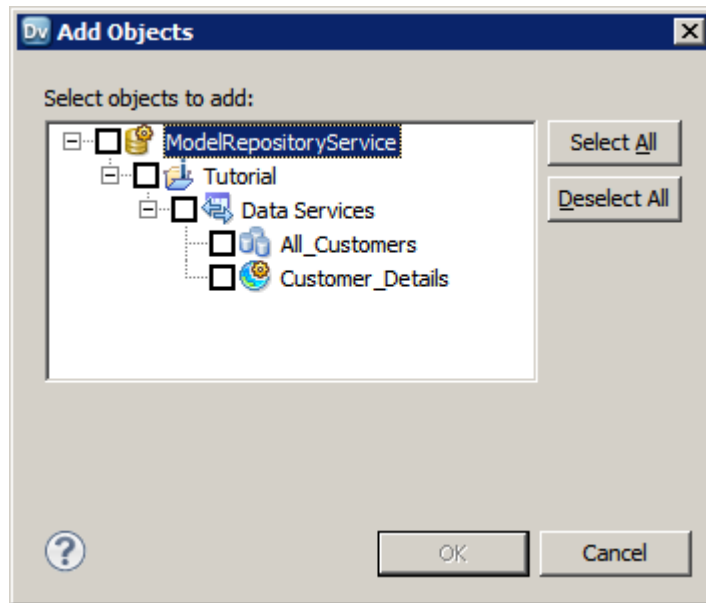
3. Enter `Customer_Details_App` for the application name and click **Next**.

The New Application dialog box shows that the application contains no objects.



4. Click **Add**.

The **Add Objects** dialog box appears.



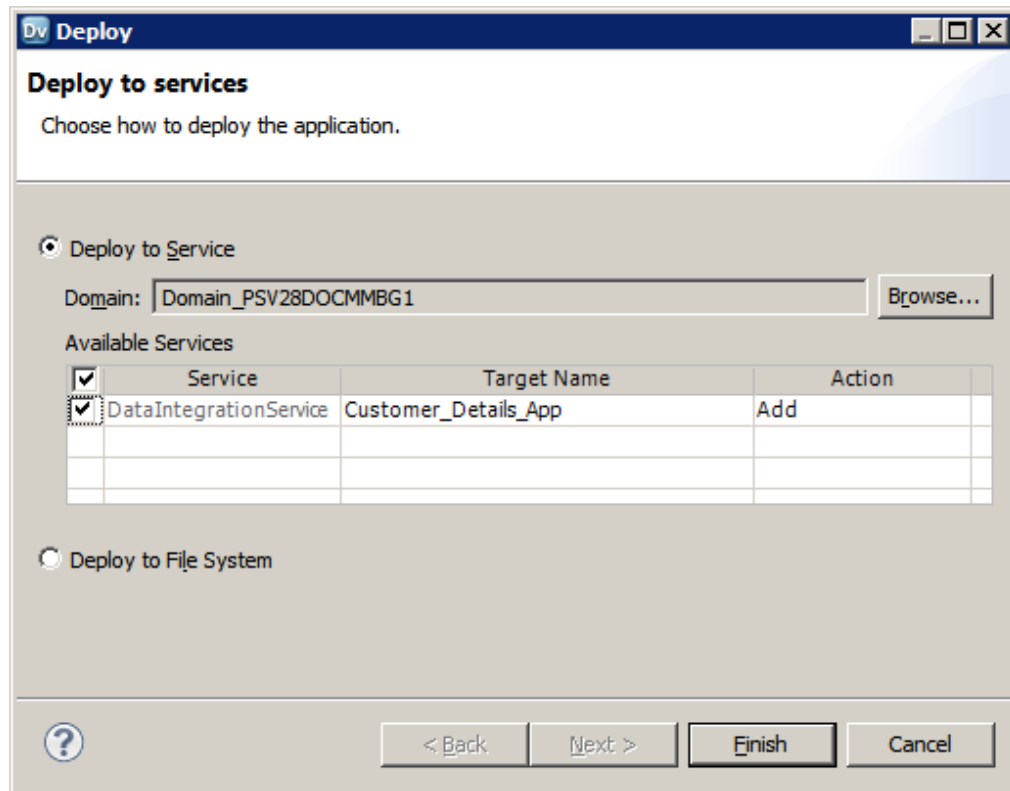
5. Select the **Customer_Details** check box and click **OK**.
The **Customer_Details** web service appears in the **New Application** dialog box.
6. Click **Finish**.
The Developer tool creates an application that contains the **Customer_Details** web service.

Step 5. Deploy the Application

In this task, you deploy the application to a Data Integration Service. You deploy the application to enable a web service client to connect to the web service.

1. In the **Object Explorer** view, expand **Applications** in the **Tutorial** project.
2. Right-click the **Customer_Details_App** application and click **Deploy**.
The **Deploy** dialog box appears.
3. Click **Browse** next to the **Domain** option.
4. Select a domain and click **OK**.
5. Select the Data Integration Service that you want to deploy the application to.

The following figure shows the selected domain and Data Integration Service:



6. Click **Finish**.

When the deployment succeeds, the **Deploy Completed** dialog box appears.

7. Click **OK**.

After you deploy the application, you can view the WSDL URL in the Administrator tool.

Informatica Developer Tips

Use the following tip when you create a web service.

Create an operation for an existing web service.

To create an operation for an existing web service, open the web service, right-click the web service in the **Object Explorer** view or **Outline** view, and then select **New > Operation**.

CHAPTER 8

Exporting a Mapping to PowerCenter

This chapter includes the following topics:

- [Exporting a Mapping to PowerCenter Overview, 80](#)
- [Step 1. Export a Mapping to PowerCenter, 81](#)
- [Informatica Developer Tips, 84](#)

Exporting a Mapping to PowerCenter Overview

You can export a logical data object read mapping from a Model repository to a PowerCenter repository. Export the mapping to PowerCenter when you want to physically move data from one location to another.

Lesson Concepts

You can export mappings and mapplets from a Model repository to a PowerCenter repository. You export the objects to run them in PowerCenter.

Before you use PowerCenter to build a data warehouse, you can use Data Services to build a data warehouse prototype. In the prototype, you can build logical data objects in a logical data object model to describe and relate enterprise entities such as customer and order. For each logical data object, you can build a logical data object read mapping to make data in physical data objects accessible in the logical data object.

If performance or usage needs increase, you can replace the prototype with a physical data warehouse. To populate the physical data warehouse using the transformation logic that you built in the logical data object read mappings, export the mappings to PowerCenter. When you export the logical data object read mappings, the Developer tool converts them into PowerCenter mapplets that you can use to load the physical data warehouse.

Lesson Objectives

In this lesson, you complete the following beginner-level task:

- Export a logical data object read mapping to a PowerCenter repository.

Lesson Prerequisites

Before you start this lesson, complete the following prerequisites:

- Set up Informatica Developer. For more information, see [“Setting Up Informatica Developer Overview” on page 25](#).

- Import the physical data objects. For more information, see [“Importing Physical Data Objects Overview” on page 35](#).
- Create the Customer_Order logical data object model. For more information, see [“Creating a Logical View of Data Overview” on page 44](#).
- Verify that you can connect to the PowerCenter repository into which you want to export the Developer tool mapping. To get the repository login information, contact a domain administrator.

Lesson Timing

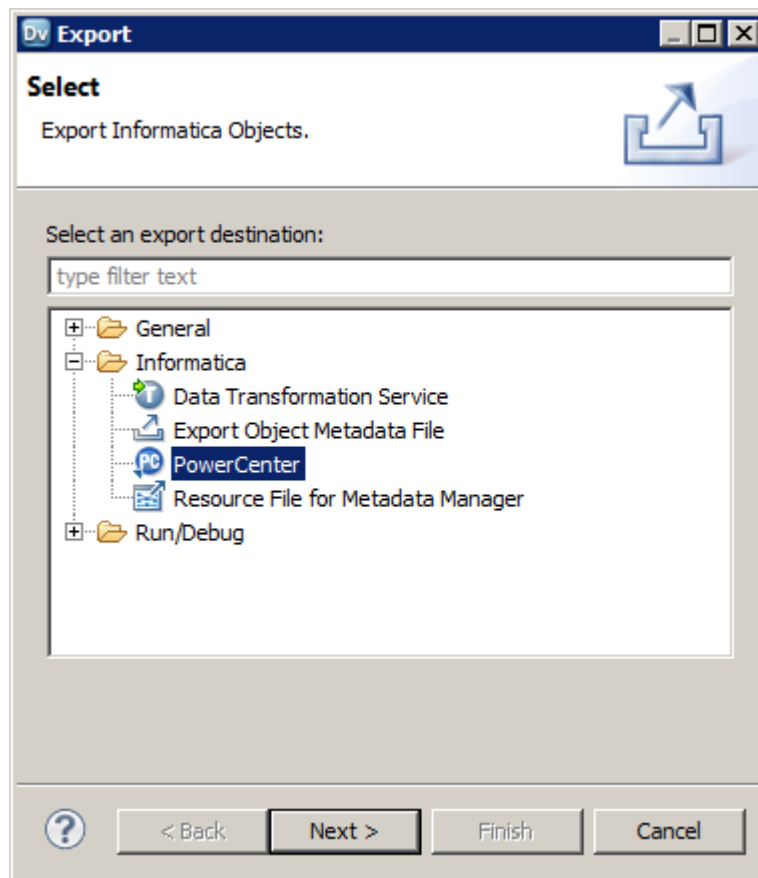
Set aside 5 to 10 minutes to complete this task.

Step 1. Export a Mapping to PowerCenter

In this task, you export a mapping that you created in the Developer tool to a PowerCenter repository.

1. In the **Object Explorer** view, select the Tutorial project.
2. Click **File > Export**.

The **Export** dialog box appears.



3. Select **Informatica > PowerCenter**.
4. Click **Next**.

The **Export to PowerCenter** dialog box appears.

Export To PowerCenter

Please enter a destination archive file.

Project: ModelRepositoryService/Tutorial Browse...

Target release: PowerCenter 9.1.0

Export selected objects to:

☒ File Browse...

☐ PowerCenter Repository Browse...

☐ Send to repository folder: Browse...

☐ Use control file: Browse...

Export settings

☐ Convert exported mappings to PowerCenter mapplets

☐ Convert Target mapplets

☐ Export Reference Data

Reference Data Location: Browse...

Code Page: MS Windows Latin 1 (ANSI), superset of Latin1

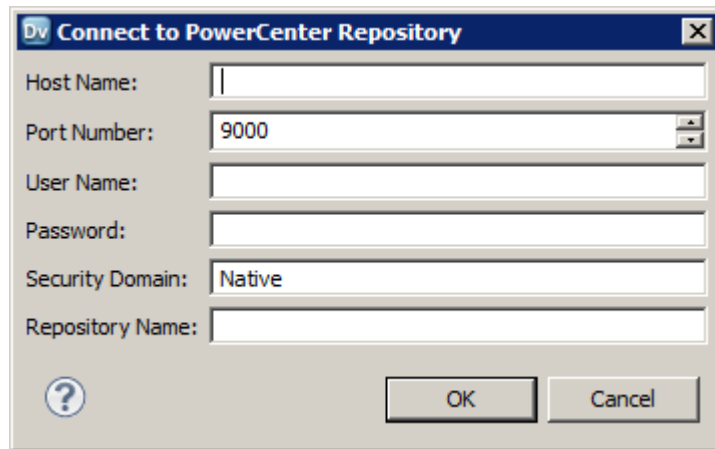
? < Back Next > Finish Cancel

5. In the **Project** field, select the project from which you want to import objects.
6. In the **Target Release** field, select the version of the PowerCenter repository in to which you want to import the objects.

Different versions of PowerCenter store metadata differently. Select the version of PowerCenter to ensure that the mapping metadata is imported into the PowerCenter repository correctly.

7. In the **Export Selected Objects To** field, select `PowerCenter Repository` to export the objects to a PowerCenter repository.
8. Click **Browse** next to the **PowerCenter Repository** field to enter the connection properties for the PowerCenter repository.

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

The image shows a Windows-style dialog box titled "Connect to PowerCenter Repository". It has a standard title bar with a close button (X). The dialog contains several input fields: "Host Name:" with an empty text box, "Port Number:" with a text box containing "9000" and a small spinner control to its right, "User Name:" with an empty text box, "Password:" with an empty text box, "Security Domain:" with a text box containing "Native", and "Repository Name:" with an empty text box. At the bottom left is a help icon (question mark in a circle). At the bottom right are two buttons: "OK" and "Cancel".

9. Enter the following connection information:

- Host name. Informatica domain gateway host name.
- Port number. Informatica domain gateway HTTP port number.
- User name. Repository user name.
- Password. Password for repository user name.
- Security domain. LDAP security domain name, if one exists. Otherwise, enter "Native."
- Repository name. PowerCenter repository name.

10. Click **OK**.

The Developer tool displays a message that indicates whether it was able to successfully connect to the repository.

Note: If you select the wrong PowerCenter version, the connection fails.

11. Click **OK**.

The **Export to PowerCenter** dialog box appears.

12. In the **Send To Repository Folder** field, select the repository folder that you want to export the mapping to.

If the repository contains a folder for tutorials, select it.

13. Click **Next**.

The Developer tool prompts you to select the objects to export.

14. Select Customer_Order and click **Finish**.

The Developer tool exports the object to the PowerCenter repository folder that you selected.

Informatica Developer Tips

Use the following tip when you export a mapping to PowerCenter:

Before you export the Developer tool objects to PowerCenter, validate them against the PowerCenter version.

Before you export the Developer tool objects to PowerCenter, you can validate whether the Developer tool objects are compatible with a particular PowerCenter version.

To enable validation, set the compatibility level to a particular PowerCenter version. To disable validation, do not select a PowerCenter version. To set the compatibility level, click **Edit > Compatibility Level**.

APPENDIX A

Glossary

application

A deployable object that can contain data objects, mappings, SQL data services, web services, and workflows.

cost-based optimization

Optimization method that reduces the run time for mappings that perform join operations. With cost-based optimization, the Data Integration Service creates different plans to run a mapping and calculates a cost for each plan. The Data Integration Service runs the plan with the smallest cost. The Data Integration Service calculates cost based on database statistics, I/O, CPU, network, and memory.

customized data object

A physical data object that uses one or more related relational resources or relational data objects as sources. You use a customized data object to perform tasks such as join data from related resources or filter rows. Customized data object uses a single connection and SQL statement for the source tables.

Data Integration Service

An application service that performs data integration jobs for Informatica Analyst, Informatica Developer, and external clients. Data integration jobs include previewing data and running mappings, profiles, SQL data services, web services, and workflows.

data service

A collection of reusable operations that you can run to access and transform data. A data service provides a unified model of data you can access through a web service or run an SQL query against.

deploy

To make objects within an application accessible to end users. Depending on the types of objects in the application, end users can then run queries against the objects, access web services, run mappings, or run workflows.

early projection optimization

Optimization method that reduces the amount of data that moves between transformations in the mapping. With early projection optimization, the Data Integration Service identifies unused ports and removes the links between the ports in a mapping.

early selection optimization

Optimization method that reduces the number of rows that pass through the mapping. With early selection optimization, the Data Integration Service moves filters closer to the mapping source in the pipeline.

folder

A container for objects in the Model repository. Use folders to organize objects in a project and create folders to group objects based on business needs.

Informatica Administrator

Informatica Administrator (the Administrator tool) is an application that consolidates the administrative tasks for domain objects such as services, nodes, licenses, and grids. You manage the domain and the security of the domain through the Administrator tool.

Informatica Developer

Informatica Developer (the Developer tool) is an application that you use to design data integration solutions. The Model repository stores the objects that you create in the Developer tool.

Informatica Monitoring tool

Informatica Monitoring tool (the Monitoring tool) is an application that provides a direct link to the Monitor tab of the Administrator tool. The Monitor tab shows properties, run-time statistics, and run-time reports about the integration objects that run on a Data Integration Service.

logical data object

An object that describes a logical entity in an organization. It has attributes and keys, and it describes relationships between attributes.

logical data object mapping

A mapping that links a logical data object to one or more physical data objects. It can include transformation logic.

logical data object model

A data model that describes data in an organization and the relationship between the data. It contains logical data objects and defines relationships between them.

logical data object read mapping

A mapping that provides a view of data through a logical data object. It contains one or more physical data objects as sources and a logical data object as the mapping output.

logical data object write mapping

A mapping that writes data to targets using a logical data object as input. It contains one or more logical data objects as input and a physical data object as the target.

mapping

A set of inputs and outputs linked by transformation objects that define the rules for data transformation.

mapplet

A reusable object that contains a set of transformations that you can use in multiple mappings or validate as a rule.

Model Repository Service

An application service in the Informatica domain that runs and manages the Model repository. The Model repository stores metadata created by Informatica products in a relational database to enable collaboration among the products.

node

A representation of a level in the hierarchy of a web service message.

operation mapping

A mapping that performs the web service operation for the web service client. An operation mapping can contain an Input transformation, an Output transformation, and multiple Fault transformations.

physical data object

A physical representation of data that is used to read from, look up, or write to resources.

predicate expression

An expression that filters the data in a mapping. A predicate expression returns true or false.

predicate optimization

Optimization method that simplifies or rewrites the predicate expressions in a mapping. With predicate optimization, the Data Integration Service attempts to apply predicate expressions as early as possible to increase mapping performance.

project

The top-level container to store objects created in Informatica Analyst and Informatica Developer. Create projects based on business goals or requirements. Projects appear in both Informatica Analyst and Informatica Developer.

pushdown optimization

Optimization method that pushes transformation logic to a source or target database. With pushdown optimization, the Data Integration Service translates the transformation logic into SQL queries and sends the SQL queries to the database. The database runs the SQL queries to process the data.

result set caching

A cache that contains the results of each SQL data service query or web service request. With result set caching, the Data Integration Service returns cached results when users run identical queries. Result set caching decreases the run time for identical queries.

semi-join optimization

Optimization method that reduces the number of rows extracted from the source. With semi-join optimization, the Data Integration Service modifies the join operations in a mapping. The Data Integration Service applies the semi-join optimization method to a Joiner transformation when a larger input group has rows that do not match a smaller input group in the join condition. The Data Integration Service reads the rows from the smaller group, finds the matching rows in the larger group, and performs the join operation.

SQL data service

A virtual database that you can query. It contains virtual objects and provides a uniform view of data from disparate, heterogeneous data sources.

SQL Service Module

The component service in the Data Integration Service that manages SQL queries sent to an SQL data service from a third-party client tool.

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.

transformation

A repository object in a mapping that generates, modifies, or passes data. Each transformation performs a different function.

virtual data

The information get when you query virtual tables or run stored procedures in an SQL data service.

virtual database

An SQL data service that you can query. It contains virtual objects and provides a uniform view of data from disparate, heterogeneous data sources.

virtual schema

A schema in a virtual database that defines the database structure.

virtual stored procedure

A set of procedural or data flow instructions in an SQL data service.

virtual table

A table in a virtual database.

virtual table mapping

A mapping that contains a virtual table as a target.

virtual view of data

A virtual database defined by an SQL data service that you can query as if it were a physical database.

Web Service Module

A component in the Data Integration Service that manages web service operation requests sent to a web service from a web service client.

web service transformation

A transformation that processes web service requests or web service responses. Examples of web service transformations include an Input transformation, Output transformation, Fault transformation, and the Web Service Consumer transformation.