Version 2 Release 2 Modification 3

IBM InfoSphere Optim
Using Optim Designer

Version 2 Release 2 Modification 3

IBM InfoSphere Optim
Using Optim Designer

Note
Before using this information and the product it supports, read the information in "Notices" on page 121.

First Edition

This edition applies to version 2, release 2, modification 3 of Optim Designer and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Optim Designer overview . . 1
What's new in Optim Designer 2
Getting started. 3
Creating a data design project 3
Masking data in a relational database 4
Using Optim Manager from Optim Designer . . 4
Optim Perspective 5
Sample Data 7
Database support 8
Accessibility Features 9

Chapter 2. Defining a database connection 11
Optim data sources 11
Working with native data source connections . . . 11
Using a lookup data source 12
Defining a distributed lookup data source . . . 12
Defining a z/OS lookup data source 12
Defining an executor lookup data source . . . 13

Chapter 3. Managing data models . . . 15
Physical data models based on reverse engineering 15
Defining a physical data model based on reverse engineering 15
Using Database Relationship Analyzer physical data models 15
Transforming a physical data model to an Optim logical data model 17
Converting a schema in a logical data model to an Optim logical data model. 17
Using data access plans 18
Creating a data access plan 18
Editing a data access plan 18
Working with data sources in a data access plan 18
Working with a selection policy 18

Chapter 4. Designing data management services 23
Working with executor services. 23
Creating an executor service. 23
Editing an executor service 24
Testing executor services 28
Working with Optim interoperability services . . . 29
Working with Optim interoperability services on Linux, UNIX, and Windows 29
Working with Optim interoperability services on z/OS 36
Editing an Optim interoperability service . . . 40
Testing an Optim interoperability service . . . 41
Working with the Optim registry 42
Entering a default Optim registry location . . . 42
Publishing a service 42
Establishing a secure connection 43
Exporting a service to a file system 43

Chapter 5. Using data privacy policies 45
Data privacy policies 45
Date privacy policies 45
Identity privacy policies 48
Numeric privacy policies 61
Scramble privacy policies. 66
Generic lookup privacy policies 72
Random shuffle function 75
JavaScript policies 75
Data privacy compliance requirements 78
Using the data privacy editor 78
Editing a data privacy policy 79

Chapter 6. Using Optim Designer with your Optim Solution 81
Using Optim Designer with Optim interoperability services on a distributed platform 81
Creating a data design project 82
Connecting to the Optim sample database . . . 82
Creating a physical data model based on reverse engineering 84
Transforming a schema in a physical data model to an Optim logical data model. 85
Creating a data access plan and a selection policy 87
Defining selection criteria. 89
Defining a data privacy policy to mask credit card numbers. 91
Creating an Optim interoperability service . . . 92
Using Optim Designer with Optim interoperability services on a z/OS platform. 94
Creating a data design project 95
Connecting to the Optim sample database . . . 95
Creating a physical data model based on reverse engineering 97
Transforming a schema in a physical data model to an Optim logical data model. 99
Creating a data access plan and a selection policy 100
Defining selection criteria 102
Defining a data privacy policy to mask credit card numbers 104
Creating an Optim interoperability service. . . 105
Using Optim Designer with Optim Data Masking Solution 107
Creating a data design project. 107
Connecting to the Optim sample database. . . 108
Creating a physical data model based on reverse engineering 109
Transforming schemas in a physical data model to an Optim logical data model 111
Creating a data access plan and a selection policy 112
Defining selection criteria 114
Defining a data privacy policy to mask credit card numbers 116

Defining a data privacy policy to mask numeric data 117
Defining an executor service to copy and transform data 119

Index 125

Chapter 1. Optim Designer overview

IBM® Optim™ Designer allows you to define data models, data privacy policies, and data management services. You can use Optim Designer to run Optim and Optim for z/OS® requests. You can also use Optim Designer to create and test data management services and Optim interoperability services.

Optim logical data models and physical data models
To define a data management service, you must use an Optim logical data model to define the source or target data. You can create a new Optim logical model by transforming a physical data model.

Data management services
Use an Optim data management service to transform data, copy data between schemas, or run Optim and Optim for z/OS requests. You can mask data by applying a data privacy policy to an entity processed by a service. There are two types of services: executor services (for the Optim Executor platform) and Optim interoperability services (for the Optim and Optim for z/OS platforms). You can publish services to the Optim Manager environment, where you can run or schedule services. You can also use Optim Manager in embedded mode to test services.

Data privacy policies
Data privacy policies allow you to mask data in a data management service. There are three options to mask data with a privacy policy: lookup, rule-based, and JavaScript. The lookup option uses a lookup table to provide masked data. The rule-based option uses functions to generate masked data. The JavaScript option uses JavaScript expressions to define a data transformation and is available for use with executor services only.

These policies offer the following features:

- Use lookup functions to replace values from selected source entities with values from corresponding lookup table columns
- Use rule-based functions to mask national ID numbers, credit card numbers, and e-mail addresses with valid and unique values
- Use rule-based functions to generate values for dates, characters, and numbers
- Apply a lookup or rule-based function based on a "switch" value
- Use JavaScript to define custom transformations in an executor service

Data access plans
A data access plan contains policies that determine which data to process or transform from a source Optim logical data model in a data management service. You can use a data access plan to define a selection policy and data privacy policies for a logical data model. A selection policy determines the entities and attributes to use in a data management service.

Optim interoperability services
Optim interoperability services allow you to process Optim and Optim for z/OS requests. The requests are defined in an interoperability service and use data models created in Optim Designer. The data models can include data privacy policies. You can connect to an Optim directory and import or export Optim definitions. You can test services in Optim Designer and publish services to the Optim Manager environment.

Executor services
Executor services allow you to transform data as well as copy data between schemas. The services use the Optim Executor platform for processing. You can mask data by applying a data privacy policy to an entity processed by the service. You can test services in Optim Designer and publish services to the Optim Manager environment.

Optim Manager
You can open Optim Manager from Optim Designer (embedded mode), allowing you to test and publish data management services.

What's new in Optim Designer

Version 2, Release 2 of IBM Optim Designer provides the following enhancements:

- Support for convert requests for IBM InfoSphere™ Optim 8.1 and Optim for z/OS 7.1
- Support for the creation of DB aliases
- Support for the creation and publishing of an Optim interoperability model (OIM) as a service
- Capability to define Optim Servers
- Capability to select Optim Servers in Optim service wizards via drop-down lists
- History lists for all Optim and Optim for z/OS service wizards
- Local file system browse for all Optim and Optim for z/OS service wizards
- Capability to create an Optim or Optim for z/OS request from within a service definition
- New data access plan editor
- Data privacy classification and enforcement through the use of domain models
- Capability to use the IBM Optim Manager interface to perform the following tasks on services in an Optim Designer workspace:
  - Execute services
  - Publish services to registries
  - Export services to files
- Usability and accessibility features that make the user interface easier to see, read, and use
- Native data source support for IBM Informix® and IBM DB2® for z/OS
- Support for Teradata V2.6, V12, and V13
- Additional locales for scramble policies
- Addition of Blue Print Director to the Optim Designer launchpad
- Capability to install all Optim components simultaneously.

What's new in IBM InfoSphere Optim Data Masking Solution

Version 2, Release 2 of IBM InfoSphere Optim Data Masking Solution provides enhancements to Optim Designer, Optim Manager, Optim Management Server, Optim Proxy, and Optim Executor. An installation launch pad is provided for Optim Data Masking Solution.

Enhancements to Optim Designer

Optim Designer provides the following enhancements:

- Support for convert requests for Optim 7.3 and Optim for z/OS 7.1
- Support for the creation of DB aliases
- Support for the creation and publishing of an Optim interoperability model (OIM) as a service
- Ability to define Optim Servers
- Ability to select Optim Servers in the OIM wizards via drop-down lists.
- History lists for all OIM wizards
- Local file system browse for all Optim OIM wizards
- New data access plan editor
- Data privacy classification and enforcement through the use of domain models

- Ability to use the Optim Manager interface to perform the following tasks on services in an Optim Designer workspace:
  - Execute services
  - Publish services to registries
  - Export services to files
- Usability and accessibility features that make the user interface easier to see, read, and use
- Native data source support for IBM Informix and IBM DB2 for z/OS
- Additional locales for scramble policies.

Enhancements to Optim Manager, Optim Management Server, and Optim Proxy

Optim Manager, Optim Management Server, and Optim Proxy provide the following enhancements:

- Integration support for Optim 7.3 and Optim for z/OS 7.1, which provides the capability to manage services from the Optim Manager web interface
- Capability to change and save published parameters for service plan, selection policy, lookup database, and native database driver parameters prior to execution
- Automated startup for Optim Manager and Optim Management Server on Microsoft Windows when using IBM WebSphere® Application Server Community Edition
- Automated startup for Optim Proxy on Windows
- Improved error-message content
- Usability and accessibility features that make the user interface easier to see, read, and use
- Added support to detect and display missing proxy capabilities required to run a service
- Added right-click menus to parts of the Optim Manager user interface
- Added support to import services from a file to a registry using the Optim Manager web interface
- Added support to promote a service from one registry to another using the Optim Manager web interface.

Enhancements to Optim Executor

Optim Executor provides the following enhancements:

- Support for release 2.2 services
- Improved error-reporting capability.

Getting started

To get started, you must create a data design project in the Data Project Explorer. You can use the project to create objects that will allow you to mask relational data.

Creating a data design project

Before you create data models or other data design objects, create a data design project to store your objects.

A data design project is primarily used to store modeling objects. You can store the following types of objects in a data design project:

- Logical data models
- Physical data models
- Data management service requests
- Optim interoperability models
- Domain models

- Glossary models
- SQL scripts, including DDL scripts
- JCL Files from z/OS job requests
- In some products, if you have Information Integrator installed: Mapping models and XML schemas

Any other file types, such as doc files, text files, presentations, or spreadsheets can also be stored in a data design project. Any files other than the files listed above or Eclipse-specific files (such as .project files) are displayed in the **Other files** folder under a data design project in the Data Project Explorer.

You do not have to create a database connection in the Data Source Explorer before you create a data design project. However, some of the actions you typically perform in a data design project (for example, reverse engineering a physical data model) do require a database connection.

Using the New Data Design Project wizard, you specify basic information about the data design project including the name and the local directory in which to store files. You can also specify project references.

To create a data design project:

1. On the main menu bar, click **File** > **New** > **Data Design Project**. As an alternative, right-click in any blank space in the Data Project Explorer and select **New** > **Project** > **Data Design Project**. The New Data Design Project wizard opens.
2. Complete the steps of the wizard, then click **Finish**. The data design project is displayed in the Data Project Explorer view.

Masking data in a relational database

You can use Optim Designer to define a data privacy policy to mask sensitive relational data.

To mask data in a relational database:

1. Define a data source connection to a relational database, as described in Chapter 2, “Defining a database connection,” on page 11.
2. Define a physical data model based on reverse engineering from a database, as described in “Physical data models based on reverse engineering” on page 15.
3. Define an Optim logical data model, as described in “Transforming a physical data model to an Optim logical data model” on page 17.
4. Define a data access plan, as described in “Using data access plans” on page 18.
5. Define a selection policy, as described in “Working with a selection policy” on page 18.
6. Define a privacy policy, as described in Chapter 5, “Using data privacy policies,” on page 45.

   After you define a privacy policy, you can mask data by using the Optim logical data model with a data management service.

Using Optim Manager from Optim Designer

Use Optim Manager from Optim Designer to publish and test data management services and Optim interoperability services. Using Optim Manager from Optim Designer is also known as Optim Manager in embedded mode.

Optim Manager is a web application that you can use to test services before publishing them to the registry. Optim Manager is displayed either in the internal browser provided by Optim Designer or in an external browser. You can select the browser that you want Optim Designer to use for Optim Manager by clicking **Window** > **Preferences** > **General** > **Web Browser** in Optim Designer.

Opening Optim Manager from Optim Designer

You can open Optim Manager by publishing or executing a service. You can also open Optim Manager by entering the following URL in a web browser: `http://localhost:portnumber/console`, where *portnumber* is the port number assigned to Optim Manager. The default port number is 60000.

If the browser displays a message about a page not found when you open Optim Manager, a port conflict may exist and you must change the Optim Manager port number.

Changing the Optim Manager port number

To change the port number, you must edit the following property in the eclipse.ini file located in the default InfoSphere Data Architect installation directory:

`-Dorg.eclipse.equinox.http.jetty.http.port=portnumber`

where *portnumber* is the new Optim Manager port number. If Optim Designer is open, you must restart the application to apply the new port number.

Optim Perspective

In Optim Designer, the Optim perspective provides tools you need to define data models and privacy policies. When you first open Optim Designer after installation, the Optim perspective is the default display.

The Optim perspective includes the following views:

Data Project Explorer
: Use the Data Project Explorer to define data objects, privacy policies, Optim interoperability models, and data management services.

Data Source Explorer
: Use the Data Source Explorer to define connections to data sources.

To return to the Optim perspective after navigating away from it, click **Window** > **Open Perspective** > **Other**. In the Open Perspective window, select **Optim**.

Data Project Explorer

In the Data Project Explorer, you can work locally with data objects.

The Data Project Explorer displays the following projects:

Data design projects
: Data design projects are used for database design and information integration. Use this type of project to develop physical data models, logical data models, domain models, glossary models, XSD models, and scripts.

  - Use physical data models based on reverse engineering from a relational database to create an Optim logical data model. Physical data models can be used to generate DDL statements that can be deployed to a database server.
  - Use an Optim logical data model, which is a logical data model that includes a data access plan. A data access plan includes policies for selecting and masking data. Logical data models are not specific to a database that describes things about which an organization wants to collect data, and the relationships among these things. You can generate physical data models or UML models from logical data models.
  - Use service requests to define a data management service.
  - Use Optim interoperability models to define and process Optim requests for Optim and Optim for z/OS.

- Use domain models to describe an organization's allowed atomic domain types and their constraints. You can specify atomic domains as data types for physical and logical data models. Atomic domains can also be specified as an integrated part of a logical data model.
- Use glossary models to validate a data model for naming standard compliance, or to determine naming conventions.

Data development projects

Data development projects are used for containing an Optim directory and database application development. This type of project is associated with a single connection in the Data Source Explorer. Use data development projects to perform the following tasks:

- You can import and manage Optim directories.
- You can develop, test, and deploy stored procedures and user-defined functions.
- If the target server supports XML, you can develop XML files and artifacts for XML applications.
- You can also develop and test SQL queries.
- You can develop and deploy Web services that access data by using SQL scripts or stored procedures.

Using the Data Project Explorer, you can also perform the following tasks:

- Analyze the impact and dependency of data objects
- Analyze a data model to ensure model integrity
- Compare two data objects
- Generate DDL for data objects or data models
- Drag and drop or copy database objects from the Data Source Explorer or from within the Data Project Explorer
- Share projects using a source control system

Data Source Explorer

In the Data Source Explorer, you can connect to existing databases and view their designs and objects.

You can browse database designs and import them to the Data Project Explorer, where you can extend or modify the designs. You can also run stored procedures and user-defined functions and view the results in the SQL Results view.

Using the Data Source Explorer, you can perform the following tasks. Some of these tasks are not supported in some products that use the Data Source Explorer.

- Create and manage database connections, and browse data objects in a connection.
- Modify data objects, and manage changes.
- Define native data source connections for testing data management services.
- Define local Optim managed data sources.
- Export data object metadata to data projects, where you can modify and redeploy the objects.
- Create, run, and tune SQL queries and routines.

Migrating an Optim Designer workspace from a previous release

You can migrate an Optim Designer workspace from a previous release to the current release.

Workspaces from 2.2.x releases are automatically migrated to the current release when the workspace is first opened by this release. For these workspaces, requests included in Optim interoperability models are converted to Optim interoperability services.

Workspaces from 2.1.x releases must be migrated by using the Migrate Optim Workspace wizard.

To migrate a 2.1.x workspace:

1. Click **Migrate** > **Migrate Optim Workspaces**. The Migrate Optim Workspace wizard opens.
2. Complete the steps of the wizard.

   You must select an empty and existing target directory for the migrated workspace.
3. Open the workspace in the current release to complete the migration.

Sample Data

Optim provides a sample Derby database that contains replacement data as well as predefined source and target data sources.

Optim Sample Database

By default, Optim Designer will automatically run the Derby database that manages the sample data. In the Data Source Explorer, the sample database has the connection name Optim Sample Database.

The sample database is located in the \.metadata\.plugins\com.ibm.nex.designer.ui\database\optim directory of the Optim workspace. The database includes the following schemas:

- OPTIMUSER - Related tables with customer, order, inventory, and shipping data.
- OPTIMUSER2 - A schema with metadata that matches the tables in the OPTIMUSER schema. This schema can be used as a destination when OPTIMUSER is the source schema for a data management service.

Optim Replacement Data

The Optim Replacement Data profile in the Data Source Explorer includes a default connection to the EXTENDED_LOOKUP schema in a local management server installation. A management server installation includes an instance of a database that includes the EXTENDED_LOOKUP schema with the default lookup tables. The default connection for the executor lookup data source is the Optim Replacement Data connection profile.

Lookup Tables

The EXTENDED_LOOKUP schema includes lookup tables that can be used with the Optim lookup policies. Lookup policies processed by the executor platform must include a connection to a database with this schema.

The EXTENDED_LOOKUP schema includes lookup tables for masking personal data such as addresses, names, national ID numbers, birth dates, etc. Each category of personal data is provided in several tables that include country-specific personal data. For example, a table includes American addresses and another table includes German addresses.

The optim\designer\sampledata directory, located in the Infosphere Data Architect installation directory, includes .ddl and .data files that allow you to create the tables in the EXTENDED_LOOKUP schema. Create these tables using the interactive tool provided by your database vendor.

The optim\designer\sampledata directory includes the following subdirectories:

extended_lookup
: Includes .data files for each table in the EXTENDED_LOOKUP schema.

extended_lookup_schemas
: Includes .ddl files for creating the EXTENDED_LOOKUP schema for each database type.

Each category of personal data is provided in a separate table for the following countries (abbreviations are in parentheses): Australia (AU), Canada (CA), France (FR), Germany (DE), Italy (IT), Japan (JP), Spain

(ES), United Kingdom (UK), and United States (US). Each table includes a column of sequential numbers that is used with lookup policies that use hash values to select a row in the lookup table.

In the schema, each table name is composed of a country abbreviation prefix and the category (*countryabbreviation_category*). For example, the address table for Canada is named CA_ADDRESSES and the address table for Germany is named DE_ADDRESSES.

The schema includes the following categories:

ADDRESSES - includes columns for street address, city, locality (e.g., state or province), and postal code.

FIRSTNAME - includes a column with male and female first names.

FIRSTNAME_F - includes a column with female first names.

FIRSTNAME_M - includes a column with male first names.

LASTNAME - includes a column with last names.

PERSON - includes columns for birth date, first name, last name, gender, phone number, national ID number, company name, and e-mail address.

Database support

Optim Designer provides support for multiple database management systems.

Optim Designer supports JDBC connections for the following databases:

- DB2 for z/OS V8.1, V9.1, V10.1
- DB2 for Linux, UNIX, and Windows V8.2, V9.1, V9.5, V9.7
- DB2 for i V5.4
- Informix V10
- Microsoft SQL Server 2005, 2008
- Oracle V10.2, V11, V11.2
- Sybase V12.5, V15
- Teradata V2.6, V12, V13

Optim Designer supports native data source connections for the following databases:

- DB2 for Linux, UNIX, and Windows V9.1, V9.5
- IBM DB2 for z/OS V9.1
- IBM Informix V11.5
- Oracle V10.2

DB2 Prerequisites

To allow Optim to obtain full JDBC metadata from an instance of DB2 z/OS, the DESCSTAT value in ZPARMS must be set to YES. Also, you must run job DSNTIJMS in order to install the stored procedures needed by JDBC, bind the necessary packages, and set security permissions. In addition, workload manager (WLM) definitions are needed to ensure that the WLM can start the stored procedure address space when requested by DB2.

Accessibility Features

Accessibility features help people with a physical disability, such as restricted mobility or limited vision, or with other special needs, to use software products successfully.

Optim Designer uses accessibility features available with the Eclipse environment.

Accessibility features help people with a physical disability, such as restricted mobility or limited vision, or those with special needs to use software products successfully. The following is a list of the major accessibility features in Optim Designer:

- You can view the objects and hierarchies of a data diagram in the Data Source Explorer.
- You can use the Outline view to navigate through the mapping editor, and find additional information in the Properties view. Some actions are only available from the mapping editor. Select the mapping in the Outline view, then go to the mapping editor, and right-click to invoke the menu items.
- Some read-only fields in the Properties view cannot be read by a screen reader. You can find information about these fields in the Data Source Explorer. When you highlight an object in the Data Source Explorer, some of the information in the Properties view for the object is still read-only. Because the information is read-only, screen readers cannot read it. To work around this issue, you can copy and paste the object from the Data Source Explorer into a data design project in the Data Project Explorer, then highlight the object, open the Properties view, and the screen reader can read all of the fields.
- To draw relationships in a data diagram, select two objects, then tab to the palette, and select a relationship object. To specify direction, select the "from" object first, and then select the "to" object.
- To get a screen reader to read object names in a data diagram, select the object, and press F2 to put the object name into edit mode. The screen reader reads the object name. Press Esc to get out of edit mode.
- All of the information presented in a data diagram is also available in the Data Project Explorer, Data Source Explorer, and Properties view.
- The graphical icons have tooltips. Reading the tooltips depends on the screen reader that you use.

Chapter 2. Defining a database connection

Use the Data Source Explorer to define a database connection.

Optim Designer uses JDBC to connect directly to databases. To use a data model in a data management service, the model must be associated with a JDBC connection profile.

Optim Designer allows creation of services with native data source connection for faster run time performance. To use a native connection in the designer, a client for the database must be installed on the Optim Designer machine.

Optim data sources

An Optim data source contains JDBC and native connection properties (if available) for a data source connection.

An Optim data source is created when a database connection is first associated with an Optim logical data model. A database connection can be associated with only one Optim data source. Each Optim logical data model that is associated with the same database connection will use the same Optim data source.

If a data source connection is updated, you can use the data access plan editor or service editor to refresh the associated Optim data source.

The lookup data sources used with the data privacy lookup policies each use a default Optim data source name.

Working with native data source connections

By default, Optim data sources use a JDBC connection for relational databases. For faster processing, you can define a native data source connection for an Optim data source. Native data source connections are not available for all supported databases.

Native data source connection properties apply to the associated Optim data source.

A native data source connection is based on a database client connection. To use a native database client connection, a client for the database must be installed on the Optim Designer machine.

If a native data source connection is available, you can define or edit a native data source connection from the following locations:

- Transform to Optim Logical Data Model wizard
- New Service wizard
- Data access plan editor
- Service plan editor

To define a native data source, you must provide the connection string and character set for the database as well as the credentials for the user who will execute data management services that use the data source.

Using a lookup data source

Use the Optim preferences to define a lookup data source for the generic lookup and identity privacy policies.

Before you can create a privacy policy that uses lookup data, you must define a lookup data source for the target platform of the policy.

When you create a privacy policy that uses lookup data, the Add Policy wizard will use metadata from the lookup data source to define the policy. If a you create a policy for a platform without a lookup data source definition, you will be prompted to define the lookup data source.

Defining a distributed lookup data source

Use the Optim preferences to define a lookup data source for the distributed platform.

A database connection is required. You can add the connection in the Data Source Explorer view or create a connection when you define the lookup data source.

You will need to identify a DB alias and schema for the lookup data.

To define a lookup data source for the distributed platform:

1. Click **Window** > **Preferences**. The Preferences window opens.
2. In the navigation tree, expand the **Optim** node and then click **Distributed Lookup Data Source**. The Distributed Lookup Data Source page opens.
3. Click **Edit**. The Distributed Lookup Data Source Selection wizard opens.
4. Complete the steps of the wizard.

   You must select a connection to the lookup data source and specify a DB alias and schema for the lookup data. You can also create or edit a connection.
5. Click **OK**.

Defining a z/OS lookup data source

Use the Optim preferences to define a lookup data source for the z/OS platform.

A database connection is required. You can add the connection in the Data Source Explorer view or create a connection when you define the lookup data source.

You will need to identify a schema for the lookup data.

To define a lookup data source for the z/OS platform:

1. Click **Window** > **Preferences**. The Preferences window opens.
2. In the navigation tree, expand the **Optim** node and then click **z/OS Lookup Data Source**. The z/OS Lookup Data Source page opens.
3. Click **Edit**. The z/OS Lookup Data Source Selection wizard opens.
4. Complete the steps of the wizard.

   You must select a connection to the lookup data source and specify a schema for the lookup data. You can also create or edit a connection.
5. Click **OK**.

Defining an executor lookup data source

Use the Optim preferences to define a lookup data source for the executor platform.

A database connection that includes a schema named EXTENDED_LOOKUP is required.

To use the identity privacy lookup policies, the database must include the tables from the EXTENDED_LOOKUP schema provided with the sample data.

The default connection for the executor lookup data source is the Optim Replacement Data connection profile, which is based on the EXTENDED_LOOKUP schema in a local management server installation.

You can add the connection in the Data Source Explorer view or create a connection when you define the lookup data source.

To define a lookup data source for the executor platform:

1. Click **Window** > **Preferences**. The Preferences window opens.
2. In the navigation tree, expand the **Optim** node and then click **Executor Lookup Data Source**. The Executor Lookup Data Source page opens.
3. Click **Edit**. The Executor Lookup Data Source Selection window opens.
4. Select a connection to the lookup data source that contains a schema named EXTENDED_LOOKUP. You can also create or edit a connection.
5. Click **OK**.

Chapter 3. Managing data models

To define an Optim interoperability model or process a data management service request, you must use a logical data model to define the source or target data. The logical data model must include metadata that identifies a database connection.

If a logical model does not contain connection information, you can provide the information when you define a data management service or convert the model to an Optim logical data model.

An Optim logical data model is a logical data model that includes a data access plan. A data access plan includes policies for selecting and masking data.

You can create an Optim logical data model by doing one of the following actions:

- transforming a physical data model that was created by reverse engineering from a data source connection
- transforming an Optim Database Relationship Analyzer physical data model
- converting a schema in a logical data model to an Optim logical data model

Physical data models based on reverse engineering

Use a physical model that is created by reverse engineering from a database to create an Optim logical data model. The database must have a connection profile defined in the Data Source Explorer.

Defining a physical data model based on reverse engineering

You can use the New Physical Data Model wizard to define a physical data model based on reverse engineering from a database or DDL file.

The database must have a connection profile defined in the Data Source Explorer.

To define a physical data model based on reverse engineering:

1. Click **File** > **New** > **Physical Data Model** from the main file menu. The New Physical Data Model wizard opens.
2. On the Model File page, specify the database, version, and location of the new model file.
3. Select **Create from reverse engineering**.

   If you choose to reverse engineer from a database, you must provide connection information on the next pages of the wizard. If you choose to reverse engineer from a DDL file, you must provide the path to a DDL file in the next pages of the wizard.
4. Complete the steps of the wizard.

The physical data model is created and displayed in the **Data Models** folder.

Using Database Relationship Analyzer physical data models

You can create physical data models based on a group of related tables defined in an Optim Database Relationship Analyzer metadata database.

The Optim Database Relationship Analyzer group discovery process allows you to create a group that references related tables in a relational database schema. To use the group in a Database Relationship Analyzer physical data model, you must define an Optim Database Relationship Analyzer connection profile that specifies the Optim Database Relationship Analyzer metadata database and metadata schema that includes the group.

To define a Database Relationship Analyzer physical data model, select the Optim Database Relationship Analyzer connection profile that includes the group, select the group that will provide the tables for the model, and then select the connection to the database that contains the tables defined in the group.

Setting up Optim Database Relationship Analyzer

To use Optim Database Relationship Analyzer in Optim Designer, do the following steps:

1. Install the Optim Database Relationship Analyzer server and perform the group discovery process for relational data you will use in Optim Designer.
2. Use the Data Source Explorer to define a data source connection to the Optim Database Relationship Analyzer metadata database.
3. Use the Optim preferences to define a connection profile based on the Optim Database Relationship Analyzer data source connection and a metadata schema. You can define a connection profile only for a metadata database created with Optim Database Relationship Analyzer version 1.1.1 or later.

Defining an Optim Database Relationship Analyzer connection profile

You can use a data source connection created in the Data Source Explorer to define an Optim Database Relationship Analyzer connection profile.

Before you can define a data source in an Optim Database Relationship Analyzer connection profile, the Optim Designer machine must be connected to the Optim Database Relationship Analyzer metadata database.

You can define a connection profile only for a metadata database created with Optim Database Relationship Analyzer version 1.1.1 or later.

To define a data source connection as an Optim Database Relationship Analyzer connection profile:

1. From the Optim Designer menu, click **Window** > **Preferences** to open the Preferences window.
2. In the Preferences window options list, expand the **Optim** node and select **Database Relationship Analyzer**. The Database Relationship Analyzer Preferences page opens.
3. On the Database Relationship Analyzer Preferences page, click **Add**. The Add a Database Relationship Analyzer Connection Profile window opens.
4. Select a listed data source connection. After Optim Designer retrieves the schemas from the data source, the **Choose a schema for the selected profile** list is available.
5. Select an Optim Database Relationship Analyzer metadata schema.
6. Click **Add** to return to the Database Relationship Analyzer preferences page. The page will list the connection and metadata schema selected in the Add a Database Relationship Analyzer Connection Profile window.
7. Click **Apply** or **OK** to save the Optim Database Relationship Analyzer connection profile.

Creating a Database Relationship Analyzer physical data model

You can use the New Database Relationship Analyzer Physical Data Model wizard to create a physical data model based on a group of related tables defined in an Optim Database Relationship Analyzer metadata database.

The Optim Database Relationship Analyzer metadata database must be have a connection profile defined in the Database Relationship Analyzer preferences.

The source database that contains the tables selected in the Database Relationship Analyzer group must have a connection profile defined in the Data Source Explorer.

To create a Database Relationship Analyzer physical data model:

1. In the Data Project Explorer, right-click the **Data Models** folder and click **New** > **Database Relationship Analyzer Physical Model**. The New Database Relationship Analyzer Physical Data Model wizard opens.
2. Complete the steps of the wizard.

   You must select a Database Relationship Analyzer connection profile and a Database Relationship Analyzer group that will provide relationship mapping for the model. You must also select and connect to the source database for the model. The source database must contain all tables in the Database Relationship Analyzer group selected for the model.

The physical data model is created and displayed in the **Data Models** folder.

Transforming a physical data model to an Optim logical data model

You can use the Transform to Optim Logical Data Model wizard to create a new Optim logical data model. You can create an Optim logical model based on an entire physical data model as well as a selected schema or entity in a physical data model.

If you create an Optim logical data model from a physical data model that does not include database connection information, you must provide connection information when you create the Optim logical data model.

To transform a physical data model to an Optim logical data model:

1. Expand the **Data Models** folder, right-click the physical data model, or either a schema or entity in the model, and click **Transform to Optim Logical Data Model**. The Transform to Optim Logical Data Model wizard opens.
2. On the first page of the wizard, select **Create new model**.
3. Complete the steps of the wizard.

   If this is the first Optim logical data model associated with the database connection, you must enter a name for a new Optim data source.

   If a native data source connection is available for the database, you can define or edit a native data source connection for the database.

The Optim logical data model is created and displayed in the **Data Models** folder.

Converting a schema in a logical data model to an Optim logical data model

You can use the Transform to Optim Logical Data Model wizard to convert a schema in a logical data model to an Optim logical data model.

To convert a schema in a logical data model to an Optim logical data model:

1. Expand the Data Models folder, expand a logical data model, expand the package, right-click a schema name, and click **Transform to Optim Logical Data Model**. The Transform to Optim Logical Data Model wizard opens.
2. Complete the steps of the wizard.

   You must select a database connection and match entities in the database to entities in the schema.

The Optim logical data model is created and displayed in the **Data Models** folder.

Using data access plans

A data access plan contains policies that determine which data to process or transform from a source Optim logical data model in a data management service.

Data access plans can include the following policies:

- Selection policies, which determines the entities and attributes to use in a data management service.
- Data privacy policies, which define how to mask data in a data management service.

Creating a data access plan

You can use the Data Access Plan wizard to add a data access plan to an Optim logical data model.

To create a data access plan:

1. Expand a logical data model package in the Data Project Explorer.
2. Right-click the **Data Access Plans** folder and click **New** > **Data Access Plan**. The Data Access Plan wizard opens.
3. Complete the steps of the wizard.
   - You must enter a name for the data access plan.
   - You must select data to include in the data access plan. The selected data will be added to the selection policy for the plan.

Editing a data access plan

You can use the data access plan editor to edit policies in a data access plan.

To edit a data access plan:

1. From the Data Project Explorer, expand a **Data Access Plans** folder in a logical data model package.
2. Right-click a data access plan and click **Open**. The data access plan editor opens.
3. Select the type of policy to edit at the top of the editor.

Working with data sources in a data access plan

You can use the data access plan editor to manage data sources in a data access plan.

A data access plan includes the following data sources:

- an Optim data source associated with the Optim logical data model
- the data sources associated with data privacy lookup policies

Use the data sources editor to refresh JDBC connection information and edit native data source connection information.

Working with a selection policy

A selection policy specifies the entities and attributes to use in an Optim interoperability model or data management service. A selection policy is defined when you create a data access plan.

Use the selection policy editor to edit a selection policy. The selection policy editor is available in the data access plan editor.

Use selection criteria to filter rows from an entity based on the criteria specified on the attributes. Selection criteria use an SQL select statement to select data from rows in an entity. You can apply selection criteria to an attribute or create an SQL where clause to apply criteria to an entity.

You can also determine which relationships to include in the policy, allowing you to determine the entities that are traversed and the data that is selected.

A selection policy includes the following entity types:

start entity
A start entity is the entity from which data is selected first during processing. Data from related entities is selected based on relationships with the start entity. When you select a start entity, all related entities are added to the policy.

related entity
A related entity is an entity from which data is selected based on a relationship with the start entity.

reference entity
A reference entity is an entity from which all attributes are selected during processing, regardless of any relationship to a start entity.

Defining related and reference entities in a selection policy

You can use the entities list to define related and reference entities in a selection policy.

If an entity is not related to the start entity, the entity can be a reference entity only.

To define related and reference entities in a selection policy:

1. From the data access plan editor, click **Selection**. The selection policy editor opens.
2. Click the **Entities** tab.
3. In the **Schema/Entities** column, select the check box next to each entity you want to change, and click **Change to Related** or **Change to Reference**.
4. Click **File** > **Save**.

Adding an entity to a selection policy

You can use the Add Entity wizard to add an entity to a selection policy.

To add an entity to a selection policy:

1. From the data access plan editor, click **Selection**. The selection policy editor opens.
2. Click the **Entities** tab.
3. From the selection policy editor, click **Add**. The Add Entity wizard opens.
4. Expand the associated packages and select an entity.
5. Click **Finish**.
6. Click **File** > **Save**.

The entities list will display the new entity under the name of the source logical data model.

Changing the entity selection in a selection policy

You can use the Change Entity Selection wizard to change the start entity or entity selection in a selection policy.

If you change the entity selection, selection criteria and privacy policies will be preserved for entities in the data access plan that are also included in the new entity selection.

To change the entity selection in a selection policy:

1. From the data access plan editor, click **Selection**. The selection policy editor opens.
2. Click the **Entities** tab.
3. Click **Change Entity Selection**. The Change Entity Selection wizard opens.

4. Complete the steps of the wizard.
5. Click **Finish**.
6. Click **File** > **Save**.

The **Entities** list in the selection policy editor will display the selected entities.

Removing an entity from a selection policy

You can use the entities list in the selection policy editor to remove an entity from a selection policy.

To remove an entity from a selection policy:

1. From the data access plan editor, click **Selection**. The selection policy editor opens.
2. Click the **Entities** tab.
3. Select the check box next to each entity you want to remove from the entities list.
4. Click **Remove**.
5. Click **File** > **Save**.

Defining selection criteria for an attribute

You can use the attributes list in the selection policy editor to define selection criteria for an attribute in an entity.

Selection criteria allow you to pinpoint the data you want to process. You can select data according to values in one or more attributes. Selection criteria must conform to SQL syntax and include relational or logical operators.

To define selection criteria for an attribute:

1. From the data access plan editor, click **Selection**. The selection policy editor opens.
2. Click the **Entity Specification** heading. The Entity Specification editor opens.
3. From the **Entity Name** list, select the entity that contains the attributes to which you want to add selection criteria.
4. Select the **Combine all criteria with** iterator that determines how the criteria is applied.
   a. Select **AND** if a row must match selection criteria for all attributes.
   b. Select **OR** if a row must match selection criteria for one attribute.
5. In the **Selection Criteria** column of the attributes list, click the browse button. The Selection Criteria window opens.
6. Enter SQL syntax in the editor area. For convenience, you can select the **Operator Symbols** or **Logical Operators** to include.

   Click **Check Syntax** to identify SQL syntax errors.
7. Click **OK** to return to the Entity Specification editor. The selection criteria is displayed in the **Selection Criteria for** area.
8. Select **View Selection Criteria SQL Summary** to view the attribute selection criteria in the Select statement for the entity.
9. Click **File** > **Save**.

Defining selection criteria for an entity

You can use the Entity Specification editor to define selection criteria for an entity in a selection policy.

Selection criteria allow you to pinpoint the data you want to process. You can select data according to values in one or more attributes. Selection criteria must conform to SQL syntax and include relational or logical operators.

To define selection criteria for an entity:

1. From the data access plan editor, click **Selection**. The selection policy editor opens.
2. Click the **Entity Specification** heading. The Entity Specification editor opens.
3. From the **Entity Name** list, select the entity to which you want to add selection criteria.
4. Click **Add/Edit Entity Selection Criteria**. The Entity Selection Criteria window opens.
5. Enter SQL syntax in the editor area. For convenience, you can select the **Attributes**, **Operator Symbols**, or **Logical Operators** to include.

   Click **Check Syntax** to identify SQL syntax errors.
6. Click **OK** to return to the Entity Specification editor. The selection criteria is displayed in the **Selection Criteria for** area.
7. Select **View Selection Criteria SQL Summary** to view the attribute selection criteria in the Select statement for the entity.
8. Click **File** > **Save**.

Managing relationships in a selection policy

You can use the relationships editor to select the relationships to exclude from the selection policy.

You can selectively ignore relationships between entities in the selection policy. Since the entity parent-child hierarchy can include multiple levels, an ignored relationship may result in orphaned relationships as well as entities that will not be traversed. The **Entities** tab will indicate if an entity will not be traversed because of an ignored relationship.

To manage relationships in a selection policy:

1. From the data access plan editor, click **Selection**. The selection policy editor opens.
2. Click the **Relationships** tab. The relationships editor opens.
3. In the **Available Relationships** area, use the **Ignore** column to select relationships to exclude from the selection policy. Any orphan relationships that result from an ignored relationship will display in the **Orphan Relationships** area.
4. Click **File** > **Save**.

Chapter 4. Designing data management services

Use an Optim data management service to transform data, copy data between schemas, or run Optim and Optim for z/OS requests.

You can mask data by applying a data privacy policy to an entity processed by a service. There are two types of services: executor services (for the executor platform) and Optim interoperability services (for the Optim and Optim for z/OS platforms). You can publish services to the Optim Manager environment, where you can run or schedule services. You can also use Optim Manager in embedded mode to test services.

Working with executor services

Executor services extract data described in a source Optim logical data model and insert or update data in a target Optim logical data model. An executor service requires an installation of Optim executor.

You can use an executor service to mask data based on privacy policies applied to an entity in an Optim logical data model. You can also mask data by defining a JavaScript policy in a service request. You can use Optim manager to test services from Optim Designer and use the Optim manager environment to run and schedule services.

Creating an executor service

Use the New Service wizard to create an executor service that uses a logical data model to provide the source data.

There are two types of executor services:

Copy service
A copy service copies data from one data model to another and can use filter criteria to select data.

Data transformation service
A data transformation service masks the data in the source data model.

Creating a copy service

You can use the New Service wizard to create a copy service.

A source and target Optim logical data model are required.

The source model must include a data access plan. You can use the New Service wizard to define a plan for the model.

To create a copy service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **Executor Service**. The New Service wizard opens.
2. Complete the steps of the wizard.

   You must select a source Optim logical data model and a data access plan in the model.

   Use the Target Model Options page to determine how to select a target data model for the service as well as the type of operation to perform on the target data model.

   The following target model selection options are available:

   Select a target model and perform auto map
   Use this option to select a target Optim logical data model and allow Optim Designer to

automap the source and target model. The automap function maps entities and attributes in the source model to matching entities and attributes in the target model. For the automap function to work, the source and target models should have similar schemas.

Open the service plan editor and manually complete the target model selection and map the source to the target

Use this option to open the service plan editor, in which you will select a target model and map the source model to the target model. To select a target model from the service plan editor, click **Add Target Model**.

The following target operations are available:

Insert Inserts new rows into the destination entities. If the primary key of a row in the source data does not match the primary key of a row in the destination entity, the row is inserted. If the primary key of a row in the source data matches the primary key of a row in the destination entity, the operation fails.

Update

Updates existing rows in the destination entities. If the primary key of a row in the source data matches the primary key of a row in the destination entity, the row is updated. If the primary key of a row in the source data does not match the primary key of a row in the destination entity, the operation fails.

Creating a data transformation service

You can use the New Service wizard to create a data transformation service.

A source Optim logical data model is required.

The source model must include a data access plan. You can use the New Service wizard to define a plan for the model.

To create a data transformation service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **Executor Service**. The New Service wizard opens.
2. Complete the steps of the wizard.

   You must select a source Optim logical data model and a data access plan in the model.

Editing an executor service

Use the service editor to edit an executor service.

Use the **Data Sources** to view and refresh JDBC connections and to view and edit native data source connections.

Use the **Service Plan** to work with the associated data access plan, target policies, and source to target map.

Data access plan

The data access plan determines which data to process or transform from the source Optim logical data model. You can open the associated data access plan from the service plan editor.

Source to Target Map

The source to target map determines the mapping between source and target attributes. You can use the map to edit mappings and select a new target Optim logical data model.

Target Policies

The following target policies are available:

- An update policy that determines whether source entities are inserted or updated in the target model. You can edit the update policy.

- Disable constraints policies that allow you to enable and disable constraints such as primary and foreign keys defined in an entity used in a service.
- JavaScript policies that allow you to use JavaScript expressions to define a data transformation for an attribute.
- Service diagnostic policies that allow you to set options for messages generated by a service.

Using the service editor

You can use the service editor to edit data source or service plan information for an executor service.

To edit an executor service:

1. From the Data Project Explorer, expand the **Services** folder and then double-click the service you want to edit. The service plan editor opens.
2. Select the item you want to edit: **Data Sources** or the **Service Plan**.
3. Click **File** > **Save** to save your changes.

Working with source to target mapping:

Use source to target mapping to determine the mapping between source and target attributes. You can also add or remove entities, select a new target Optim logical data model, and restore auto mapping.

Adding an entity to a source to target mapping:

Use the Add Map Entity wizard to add an entity to a source to target mapping.

To add an entity to a source to target mapping:

1. From the **Policies** list in the service plan editor, select **Source to Target Map**. The **Source to Target Mapping** editor opens.
2. Click **Add Map**. The Add Map Entity wizard opens.
3. Complete the steps of the wizard.

   You must select a source entity and a target entity.

Changing a target Optim logical data model:

You can use the Target Model Selection wizard to select a new target Optim logical data model for a service request.

Any update policies applied to the previous target Optim logical data model will be removed.

To change a target Optim logical data model:

1. From the **Policies** list in the service plan editor, select **Source to Target Map**. The **Source to Target Mapping** editor opens.
2. Click **Browse**. The Target Model Selection window opens.
3. Select an Optim logical data model.
4. Click **OK**.

Removing an entity from a source to target mapping:

Use the Remove Mapped Entities window to remove an entity from a source to target mapping.

To remove an entity from a source to target mapping:

1. From the **Policies** list in the service plan editor, select **Source to Target Map**. The **Source to Target Mapping** editor opens.
2. Click **Remove**. The Remove Mapped Entities window opens.

3. Select an entity.
4. Click **OK**.

Restoring auto mappings:

You can restore the default mapping to a source to target mapping. Any prior edits to the mapping will be overwritten.

To restore auto mappings:
1. From the **Policies** list in the service plan editor, select **Source to Target Map**. The **Source to Target Mapping** editor opens.
2. Click **Restore Auto Mapping**. The Restore Auto Mappings window opens, indicating the auto mapping will be based on the entities and attributes in the source and target Optim logical data models.
3. Click **OK**.

Managing Constraints:

You can use a disable constraints policy to enable and disable constraints such as primary and foreign keys defined in an entity used in a service.

Creating a disable constraints policy:

You can use the Add Policy wizard to create a disable constraints policy for a service plan.

To create a disable constraints policy:
1. From the Data Project Explorer, expand the **Services** folder and then open the service request that contains the service plan to which you will add the policy.
2. Double-click the **Service Plan** node. The service plan editor opens.
3. Click **Add Policy**. The Add Policy wizard opens.
4. Complete the steps of the wizard.

Enabling or disabling all constraints in an entity:

You can use the Disable Constraints Policy editor to enable or disable all constraints in an entity.

To enable or disable all constraints in an entity:
1. From the Data Project Explorer, expand the **Services** folder and then open the service request that contains the service plan with the policy.
2. Double-click the **Service Plan** node. The service plan editor opens.
3. In the **Target Policies** list, select the disable constraints policy. The Disable Constraints Policy editor opens.
4. Select one and more entities and click **Enable All Constraints** or **Disable All Constraints**. The **Disabled Constraints** column displays the disabled constraints for each entity.

Enabling or disabling selected constraints in an entity:

You can use the Disable Constraints Policy editor to enable or disable selected constraints in an entity.

To enable or disable selected constraints in an entity:
1. From the Data Project Explorer, expand the **Services** folder and then open the service request that contains the service plan to which you will add the policy.
2. Right-click the **Service Plan** node and click **Open**. The service plan editor opens.

3. In the **Target Policies** list, select the disable constraints policy. The Disable Constraints Policy editor opens.
4. In the **Disabled Constraints** column for the entity, click **...**. The Remove Mapped Entities dialog opens.
5. Select the constraints you want to disable or clear the constraints you want to enable. Click **OK**. The **Disabled Constraints** column displays the disabled constraints for the entity.

Working with a service diagnostics:

You can use a service diagnostics policy to set options for messages generated by a service.

Log messages generated by a service are stored in the .log file in the \.metadata directory in the workspace of the proxy installation. You can select the following log levels for a service request, listed in ascending order according to message severity.

ALL Log all messages.

FINEST
Highly detailed messages.

FINER
Fairly detailed messages.

FINE Detailed messages.

CONFIG
Static configuration messages, useful for debugging.

INFO Informational messages for end users and administrators. Default.

WARNING
Messages describing potential problems.

SEVERE
Messages indicating a serious failure.

OFF Turn logging off.

Creating a service diagnostics policy:

You can use the Add Policy wizard to create a service diagnostics policy for a service plan.

To create a service diagnostics policy:
1. From the Data Project Explorer, expand the **Services** folder and then open the service request that contains the service plan to which you will add the policy.
2. Double-click the **Service Plan** node. The service plan editor opens.
3. Click **Add Policy**. The Add Policy wizard opens.
4. Complete the steps of the wizard.
   Select log level and service notification options.

Editing an update policy:

Use the **Update Policy** editor to edit an update policy, which determines whether source entities are inserted or updated in the target model.

To edit an update policy:
1. From the **Policies** list in the service plan editor, select the update policy. The **Update Policy** editor opens.

2. Select the check box for each target entity you want to edit.
   To select all entities, click **Select All**.
   To clear all selected entities, click **Deselect All**.
3. Change the service action for the selected entities by clicking **Insert** or **Update**.
4. Enter the **Commit Frequency**, which determines the number of rows to process before committing changes to the database.

Testing executor services

You can use Optim Designer to test executor services before they are run in production.

To run a service from Optim Designer, the following are required:

- Optim executor must be installed on the Optim Designer machine
- an Optim license must be defined to Optim Designer

By default, Optim logical data models use a JDBC connection. For faster processing, select a native data source connection when you run a service.

Configuring Optim executor

The executor provides the framework needed by an executor service to communicate with a database or any other type of resource needed by the service. You can use the Optim preferences to configure Optim executor by entering the Optim executor installation path.

To configure Optim executor:

1. Click **Window** > **Preferences**. The Preferences window opens.
2. In the navigation tree, expand the **Optim** node and then click **Optim Executor**. The Optim Executor editor opens.
3. In the **Executor location** field, enter the path to the Optim Executor `eclipse.exe` file or click **Browse** to select the path.
4. Click **OK**.

Testing an executor service

You can run an executor service from Optim Designer. For example, if you want to test the service before you publish the service, you can test the service from Optim Designer.

To run a service from Optim Designer, Optim Executor must be installed on the Designer machine and an Optim license must be defined to the designer. You must also verify that the location of Optim Executor is set correctly in Optim Designer under **Window** > **Preferences** > **Optim** > **Optim Executor**.

To execute an executor service:

1. Open the **Services** folder.
2. Right-click an executor service and click **Execute Optim Service**. Optim Manager opens and the Run Service wizard is displayed.
3. Click **Run**. You can click **Service Monitoring** to monitor the progress of the service.

Managing Optim licenses

To run an executor service from Optim Designer, you must define the location of an Optim license or generate a 30-day trial license.

Defining an Optim license location:

You can use the Optim preferences to configure Optim Designer with an Optim license located on a management server.

To define the location of an Optim license:

1. Click **Window** > **Preferences**. The Preferences window opens.
2. In the navigation tree, expand the **Optim** node and then click **License**.
3. In the **License URL** field, enter the URL of a management server configured with a license.
   The URL uses the following format: *http://hostname:port/server/license*. For example, to enter the URL of a management server located on the local machine enter http://localhost:8080/server/license.
4. Click **Validate** to validate the license on the management server.
5. Click **OK**.

Generating a trial license:

You can use the Optim preferences to configure Optim Designer with a 30-day trial Optim license.

To generate a 30-day trial license:

1. Click **Window** > **Preferences**. The Preferences window opens.
2. In the navigation tree, expand the **Optim** node and then click **License**. The License page opens.
3. Click **Generate 30-Day Trial License**.
4. Click **OK**.

Working with Optim interoperability services

You can use Optim interoperability services to process requests in Optim and Optim for z/OS.

An Optim interoperability service is based on an Optim or Optim for z/OS request that you can run from the Optim Manager environment. You can also test the service from Optim Designer.

Working with Optim interoperability services on Linux, UNIX, and Windows

You can define Optim requests for Linux, UNIX, and Windows in an Optim interoperability service. You can also work with an Optim directory.

To test an Optim interoperability service from Optim Designer:

- the Optim Designer machine must include an installation of Optim
- you must use the Optim Distributed option in the Optim preferences to define the location of the Optim pr0cmnd utility in the Optim installation

All Optim requests run by the pr0cmnd utility will be associated with the default Optim directory associated with the Optim installation.

You can also define a connection to an Optim directory and include that directory in an Optim directory project. You can use the project to import and export Optim requests.

Creating Optim interoperability services

Use Optim interoperability services to process Optim requests.

Creating a distributed archive service:

You can use the New Archive Service wizard to create a distributed archive service.

An archive service copies a set of related rows from one or more tables and stores this data in an archive file. An archive service request defines the parameters for archiving and (if desired) deleting data from

source tables, and saving that data to an archive file. An archive service request references an access definition to define the data to archive and the parameters needed to run the archive process.

A archive service requires an Optim logical data model to provide the source data.

To create an archive service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **Distributed Service** > **Distributed Archive**. The New Archive Service wizard opens.
2. Complete the steps of the wizard.

   You must enter a name for the service and select an Optim data source, Optim logical data model, and data access plan.

   You must also provide a name for the archive file, specify options for the archive process, and select objects to archive.

Creating a distributed convert service:

You can use the New Convert Service wizard to create a distributed convert service.

A convert service transforms data in an extract file. You can convert data to assure data privacy or to systematically transform data to meet your application testing requirements. You can import converted data into a spreadsheet program, insert it into a testing database, or restore it to a reporting database.

A convert service requires an Optim logical data model to provide the source data.

To create a convert service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **Distributed Service** > **Distributed Convert**. The New Convert Service wizard opens.
2. Complete the steps of the wizard.

   You must enter a name for the service and select an Optim data source, Optim logical data model, and data access plan.

   You must enter the name of a table map to use with the request. You must also then enter a name for the extract or archive file with the source data, enter a destination file name, enter a control file name, and specify options for the convert service.

   If you choose a comma separated values file as the destination file, you must select formatting options.

Creating a distributed delete service:

You can use the New Delete Service wizard to create a distributed delete service.

A delete service removes sets of related data from a database after an extract or archive process. The delete process is initiated by a delete request, which identifies an extract or archive file as the source file containing the data you want to delete, and specifies the parameters for the delete process.

To create a delete service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **Distributed Service** > **Distributed Delete**. The New Delete Service wizard opens.
2. Complete the steps of the wizard.

   You must specify the source archive or extract file name, the control file name, and options for the delete process.

Creating a distributed extract service:

You can use the New Extract Service wizard to create a distributed extract service.

An extract service copies a set of related rows from one or more tables and stores this data in an extract file. The extract service always includes the definitions for tables and columns. You can also choose to extract object definitions, including primary keys, relationships, and indexes. An extract service request specifies an access definition to define the data to extract and the parameters needed to run the extract process.

A extract service requires an Optim logical data model to provide the source data.

To create an extract service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **Distributed Service** > **Distributed Extract**. The New Extract Service wizard opens.
2. Complete the steps of the wizard.

   You must enter a name for the service and select an Optim data source, Optim logical data model, and data access plan.

   You must also provide a name for the extract file, specify options for the extract process, and select objects to extract.

Creating a distributed insert service:

You can use the New Insert Service wizard to create a distributed insert service.

An insert service copies data from a source file into specified destination tables. An insert service request specifies a source file containing the data you want to insert or update and the parameters needed to run the process.

An insert service requires an Optim logical data model to provide a map to the source data.

To create an insert service:

1. From the Data Project Explorer, right-click the **Services** folder, and click **New** > **Distributed Service** > **Distributed Insert**. The New Insert Service wizard opens.
2. Complete the steps of the wizard.

   You must enter a name for the service and select the Optim data source, logical data model, and data access plan used to create the source file.

   You must also provide names for the source and control files, and specify options for the insert process.

Creating a distributed load service:

You can use the New Load Service wizard to create a distributed load service.

A load service transforms the contents of a source file (either an extract or archive file) into the load utility format for a supported database. A load service request specifies the source file containing the data to load and other process parameters.

A load service requires an Optim logical data model to provide a map to the source data.

To create a load service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **Distributed Service** > **Distributed Load**. The New Load Service wizard opens.
2. Complete the steps of the wizard.

You must enter a name for the service and select an Optim data source, Optim logical data model, and data access plan.

You must also provide names for the source and control files, and enter load utility properties.

Creating a distributed restore service:

You can use the New Restore Service wizard to create a distributed restore service.

A restore service selects data from one or more archive files and restores the data to the original or a different database. A restore service request specifies the archive files and defines the insert or load request used to restore the archived data.

A restore service requires an Optim logical data model to provide the source data.

To create a restore service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **Distributed Service** > **Distributed Restore**. The New Restore Service wizard opens.
2. Complete the steps of the wizard.

   You must enter a name for the service and select the Optim data source, logical data model, and data access plan used to create the source file.

   You must also enter names for the archive and control files, and specify options for the insert process. You can also specify selective restore properties.

Defining a pr0cmnd and pr0cnfg location

You can use the Optim Distributed option in the Optim preferences to define the location of the pr0cmnd and pr0cnfg utilities in an Optim installation.

You must have Optim installed on the Optim Designer machine.

To define the pr0cmnd and pr0cnfg location:

1. Click **Window** > **Preferences**. The Preferences window opens.
2. In the navigation tree, expand the **Optim** node and then click **Optim Distributed**. The Optim Distributed editor opens.
3. In the **Command line directory** field, enter the path to the pr0cmnd.exe and pr0cnfg.exe files or click **Browse** to select the path. The default location is C:\Program Files\IBM Optim\RT\BIN\.
4. Click **OK**.

Creating an Optim directory project

You can use the New Optim Directory Project wizard to create an Optim directory project, which allows you to import or export Optim requests.

An Optim directory project requires a connection profile to the database that contains the directory. You can define a connection profile by using the Data Source Explorer or the New Optim Directory Project wizard.

To create an Optim directory project:

1. Click **File** > **New** > **Optim Directory Project**. The New Optim Directory Project dialog opens.
2. Complete the steps of the wizard.

   To complete the wizard, you must enter a name for the directory project and select connection to the database that contains the directory. You must also select the schema that contains the Optim directory tables.

Defining a DB alias

Use the New DB Alias wizard to define a DB alias. A DB alias is a set of specifications that allows Optim to identify, locate, and access a particular database. The DB alias also qualifies the names of objects referenced, defined, or accessed using Optim.

Before you define a DB alias, an Optim directory project that will contain the DB alias must be defined in the Data Project Explorer. You must also use the Optim Distributed option in the Optim preferences to define the location of the Optim pr0cnfg utility in the Optim installation.

You must also use the Data Source Explorer to define a connection profile for the database connection. You can only define a DB alias for databases supported by Optim.

Defining a DB alias from a connection profile:

When you define a DB alias from a connection profile, the New DB Alias wizard is populated with properties from the connection profile.

To define a DB alias from a connection profile:

1. From the Data Source Explorer, expand the **Database Connections** folder.
2. Right-click the connection to the data source for the DB alias and click **Create Optim DB Alias**. The New DB Alias wizard opens.
3. Complete the steps of the wizard.
   - You must select an Optim directory project that will contain the DB alias.
   - You must enter information about the DBMS associated with the DB alias as well as connection information.

Defining a DB alias from the File menu:

To define a DB alias from the **File** menu:

1. Click **File** > **New** > **Other**. The New wizard opens.
2. Expand the **Optim** folder, select **DB Alias**, and click **Next**. The New DB Alias wizard opens.
3. Complete the steps of the wizard.
   - You must select an Optim directory project that will contain the DB alias.
   - You must enter information about the DBMS associated with the DB alias as well as connection information.

Defining a DB alias from an Optim directory project:

To define a DB alias from an Optim directory project:

1. From the Data Project Explorer, expand the Optim directory project that will contain the DB alias, expand the **Optim Directory** folder, right-click **DB Alias**, and click **New**. The New DB Alias wizard opens.
2. Complete the steps of the wizard.
   - You must enter information about the DBMS associated with the DB alias as well as connection information.

Defining an Optim server name

Use the Optim Distributed option in the Optim preferences to define an Optim server name. The definition allows you to select an Optim server name when you create an Optim interoperability model request.

To define an Optim server name:

1. Click **Window** > **Preferences**. The Preferences window opens.

2. In the navigation tree, expand the **Optim** node and then click **Optim Distributed**. The Optim Distributed editor opens.
3. In the **Optim Server** area, click **Add**. The Add Server window opens.
4. In the **Server** field, type an Optim server name. You can also enter a description.
5. Click **OK**. The server name is listed in the **Optim Server** list.
6. Click **Apply**.

To edit a server, select a server name and click **Edit**. In the Edit Optim Server window, edit server information, click **OK** to return to the Optim Distributed editor, and then click **Apply**.

To remove a server, select a server name, click **Remove**, and then click **Apply**.

Exporting Optim requests

You can export Optim requests included in Optim interoperability services to an Optim directory or an Optim export file.

Exporting requests in Optim interoperability services to an Optim export file:

Use the Export wizard to export requests from one or more Optim interoperability services to an Optim export file (OEF).

Before you can export requests, you must use the Optim Distributed option in the Optim preferences to define the location of the Optim pr0cmnd utility in the Optim installation.

To export definitions from one or more services to an OEF:

1. From the Data Project Explorer, expand the **Services** folder in a project.
2. Right-click a service and click **Export**. The Export wizard opens.
   You can also open the Export wizard by clicking **File** > **Export**.
3. Complete the steps of the wizard.
   On the Select page, expand the **Optim Interoperability Services** folder and select **Optim Export File** as the export destination.
   You must select the services that contain the requests to export and enter an OEF name.

Exporting requests from Optim interoperability services to an Optim directory:

Use the Export wizard to export requests from one or more Optim interoperability services to an Optim directory.

Before you can export requests, you must use the Optim Distributed option in the Optim preferences to define the location of the Optim pr0cmnd utility in the Optim installation. You must also create an Optim directory project.

To export definitions from one or more services to an Optim directory:

1. From the Data Project Explorer, expand the **Services** folder in a project.
2. Right-click a service and click **Export**. The Export wizard opens.
   You can also open the Export wizard by clicking **File** > **Export**.
3. Complete the steps of the wizard.
   On the Select page, expand the **Optim Interoperability Services** folder and select **Optim Directory** as the export destination.
   You must select the services that contain the requests to export and a target Optim directory project.

Exporting a request from an Optim directory to an Optim export file:

Use the Optim Export File Name window to export a request from an Optim directory to an Optim export file (OEF).

Before you can export requests, you must use the Optim Distributed option in the Optim preferences to define the location of the Optim pr0cmnd utility in the Optim installation. You must also create an Optim directory project.

To export a request from an Optim Directory to an OEF:

1. From the Data Project Explorer, expand the **Optim Directory** folder in a project.
2. Right-click a definition and click **Create Optim Export File**. The Optim Export File Name window opens.
3. Enter an OEF name and click **OK**.

Importing Optim requests

You can import Optim requests to Optim interoperability services or import requests to an Optim directory.

You can import Optim requests contained in an Optim export file (OEF).

Importing Optim requests to Optim interoperability services:

Use the Import wizard to import requests from an Optim export file (OEF) and transform them to Optim interoperability services.

Before you can import requests, you must use the Optim Distributed option in the Optim preferences to define the location of the Optim pr0cmnd utility in the Optim installation.

To import requests from an OEF and transform them to Optim interoperability services:

1. From the Data Project Explorer, expand the **Services** folder in a project.
2. Right-click a service and click **Import**. The Import wizard opens.
   You can also open the Import wizard by clicking **File** > **Import**.
3. On the Select page, open the **Optim Interoperability Services** folder and select **Optim Export File**.
4. Complete the steps of the wizard.
   You must select a project to contain the services and select the requests to transform.

Importing Optim requests to an Optim Directory:

Use the Import wizard to import requests from an Optim export file (OEF) to an Optim directory.

Before you can import definitions, you must use the Optim Distributed option in the Optim preferences to define the location of the Optim pr0cmnd utility in the Optim installation. You must also create an Optim directory project.

To import requests from an OEF to an Optim directory:

1. From the Data Project Explorer, expand the **Optim Directory** folder in a project.
2. Click **File** > **Import**. The Import wizard opens.
3. On the Select page, open the **Optim Directory** folder and select **Optim Export File**.
4. Complete the steps of the wizard.
   You must select a project to contain the services and select the requests to import.

Transforming an Optim request to an Optim interoperability service:

You can transform a request in an Optim Directory project to an Optim interoperability service.

Before you can import definitions, you must use the Optim Distributed option in the Optim preferences to define the location of the Optim pr0cmnd utility in the Optim installation. You must also define an Optim directory project.

To transform an Optim request to an Optim interoperability service:

1. From the Data Project Explorer, expand an Optim directory project to display the request to transform.
2. Right-click the request and click **Transform to Optim Service**. The Transform Request (type) to Optim Service window opens.
3. Enter an identifier and name for the service. You must also select a project for the service. Click **OK**.

Working with Optim interoperability services on z/OS

You can define Optim for z/OS requests in an Optim interoperability service.

Use the Add Host window in Optim preferences to configure a connection to an Optim for z/OS host machine, which is required to create and process z/OS services.

Creating Optim interoperability services for z/OS

Use Optim interoperability services to process Optim for z/OS requests.

Creating a z/OS archive service:

You can use the New Archive Service wizard to create a z/OS archive service.

An archive service copies a set of related rows from one or more tables and stores this data in an archive file. An archive service request defines the parameters for archiving and (if desired) deleting data from source tables, and saving that data to an archive file. An archive service request references an access definition to define the data to archive and the parameters needed to run the archive process.

A archive service requires an Optim logical data model to provide the source data.

To create an archive service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **z/OS Service** > **z/OS Archive**. The New Archive Service wizard opens.
2. Complete the steps of the wizard.

   You must enter a name for the service and select an Optim data source, Optim logical data model, and data access plan.

   You must also provide a name for the archive file, enter an access definition name, specify options for the archive process, and select objects to archive.

Creating a z/OS convert service:

You can use the New Convert Service wizard to create a z/OS convert service.

A convert service transforms data in an extract file. You can convert data to assure data privacy or to systematically transform data to meet your application testing requirements. You can import converted data into a spreadsheet program, insert it into a testing database, or restore it to a reporting database.

A convert service requires an Optim logical data model to provide the source data.

To create a convert service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **z/OS Service** > **z/OS Convert**. The New Convert Service wizard opens.
2. Complete the steps of the wizard.

You must enter a name for the service and select an Optim data source, Optim logical data model, and data access plan.

You must enter the name of a table map to use with the request. You must also then enter a name for the extract or archive file with the source data, enter a destination file name, enter a control file name, and specify options for the convert service.

If you choose a comma separated values file as the destination file, you must select formatting options.

Creating a z/OS delete service:

You can use the New Delete Service wizard to create a z/OS delete service.

A delete service removes sets of related data from a database after an extract or archive process. The delete process is initiated by a delete request, which identifies an extract or archive file as the source file containing the data you want to delete, and specifies the parameters for the delete process.

To create a delete service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **z/OS Service** > **z/OS Delete**. The New Delete Service wizard opens.
2. Complete the steps of the wizard.

   You must specify the source archive or extract file name, the control file name, and options for the delete process.

Creating a z/OS extract service:

You can use the New Extract Service wizard to create a z/OS extract service.

An extract service copies a set of related rows from one or more tables and stores this data in an extract file. The extract service always includes the definitions for tables and columns. You can also choose to extract object definitions, including primary keys, relationships, and indexes. An extract service request specifies an access definition to define the data to extract and the parameters needed to run the extract process.

A extract service requires an Optim logical data model to provide the source data.

To create an extract service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **z/OS Service** > **z/OS Extract**. The New Extract Service wizard opens.
2. Complete the steps of the wizard.

   You must enter a name for the service and select an Optim data source, Optim logical data model, and data access plan.

   You must also provide a name for the extract file, enter an access definition name, specify options for the extract process, and select objects to extract.

Creating a z/OS insert service:

You can use the New Insert Service wizard to create a z/OS insert service.

An insert service copies data from a source file into specified destination tables. An insert service request specifies a source file containing the data you want to insert or update and the parameters needed to run the process.

An insert service requires an Optim logical data model to provide a map to the source data.

To create an insert service:

1. From the Data Project Explorer, right-click the **Services** folder, and click **New** > **z/OS Service** > **z/OS Insert**. The New Insert Service wizard opens.
2. Complete the steps of the wizard.

   You must provide a name for the insert request and select the Optim data source, logical data model, and data access plan used to create the source file.

   You must also provide names for the source and control files, enter the table map name, and select options for the insert process.

Creating a z/OS load service:

You can use the New Load Service wizard to create a z/OS load service.

A load service transforms the contents of a source file (either an extract or archive file) into the load utility format for a supported database. A load service request specifies the source file containing the data to load and other process parameters.

A load service requires an Optim logical data model to provide a map to the source data.

To create a load service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **z/OS Service** > **z/OS Load**. The New Load Service wizard opens.
2. Complete the steps of the wizard.

   You must enter a name for the service and select an Optim data source, Optim logical data model, and data access plan.

   You must also provide names for the source and control files, and enter load utility properties. Load utility properties include the name of the loader parameter file, the dataset prefix for the loader, and field specification files. The loader parameter file must be a fixed block data set.

Creating a z/OS restore service:

You can use the New Restore Service wizard to create a z/OS restore service.

A restore service selects data from one or more archive files and restores the data to the original or a different database. A restore service request specifies the archive files and defines the insert or load request used to restore the archived data.

A restore service requires an Optim logical data model to provide the source data.

To create a restore service:

1. From the Data Project Explorer, right-click the **Services** folder and click **New** > **z/OS Service** > **z/OS Restore**. The New Restore Service wizard opens.
2. Complete the steps of the wizard.

   You must enter a name for the service and select the Optim data source, logical data model, and data access plan used to create the source file.

   You must also enter names for the archive and control files, enter a table map name, and specify options for the restore process. You can also specify selective restore properties.

Defining a z/OS host configuration

You can use the Optim preferences to define a connection to an Optim for z/OS host by entering connection and job information.

The configuration information is used to generate the batch JCL for Optim requests included in an Optim interoperability service.

To define an Optim for z/OS host configuration:

1. Click **Window** > **Preferences**.
2. In the navigation tree, expand the **Optim** node and then click **z/OS Host Configurations**. The z/OS Host Configurations editor opens.
3. Click **Add**. The Add Host window opens.
4. Complete the **Job Defaults** and **DB2 Defaults** information.
5. Click **OK**. The z/OS Host Configurations editor displays the host configuration.

Add Host window:

Use the Add Host window to configure a connection to an Optim for z/OS host by entering connection and job information.

The information entered in this window is used to generate the batch JCL for Optim requests included in an Optim interoperability service.

You can include the request definitions and parameters in either a request dataset or the JCL. Requests that contain multi-byte characters must use a request dataset.

Job Defaults

Host Name
The machine name or TCP/IP address of the Optim for z/OS host.

User Name
The ID of the user that will run the request on the host machine.

Password
The password for the user ID.

Job Name
The batch job name.

Accounting Information
The user account number.

Programmer's Name
The name of the programmer associated with the batch job.

Job Class
The job class for the batch job.

Message Class
The message class for the batch job.

Message Level
The message level for batch job output.

Notify The ID of the user to receive notification messages.

Site Options Library
The location of the library containing the site options. Contact your Optim or DB2 administrator for the location at your site.

Request Dataset

You can choose to include request definitions and parameters in a dataset. A request dataset is required for requests that contain multi-byte characters. The dataset must be variable blocked with a record length of 80 (RECFM=VB and LRECL=80).

Dataset Name
The name of the dataset for the request definitions and parameters.

Character Set
The character set for the dataset. Select **default** to use the default for the z/OS machine.

DB2 Defaults

Sub System
The current DB2 subsystem.

Plan Name
The DB2 plan name.

SQL ID
The current SQLID.

Step Libraries
The location of the step libraries. Contact your Optim or DB2 administrator for the location at your site.

Editing an Optim interoperability service

You can use the Properties view to edit a request in an Optim interoperability service.

To edit an Optim interoperability service:

1. From the Data Project Explorer, expand the **Services** folder and double-click the Optim interoperability service that contains the request you want to edit.
2. Select the request or definition you want to edit. The Properties view will display properties for the selected item.
3. To edit a request, select the **Request** tab. To edit a definition, select the **Definition** tab.
4. Edit the request or definition properties.
5. Click **File** > **Save** to save the changes.

Request definitions

Requests included in Optim interoperability services contain one or more definitions.

The following definitions are available, depending on the request.

Icon	Definition
	access definition
	archive request
	column
	column map
	convert request
	creator ID

Icon	Definition
	DB alias
	delete request
	extract request
	insert request
	load request
	primary key
	relationship
	restore request
	table
	table map
	variable

Editing a column map

You can use a column map editor to edit destination columns or to define an Optim or Optim for z/OS data privacy function for source columns.

To edit a column map:

1. From the Data Project Explorer, expand the **Services** folder, double-click the Optim interoperability service that contains the column map, and expand the column map.
2. Right-click the column mapping you want to edit and click **Properties**. The column map editor opens in the Properties view.
3. Select the **Definition** tab.

   Use the **Source column name** field to select a data privacy function for the source column. You can edit the function.

   Use the **Destination column name** field to edit the destination column name.
4. Click **File** > **Save** to save the changes.

Testing an Optim interoperability service

You can test an Optim interoperability service from Optim Designer. You can test the service before you publish the service to the Optim Manager environment.

Before you can test an Optim interoperability service for an Optim request on Linux, UNIX, or Windows, you must define an Optim pr0cmnd location in the Optim preferences.

Before you can test an Optim interoperability service for an Optim request on z/OS, you must define a z/OS host configuration in the Optim preferences.

To test an Optim interoperability service:

1. Open the **Services** folder.

2. Right-click an Optim interoperability service and click **Execute Optim Service**. Optim Manager opens. If the service includes an Optim request on Linux, UNIX, or Windows, the Run Service window is displayed. If the service includes an Optim request on z/OS, the Run Service wizard is displayed.
3. Review the service properties according to the request type:
   - For an Optim request on Linux, UNIX, or Windows, verify the import parameters and run parameters.
   - For an Optim request on z/OS, select a z/OS host configuration and verify the JCL. You can also specify a dataset to contain the request instead of using JCL.
4. Click **Run**. You can click **Service Monitoring** to monitor the progress of the service.

Working with the Optim registry

Use the Optim registry to store service requests that can be run from the Optim manager environment.

After defining a service request, you must publish the request to an Optim registry, where it will be available to the manager environment.

You can use an SSL connection with the registry.

Entering a default Optim registry location

You can use Optim Manager to enter a default Optim registry location. The default location is displayed in the Publish Service wizard available in Optim Manager.

To enter a default Optim registry location:

1. Open Optim Manager.

   You can open Optim Manager by publishing or executing a service. You can also open Optim Manager by entering the following URL in a web browser: `http://localhost:portnumber/console`, where *portnumber* is the port number assigned to Optim Manager. The default port number is 60000.

   If the browser displays a message about a page not found when you open Optim Manager, a port conflict may exist and you must change the Optim Manager port number.
2. Click **Preferences**. The Preferences window opens.
3. In the **Global Preferences** tab, enter the URL of the default Optim registry in the **Registry location** field.
4. Click **Save**.

Publishing a service

You can publish a service to a registry from Optim Designer. Publish a service when you want to make the service available to Optim Manager users.

To publish a service:

1. Open the **Services** folder.
2. Right-click a service and click **Publish Optim Service to Registry**. Optim Manager opens and the Publish Service window is displayed.
3. Verify the service information and registry location. To publish an Optim interoperability service for an Optim request on z/OS, you must also select a z/OS batch host.

   To publish the service to a different registry, change the registry location. To define a default registry, use the Optim Manager preferences.
4. Click **Validate**. Optim Manager displays the version number that is to be used to publish the service on the registry.
5. Click **OK** to finish.

Establishing a secure connection

You can use the Optim preferences to establish an SSL connection between Optim Designer and the management server that contains the Optim registry.

An SSL connection requires the following on the Optim Designer machine:

- the Optim Designer private key
- the management server public key

An SSL connection requires the following on the management server:

- the Optim Designer public key
- the management server private key

To establish a secure connection:

1. Click **Window** > **Preferences**. The Preferences window opens.
2. In the navigation tree, expand the **Optim** node and then click **SSL Connection**.
3. Enter the key store information for the Optim Designer private key and the trust store information for the management server public key.
4. Click **OK**.

Exporting a service to a file system

You can export a data management service from Optim Designer to a file system.

To export a service to a file system:

1. Open the **Services** folder.
2. Right-click a service and click **Export Optim Service to File System**. Optim Manager opens and the Export Service as File window is displayed.
3. Confirm that the information in the Export Service as File window is correct and click **OK**. For Optim interoperability services that include an Optim request on z/OS, you must also select a z/OS host.
4. Select the location to which you want to export the service and click **Save**.

Chapter 5. Using data privacy policies

Data privacy policies allow you to mask data in a data management service. There are three options to mask data with a privacy policy: lookup, rule-based, and JavaScript. The lookup option uses a lookup table to provide masked data. The rule-based option uses functions to generate masked data. The JavaScript option uses JavaScript to define a data transformation and is available for use with data management services only.

The lookup and rule-based options are applied to an entity in an Optim logical data model. When a policy that uses a lookup or rule-based option is applied to an entity, you must perform a data management service on the entity to transform data in the entity. Use a data access plan to apply a policy that uses a lookup or rule-based option. To create a data privacy policy in a data access plan, use the Add Policy wizard.

The JavaScript option is included in a JavaScript policy, which performs data transformations for a specific service beyond the lookup and rule-based options. A JavaScript policy applies to entities in the data management service in which the policy is defined and is executed when the service is run. Use a service plan to add a JavaScript policy to a service. The transformations defined in a JavaScript policy occur after Optim performs any lookup or rule-based transformations on the source data.

These policies offer the following features:

- Use lookup functions to replace values from selected source entities with values from corresponding lookup table columns
- Use rule-based functions to mask national ID numbers, credit card numbers, and e-mail addresses with valid and unique values
- Use rule-based functions to generate values for dates, characters, and numbers
- Apply a lookup or rule-based function based on a "switch" value
- Use JavaScript to define custom transformations in a data management service

Data privacy policies

Use the data privacy policies to mask data.

Date privacy policies

Use the date privacy policies to mask dates. The policies include the rule-based and JavaScript options.

Age policy

Use the age policy to age date values.

The policy can mask character, numeric, date, or timestamp data.

The following options are available:

- Age dates using an incremental time period or a specific year.
- Age dates based on rules used to manage dates that fall on holidays, weekends, etc.
- Age dates based on a specific date format.

Creating an age policy:

You can use the privacy policy editor to create an age policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create an age policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand the **Date Privacy Policies** and click **Age**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   - You can select parameters to use aging options specified in a process request or to not age values.
   - You can choose to age dates using an incremental time period or a specific year.
   - You can specify rules used to manage dates that fall on holidays, weekends, etc.
   - You can specify a format for the source and destination data.

Random date in range

The random date in range policy generates a randomly selected date within a specified date range. There are several formats available for the masked date.

JavaScript policy syntax

This policy is available with the function `DateMask.randomDateInRange()`.

```
DateMask.randomDateInRange(<arg:startDate>, <arg:endDate>, <arg:dateFormat>)
```

For example, to generate random dates from January 1, 1999 to January 1, 2009 in the MM-dd-yyyy format, use the following syntax:

```
DateMask.randomDateInRange('1999-01-01', '2009-01-01', 'MM-dd-yyyy')
```

Argument	Description
endDate	The end date of the date range in yyyy-MM-dd format.
startDate	The start date of the date range in yyyy-MM-dd format.
dateFormat	The format of the masked date. The default is yyyy-MM-dd. The following formats are supported: • dd-MM-yyyy • dd-MM-yyyy HH:mm:ss • MM-dd-yyyy • MM-dd-yyyy HH:mm:ss • MMM dd, yyyy • yyyy-MM-dd

Creating a random date in range policy:

You can use the privacy policy editor to create a random date in range policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a random date in range policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.

2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand the **Date Privacy Policies**, and click **Random date in range**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   You must supply a start and end date for the range. You can specify the format of the masked date.

Round date to month

The round date to month policy masks a date by rounding the date to the first day of the original month. For example, August 21 would round to August 1. The format of the generated date will match the input date.

JavaScript policy syntax

This policy is available with the function `DateMask.roundDateToMonth()`.

```
DateMask.roundDateToMonth(record.getItem(<arg:inputAttribute>), <arg:dateFormat>)
```

For example, to round a date to the first day of the month in MM-dd-yyyy format, enter the following syntax:

```
DateMask.roundDateToMonth(record.getItem('/DEMO/ORDERS/ORDER_DATE'), 'MM-dd-yyyy')
```

Argument	Description
inputAttribute	The source attribute containing the date to mask.
dateFormat	The format of the date. The default is yyyy-MM-dd. The following formats are supported: • dd-MM-yyyy • dd-MM-yyyy HH:mm:ss • MM-dd-yyyy • MM-dd-yyyy HH:mm:ss • MMM dd, yyyy • yyyy-MM-dd

Creating a round date to month policy:

You can use the privacy policy editor to create a round date to month policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a round date to month policy:
1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand the **Date Privacy Policies** and click **Round date to month**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.

7. Complete the steps of the wizard.
   You can specify the format of the masked date.

Round date to year

The round date to year policy masks a date by rounding the date to January 1 of the original year. For example, August 21, 2008 would round to January 1, 2008. The format of the generated date will match the input date.

JavaScript policy syntax

This policy is available with the function `DateMask.roundDateToYear()`.

```
DateMask.roundDateToYear(record.getItem(<arg:inputAttribute>), <arg:dateFormat>)
```

For example, to round a date to the first day of the year in MM-dd-yyyy format, enter the following syntax:

```
DateMask.roundDateToYear(record.getItem('/DEMO/ORDERS/ORDER_DATE'), 'MM-dd-yyyy')
```

Argument	Description
inputAttribute	The source attribute containing the date to mask.
dateFormat	The format of the masked date. The default is yyyy-MM-dd. The following formats are supported: • dd-MM-yyyy • dd-MM-yyyy HH:mm:ss • MM-dd-yyyy • MM-dd-yyyy HH:mm:ss • MMM dd, yyyy • yyyy-MM-dd

Creating a round date to year policy:

You can use the privacy policy editor to create a round date to year policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a round date to year policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Date Privacy Policies** and click **Round date to year**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   You can specify the format of the masked date.

Identity privacy policies

Use the identity privacy policies to mask personal information such as e-mail addresses, credit card numbers, and national ID numbers. The policies include rule-based and JavaScript options.

E-mail address policies

Use the e-mail address policies to mask e-mail addresses. Two policies are available: auto-generated e-mail name and formatted e-mail name.

Auto-generated e-mail name:

The auto-generated e-mail name policy generates an e-mail address with a user name based on a literal concatenated with a sequential number. The sequential numbers are suffixes that begin with 1 and are incremented by 1. The policy uses the domain name from an e-mail address in a specified source attribute.

Creating an auto-generated e-mail name policy:

You can use the privacy policy editor to create an auto-generated e-mail name policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create an auto-generated e-mail name policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Identity Privacy Policies** and **Email Address Policies**, and then click **Auto-generated e-mail name**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.

   You must provide a literal for the user name. You can convert the e-mail address to either upper or lowercase.

Formatted e-mail name:

The formatted e-mail name policy generates an e-mail address with a user name based on values obtained from one or two attributes. The policy uses the domain name from an e-mail address in a specified source attribute.

Creating a formatted e-mail name policy:

You can use the privacy policy editor to create a formatted e-mail name policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a formatted e-mail name policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Identity Privacy Policies** and **Email Address Policies**, and then click **Formatted e-mail name**.

5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   - You must select up to two source attributes to provide the user name.
   - You can choose to use only the first character from the attribute that provides the first part of a user name and include a separator between the two parts of a user name.
   - You can convert the e-mail address to either upper or lowercase.

Credit card policies

The credit card policies mask a credit card number (CCN) from the following issuers: American Express, Diners Club, Discover, JCB, MasterCard, and VISA. Each policy preserves the first 4 digits of the issuer identifier from the source CCN and masks the remaining 2 digits of the issuer identifier number and the account number based on the source CCN. Each policy also generates a check digit.

A CCN, as defined by ISO 7812, consists of a 6-digit issuer identifier followed by a variable length account number and a single check digit as the final number. The check digit verifies the accuracy of the CCN and is generated by passing the issuer identifier and account numbers through the Luhn algorithm. The maximum length of a CCN is 19 digits.

Credit card specific policies:

The credit card policies include a policy for each supported credit card issuer. The credit card specific policies mask only numbers that match the specified credit card issuer.

The following credit card specific policies are available:

- Mask American Express credit card numbers
- Mask Discover credit card numbers
- Mask Diners Club credit card numbers
- Mask JCB credit card numbers
- Mask MasterCard credit card numbers
- Mask VISA credit card numbers

Creating a credit card specific policy:

You can use the privacy policy editor to create a credit card specific policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a credit card specific policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Identity Privacy Policies** and **Credit Card**, and then click the policy for the credit card issuer.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   You can choose to mask the issuer number.

You can use the properties area in the data privacy editor to provide a seed number from which the masked number is generated.

Mask credit card numbers from all providers:

The mask credit card numbers from all providers policy will determine the issuer of the CCN and mask the number according to format of the issuer.

JavaScript policy syntax

This policy is available with the functions `CCNMask.randomCCN()` and `CCNMask.maskCCN()`.

To generate a random value not based on an input value, use the function: `CCNMask.randomCCN()`

To generate a value based on an input value, use the function:
`CCNMask.maskCCN(record.getItem('<arg.inputAttribute>'))`

For example, to generate a random value based on an input value, use the following syntax:

```
CCNMask.maskCCN(record.getItem('/DEMO/ORDERS/CCN'))
```

To generate a value based on the input value and mask an additional attribute in which all instances of the input value are masked, use the function:
`CCNMask.maskCCN(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')`

For example, to generate a value based on the CCN attribute and mask additional instances of the input value found in the CUST_INFO attribute, use the following syntax:
`CCNMask.maskCCN(record.getItem('/DEMO/CUSTOMERS/CCN'), '/DEMO/CUSTOMERS/CUST_INFO')`

Argument	Description
inputAttribute	The attribute containing the input value to mask.
additionalAttribute	The additional attribute in which all instances of the input value are masked.

Creating a mask credit card numbers from all providers policy:

You can use the privacy policy editor to create a mask credit card numbers from all providers policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a mask credit card numbers from all providers policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Identity Privacy Policies** and **Credit Card**, and then click **Mask credit card numbers from all providers**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   You can choose to mask the issuer number.

You can use the properties area in the data privacy editor to provide a seed number from which the masked number is generated.

Mask credit card numbers from all providers based on provider name:

The mask credit card numbers from all providers based on provider name policy uses a switch option to mask a CCN based on a value in a selected switch attribute.

For example, if the switch attribute in a row contains the value "VISA", the policy will mask a VISA credit card number in the row.

The switch option is based on the following values: American Express, Diners Club, Discover, JCB, MasterCard, and VISA.

Creating a mask credit card numbers from all providers based on provider name policy:

You can use the privacy policy editor to create a mask credit card numbers from all providers based on provider name policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a mask credit card numbers from all providers based on provider name policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Identity Privacy Policies** and **Credit Card**, and then click **Mask credit card numbers from all providers based on provider name**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   - You must select a source attribute to provide the switch values.
   - You can select a default policy that will be used for values that do not match the switch values.
   - You can choose to mask the issuer number.

You can use the properties area in the data privacy editor to provide a seed number from which the masked number is generated. You can also use the properties area in the data privacy editor to edit the regular expressions that determine the switch values.

National ID policies

Use the national ID policies to mask national ID numbers.

Country specific national ID policies:

The country specific national ID policies mask a specific national ID number.

Creating a country specific national ID policy:

You can use the privacy policy editor to create a country specific national ID policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a country specific national ID policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.

3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Identity Privacy Policies** and **ID**, and then click the national ID policy.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.

You can use the properties area in the data privacy editor to provide a seed number from which the masked number is generated.

Mask Canadian Social Insurance Numbers:

The mask Canadian Social Insurance Numbers policy generates a random Canadian Social Insurance Number (SIN) that includes the first three digits of the source value.

JavaScript policy syntax

This policy is available with the functions `SINMask.randomSIN()` and `SINMask.maskSIN()`.

To generate a random value not based on an input value, use the function: `SINMask.randomSIN()`

To generate a value based on an input value, use the function:
`SINMask.maskSIN(record.getItem('<arg.inputAttribute>'))`

For example, to generate a random value based on an input value, use the following:

```
SINMask.maskSIN(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'))
```

To generate a value based on the input value and mask an additional attribute in which all instances of the input value are masked, use the function:
`SINMask.maskSIN(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')`

For example, to generate a value based on the NATIONAL_ID attribute and mask additional instances of the input value found in the CUST_ID attribute, use the following:

```
SINMask.maskSIN(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'), '/DEMO/CUSTOMERS/
CUST_ID')
```

Argument	Description
inputAttribute	The attribute containing the input value to mask.
additionalAttribute	The additional attribute in which all instances of the input value are masked.

Mask French National Institute for Statistics and Economic Studies Numbers:

The mask French National Institute for Statistics and Economic Studies Numbers policy generates a random French National Institute for Statistics and Economic Studies Number (INSEE) that includes the two digits representing the department number and the two digits representing the control key of the source value.

JavaScript policy syntax

This policy is available with the functions `INSEEMask.randomINSEE()` and `INSEEMask.maskINSEE()`.

To generate a random value not based on an input value, use the function:
`INSEEMask.randomINSEE()`

To generate a value based on an input value, use the function:
`INSEEMask.maskINSEE(record.getItem('<arg.inputAttribute>'))`

For example, to generate a random value based on an input value, use the following:

```
INSEEMask.maskINSEE(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'))
```

To generate a value based on the input value and mask an additional attribute in which all instances of the input value are masked, use the function:
`INSEEMask.maskINSEE(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')`

For example, to generate a value based on the NATIONAL_ID attribute and mask additional instances of the input value found in the CUST_ID attribute, use the following:

```
INSEEMask.maskINSEE(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'), '/DEMO/CUSTOMERS/
CUST_ID')
```

Argument	Description
inputAttribute	The attribute containing the input value to mask.
additionalAttribute	The additional attribute in which all instances of the input value are masked.

Mask Italian Fiscal Code Numbers:

The mask Italian Fiscal Code Numbers policy generates a random Italian Fiscal Code number (CF) that includes the first six digits of the source value.

JavaScript policy syntax

This policy is available with the functions `CFMask.randomCF()` and `CFMask.maskCF()`.

To generate a random value not based on an input value, use the function: `CFMask.randomCF()`

To generate a value based on an input value, use the function:
`CFMask.maskCF(record.getItem('<arg.inputAttribute>'))`

For example, to generate a random value based on an input value, use the following:

```
CFMask.maskCF(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'))
```

To generate a value based on the input value and mask an additional attribute in which all instances of the input value are masked, use the function:
`CFMask.maskCF(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')`

For example, to generate a value based on the NATIONAL_ID attribute and mask additional instances of the input value found in the CUST_ID attribute, use the following:

```
CFMask.maskCF(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'), '/DEMO/CUSTOMERS/CUST_ID')
```

Argument	Description
inputAttribute	The attribute containing the input value to mask.
additionalAttribute	The additional attribute in which all instances of the input value are masked.

Mask Spanish Fiscal Identification Numbers:

The mask Spanish Fiscal Identification Numbers policy generates a random Spanish Fiscal Identification Number (NIF). If the source value includes an X prefix used to identify non-citizens, the prefix is included.

JavaScript policy syntax

This policy is available with the functions `NIFMask.randomNIF()` and `NIFMask.maskNIF()`.

To generate a random value not based on an input value, use the function: `NIFMask.randomNIF()`

To generate a value based on an input value, use the function:
`NIFMask.maskNIF(record.getItem('<arg.inputAttribute>'))`

For example, to generate a random value based on an input value, use the following:

```
NIFMask.maskNIF(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'))
```

To generate a value based on the input value and mask an additional attribute in which all instances of the input value are masked, use the function:
`NIFMask.maskNIF(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')`

For example, to generate a value based on the NATIONAL_ID attribute and mask additional instances of the input value found in the CUST_ID attribute, use the following:

```
NIFMask.maskNIF(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'), '/DEMO/CUSTOMERS/
CUST_ID')
```

Argument	Description
inputAttribute	The attribute containing the input value to mask.
additionalAttribute	The additional attribute in which all instances of the input value are masked.

Mask United Kingdom National Insurance Numbers:

The mask United Kingdom National Insurance Numbers policy generates a random United Kingdom National Insurance Number (NINO) that includes the first two letters (the prefix) and the optional final letter (the suffix) of the source value.

JavaScript policy syntax

This policy is available with the functions `NINOMask.randomNINO()` and `NINOMask.maskNINO()`.

To generate a random value not based on an input value, use the function:
`NINOMask.randomNINO()`

To generate a value based on an input value, use the function:
`NINOMask.maskNINO(record.getItem('<arg.inputAttribute>'))`

For example, to generate a random value based on an input value, use the following:

```
NINOMask.maskNINO(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'))
```

To generate a value based on the input value and mask an additional attribute in which all instances of the input value are masked, use the function:
`NINOMask.maskNINO(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')`

For example, to generate a value based on the NATIONAL_ID attribute and mask additional instances of the input value found in the CUST_ID attribute, use the following:

```
NINOMask.maskNINO(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'), '/DEMO/CUSTOMERS/
CUST_ID')
```

Argument	Description
inputAttribute	The attribute containing the input value to mask.
additionalAttribute	The additional attribute in which all instances of the input value are masked.

Mask United States Social Security Numbers:

The mask United States Social Security Numbers policy generates a random Social Security Number (SSN) that includes the source area number.

An SSN is made of 3 subfields. The first 3 digits (area) represent an area generally determined by the state in which the SSN is issued. The next 2 digits (group) define a group number corresponding to the area number. The last 4 digits (serial) are a sequential serial number. The policy generates a masked SSN with a group number appropriate to the area number.

When this policy is run as a part of a data management service on the executor platform, the executor validates group values by using a high-group file from the U.S. Social Security Administration web site: http://www.socialsecurity.gov/employer/highgroup.txt. If the executor machine cannot access www.socialsecurity.gov, or if you wish to change the location of this file, you must edit the executor eclipse.ini file.

JavaScript policy syntax

This policy is available with the functions `SSNMask.randomSSN()` and `SSNMask.maskSSN()`.

To generate a random value not based on an input value, use the function: `SSNMask.randomSSN()`

To generate a value based on an input value, use the function:
`SSNMask.maskSSN(record.getItem('<arg.inputAttribute>'))`

For example, to generate a random value based on an input value, use the following:

```
SSNMask.maskSSN(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'))
```

To generate a value based on the input value and mask an additional attribute in which all instances of the input value are masked, use the function:
`SSNMask.maskSSN(record.getItem('<arg.inputAttribute>'), '<arg.additionalAttribute>')`

For example, to generate a value based on the NATIONAL_ID attribute and mask additional instances of the input value found in the CUST_ID attribute, use the following:

```
SSNMask.maskSSN(record.getItem('/DEMO/CUSTOMERS/NATIONAL_ID'), '/DEMO/CUSTOMERS/
CUST_ID')
```

Argument	Description
inputAttribute	The attribute containing the input value to mask.
additionalAttribute	The additional attribute in which all instances of the input value are masked.

Mask national ID numbers based on the country name or country code:

The mask national ID numbers based on the country name or country code policy uses a switch option to mask a national ID number based on a value in a selected switch attribute.

For example, if the switch attribute in a row contains the value "USA", the policy will mask a United States Social Security Number in the row.

The switch option is based on the following values:

Canadian Social Insurance Number
CA, CAN, Canada, Canadian, ca, can

Spanish Fiscal Identification Number
ES, Espana, Spain, Spanish, PQH_ES, SPA, ESP, es, pqh_es, spa, esp

French National Institute for Statistics and Economic Studies Number
FR, France, French, FRE, PQH_FR, FRA, fr, fre, fra, pqh_fr

Italian Fiscal Code Number
IT, Italy, Italian, ITA, PQH_IT, it, ita, pqh_it

United Kingdom National Insurance Number
UK, U.K., United Kingdom, Great Britain, England, Scotland, Wales, Northern Ireland, British, English, Welsh, Scottish, BRI, PQH_GB, WEL, SCO, GBR, GB, G.B., uk, bri, pqh_gb, wel, sco, gbr, gb

United States Social Security Number
US, U.S., USA, U.S.A., American, AM, us, usa, am

Creating a mask national ID numbers based on the country name or country code policy:

You can use the privacy policy editor to create a mask national ID numbers based on the country name or country code policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a mask national ID numbers based on the country name or country code policy:
1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Identity Privacy Policies** and **ID**, and then click **Mask national ID numbers based on the country name or country code**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   - You must select a source attribute to provide the switch values.
   - You can select a default policy that will be used for values that do not match the switch values.

You can use the properties area in the data privacy editor to provide a seed number from which the masked number is generated. You can also use the properties area in the data privacy editor to edit the regular expressions that determine the switch values.

Identity privacy lookup policies

The identity privacy lookup policies replace values from selected source entities with values from corresponding lookup table columns, thereby masking the source values.

The identity privacy lookup policies are bound to lookup tables in the EXTENDED_LOOKUP schema of the Optim sample data. When you create a lookup policy, you will map attributes in the source data with the attributes bound to the policy.

Before you can create a identity lookup privacy policy, you must define a lookup data source for the executor platform. The data source must include the tables from the EXTENDED_LOOKUP schema provided with the sample data.

Random and hash lookup

There are two options for identity privacy lookup processing, random and hash lookup. You can select the lookup option when you create a privacy policy.

A lookup table includes a column containing contiguous sequential values. Rows in the lookup table are selected by matching the sequential values in the lookup table with a value generated using the random or hash option.

Random Lookup
A random lookup selects a row at random from the lookup table to obtain replacement values.

Hash Lookup

In hash lookup processing, the replacement values are selected by hashing a source value and using the hashed value as an index to a row in the lookup table. A source column that is hashed does not need to be a column that will be replaced by lookup table values. The maximum length of the source and lookup columns is 256 characters. The hash function is case-sensitive, and you can convert a source value to upper case before it is hashed.

If a source column used to derive the hashed value contains certain values (NULL, spaces (for CHAR columns), and zero-length VARCHAR), the value is not hashed and the following reserved values are used as keys to the lookup table:

Source Value	Lookup Table Key
NULL	-1
spaces (CHAR or VARCHAR)	-2
zero-length VARCHAR	-3
multiple hash lookup columns where all values are one or more of the following values: NULL, spaces (CHAR or VARCHAR), or zero-length VARCHAR	-4

Switch option

There a several lookup policies that use a switch value option to mask data based on a value in a selected switch attribute. The switch option is based on a country name or country code. For example, if the switch attribute in a row contains the value ⌂USA⌂, the policy will use data specific to the United States to mask data in the row.

The switch option uses the following values:

AU - Australia
AU, au, OZ, oz, Australia, australia

CA - Canada
CA, CAN, Canada, Canadian, ca, can

DE - Germany
DE, de, Deutschland, deutschland, GER, ger, Germany, germany, FRG, frg, BRD, brd, Bundesrepublik Deutschland

ES - Spain
ES, Espana, Spain, Spanish, PQH_ES, SPA, ESP, es, pqh_es, spa, esp

FR - France
FR, France, French, FRE, PQH_FR, FRA, fr, fre, fra, pqh_fr

IT - Italy
IT, Italy, Italian, ITA, PQH_IT, it, ita, pqh_it

JP - Japan
JP, Japan, Japanese, jp, Nippon, Nihon

UK - United Kingdom
UK, U.K., United Kingdom, Great Britain, England, Scotland, Wales, Northern Ireland, British, English, Welsh, Scottish, BRI, PQH_GB, WEL, SCO, GBR, GB, G.B., uk, bri, pqh_gb, wel, sco, gbr, gb

US - United States
US, U.S., USA, U.S.A., American, AM, us, usa, am

Creating an identity privacy lookup policy:

You can use the privacy policy editor to create an identity privacy lookup policy.

You can add a policy to a data access plan that includes a selection policy.

Before you can create a lookup policy, you must define an executor lookup data source.

To create a lookup policy:
1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Identity Privacy Policies** and the policy category, and then click the policy.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   You can use the properties area in the data privacy editor to provide a seed number from which the masked number is generated.
8. Complete the steps of the wizard.
   - You must select a source attribute to provide the input value.
   - You must map the source attributes to the attributes bound to the policy.
   - You must select either the random or hash lookup option. If you select the hash option, you must do the following steps:
     – Select the attributes used to generate the hash value
     – Set the order for these attributes
   - If you choose a policy that uses a switch option, you must select a source attribute to provide the switch value.

You can use the properties area in the data privacy editor to do the following actions:
- provide a seed number used to generate a value for selecting rows in the lookup table
- specify the values to ignore when the hash value is generated
- edit the regular expressions that determine the switch values

Address information lookup policies:

The address information policies mask addresses in the following countries: Australia, Canada, France, Germany, Italy, Japan, Spain, United Kingdom, and United States. The policies describe the following attributes:

There is a country specific version of each policy. For example, mask United States address information.

There is also a policy (mask address information based on the country name or country code) that uses a switch value option to mask address information based on a value in a selected switch attribute. For example, if the switch attribute in a row of data contains the value ⌂USA⌂, the policy will use the mask United States address information policy to mask data in the row.

The policies mask data mapped to the following attributes:

Attribute	Description
Address1	First line of a street address.
Address2	Second line of a street address.
City	City name.
StateOrProvince	State or province name.
ZipOrPostalCode1	First part of a ZIP or postal code.

Personal information lookup policies:

The personal information policies mask personal data for people in the following countries: Australia, Canada, France, Germany, Italy, Japan, Spain, United Kingdom, and United States.

There is a country specific version of each policy. For example, mask United States personal information.

There is also a policy (mask personal information based on the country name or country code) that uses a switch value option to mask personal information based on a value in a selected switch attribute. For example, if the switch attribute in a row of data contains the value ⌂USA⌂, the policy will use the mask United States personal information policy to mask data in the row.

The policies mask data mapped to the following attributes:

Attribute	Description
Id	National ID.
FirstName	First name.
LastName	Last name.
Company	Company name.
Gender	Male or female.
Phone	Phone number.
BirthDate	Birth date.
EMailAddress	E-mail address.

Given name information lookup policies:

The given name information policies mask given names for people in the following countries: Australia, Canada, France, Germany, Italy, Japan, Spain, United Kingdom, and United States. For each country, there is a gender neutral policy and policies for each gender. The policies describe the following attributes:

For each country, there is a form of the policy for each gender. For example, mask United States female given name information and mask United States male given name information.

For each country, there is also a gender neutral form of the policy. For example, mask United States given name information.

There are also policies that use a switch value to mask given name information based on a value in a selected switch attribute. There is a switch value policy for each gender: mask a female given name based on the country name or country code and mask a male given name based on the country name or country code. There is also a gender neutral form of the policy: mask a given name based on the country name or country code.

For example, if the switch attribute in a row of data contains the value ⌂USA⌂, the policy will use the mask United States given name information policy to mask data in the row.

The policies mask data mapped to the following attribute:

Attribute	Description
FirstName	Given name (first name).

Surname information lookup policies:

The surname information policies mask surnames for people in the following countries: Australia, Canada, France, Germany, Italy, Japan, Spain, United Kingdom, and United States.

There is a country specific version of each policy. For example, mask United States surname information.

There is also a policy (mask a surname based on the country name or country code) that uses a switch value option to mask surname information based on a value in a selected switch attribute. For example, if the switch attribute in a row of data contains the value ⌂USA⌂, the policy will use the mask United States surname information policy to mask data in the row.

The policies mask data mapped to the following attribute:

Attribute	Description
LastName	Surname (last name).

Mask a company name lookup policy:

The mask a company name policy masks company names.

The policy masks data mapped to the following attribute:

Attribute	Description
CompanyName	Company name.

Numeric privacy policies

Use the numeric privacy policies to mask numeric data by generating random values. The policies include the rule-based and JavaScript options.

Gaussian random double

The Gaussian random double policy generates a random double-precision floating-point number. The generated number is based on a Gaussian, bell-shaped curve.

In a Gaussian distribution, numbers near the mean are more likely to be selected than numbers outside the mean, as opposed to a uniform distribution of random numbers. In a uniform distribution of random numbers from 1 to 10, the number of ones generated is roughly equal to the number of fives or tens generated. In a Gaussian distribution with a mean of 6 and a standard deviation of 2, more fives and sevens are generated than threes and nines.

JavaScript policy syntax

This policy is available with the function `ScrambleMask.gaussianRandomDouble()`.

```
ScrambleMask.gaussianRandomDouble(<arg:mean>, <arg:standardDeviation>)
```

For example, to generate values based on a mean of 50.5 and a standard deviation of 10.00, enter the following syntax:

```
ScrambleMask.gaussianRandomDouble('50.5', '10.00')
```

Argument	Description
mean	The mean value for the Gaussian distribution.
standardDeviation	The standard deviation for the Gaussian distribution.

Creating a Gaussian random double policy:

You can use the privacy policy editor to create a Gaussian random double policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a Gaussian random double policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Numeric Privacy Policies** and then click **Gaussian random double**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.

   You must provide a mean value to set the midpoint of the bell curve and a standard deviation value to determine the width of the curve (a range, relative to the mean, in which most values fall).

Gaussian random integer

The Gaussian random integer policy generates a random integer. The generated number is based on a Gaussian, bell-shaped curve.

In a Gaussian distribution, numbers near the mean are more likely to be selected than numbers outside the mean, as opposed to a uniform distribution of random numbers. In a uniform distribution of random numbers from 1 to 10, the number of ones generated are roughly equal to the number of fives or tens generated. In a Gaussian distribution with a mean of 6 and a standard deviation of 2, more fives and sevens are generated than threes and nines.

JavaScript policy syntax

This policy is available with the function `ScrambleMask.gaussianRandomInteger()`.

```
ScrambleMask.gaussianRandomInteger(<arg:mean>, <arg:standardDeviation>)
```

For example, to generate values based on a mean of 100 and a standard deviation of 20, enter the following syntax:

```
ScrambleMask.gaussianRandomInteger('100', '20')
```

Argument	Description
mean	The mean value for the Gaussian distribution.
standardDeviation	The standard deviation for the Gaussian distribution.

Creating a Gaussian random integer policy:

You can use the privacy policy editor to create a Gaussian random integer policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a Gaussian random integer policy:
1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Numeric Privacy Policies** and then click **Gaussian random integer**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   You must provide a mean value to set the midpoint of the bell curve and a standard deviation value to determine the width of the curve (a range, relative to the mean, in which most values fall).

Random number function

The random number function generates numbers selected at random within the range indicated by the low and high values.

You can use the random number function to replace character or numeric data. The low and high values must be integers within the range -2,147,483,648 to 2,147,483,647. The low value must be less than the high value.

Creating a random number function policy:

You can use the privacy policy editor to create a random number function policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a random number function policy:
1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Numeric Privacy Policies** and then click **Random number function**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   You must enter the low and high integer values (from -2,147,483,648 to 2,147,483,647) to define a range for generating random numbers. The low value must be less than the high value.

Sequential number function

The sequential number function generates numbers that are incremented sequentially.

You can use the sequential number function to replace character or numeric data. You must enter a start value and a value by which the numbers are incremented. The start and incremental values must be integers within the range -2,147,483,648 to 2,147,483,647.

The generated value is limited by the data type and length of the destination column. If the generated value exceeds the length of the destination column, the function automatically resets to the start value.

Creating a sequential number function policy:

You can use the privacy policy editor to create a sequential number function policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a sequential number function policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Numeric Privacy Policies** and then click **Sequential number function**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.

   You must enter a start value and a value by which the numbers are incremented. The start and incremental values must be integers within the range -2,147,483,648 to 2,147,483,647.

Uniform random double in range

The uniform random double in range policy generates a random double-precision floating-point number within a specified range. The generated number is based on a uniform distribution.

In a uniform distribution of random numbers from 1 to 10, the number of ones generated is roughly equal to the number of fives or tens generated.

JavaScript policy syntax

This policy is available with the function `ScrambleMask.uniformRandomDoubleInRange()`.

```
ScrambleMask.uniformRandomDoubleInRange(<arg:rangeLowerBound>, <arg:rangeUpperBound>)
```

For example, to generate a value from .01 to 99.99, enter the following syntax:

```
ScrambleMask.uniformRandomDoubleInRange('.01', '99.99')
```

Argument	Description
rangeLowerBound	The start of the range. The minimum value generated.
rangeUpperBound	The end of the range. The maximum value generated.

Creating a uniform random double in range policy:

You can use the privacy policy editor to create a uniform random double in range policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a uniform random double in range policy:
1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Numeric Privacy Policies** and then click **Uniform random double in range**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   You must provide double-precision floating-point numbers as the start and end of the range. The start and end numbers are included in the range.

Uniform random long in range

The uniform random long in range policy generates a random long integer within a specified range. The generated number is based on a uniform distribution.

In a uniform distribution of random numbers from 1 to 10, the number of ones generated is roughly equal to the number of fives or tens generated.

JavaScript policy syntax

This policy is available with the function `ScrambleMask.uniformRandomLongInRange()`.

```
ScrambleMask.uniformRandomLongInRange(<arg:rangeLowerBound>, <arg:rangeUpperBound>)
```

For example, to generate a value from 2000000000 to 3000000000, enter the following syntax:

```
ScrambleMask.uniformRandomLongInRange('2000000000', '3000000000')
```

Argument	Description
rangeLowerBound)	The start of the range. The minimum value generated.
rangeUpperBound	The end of the range. The maximum value generated.

Creating a uniform random long in range policy:

You can use the privacy policy editor to create a uniform random long in range policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a uniform random long in range policy:
1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Numeric Privacy Policies** and then click **Uniform random long in range**.
5. From the **Attributes** list, select the attribute to mask with the policy.

6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.

   You must provide a long integer as the start and end of the range. The start and end numbers are included in the range.

Scramble privacy policies

Use the scramble privacy policies to mask character and numeric data types. The policies include the rule-based and JavaScript options.

Repeatable replacement

The repeatable replacement policy uses a repeatable method to mask a string with characters that match each type of character that is replaced. For example, numbers are replaced with numbers, and lowercase letters are replaced with lowercase letters. The characters used for masking are obtained from a specified character set. The policy masks characters that are part of the character set only.

The following mask methods are available:

CRC The cyclic redundancy check (CRC) method masks each string in a repeatable manner; however, the CRC method may not mask each string with a unique string.

Hash The hash method masks each string in a repeatable manner; however, the hash method may not mask each string with a unique string.

Map The map method masks each string in a repeatable manner and with a unique string.

JavaScript policy syntax

This policy is available with the function `ScrambleMask.repeatableReplacement()`.

```
ScrambleMask.repeatableReplacement(record.getItem('<arg:inputAttribute>'),
'<arg:language>', '<arg:scrambleType>')
```

For example, to mask a string with characters from an English character set using the CRC method, enter the following syntax:

```
ScrambleMask.repeatableReplacement(record.getItem('/DEMO/ORDERS/ORDER_ID'), 'English' ,
'CRC')
```

Argument	Description
inputAttribute	The attribute containing the string to mask.
language	The language for the character set that provides the characters used for masking. If a character set is not specified, or if the character set is not supported, the English character set is used. For a list of supported character sets, see "Language character sets supported for scramble mask policies" on page 72.
scrambleType	The mask method: CRC, HASH, or MAP.

Creating a repeatable replacement policy:

You can use the privacy policy editor to create a repeatable replacement policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a repeatable replacement policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.

4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Scramble Privacy Policies** and then click **Repeatable replacement**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   - You must select a character set that provides the characters used for masking.
   - You must select one of the following mask methods:

     CRC The cyclic redundancy check (CRC) method masks each string in a repeatable manner; however, the CRC method may not mask each string with a unique string.

     Hash The hash method masks each string in a repeatable manner; however, the hash method may not mask each string with a unique string.

     Map The map method masks each string in a repeatable manner and with a unique string.

Repeatable replacement by regular expression

The repeatable replacement by regular expression policy uses a repeatable method to mask a string with characters that match each type of character that is replaced. For example, numbers are replaced with numbers, and lowercase letters are replaced with lowercase letters. The policy uses a regular expression to determine which characters to mask in the string. The characters used for masking are obtained from a specified character set. The policy masks characters that are part of the character set only.

The following mask methods are available:

CRC The cyclic redundancy check (CRC) method masks each string in a repeatable manner; however, the CRC method may not mask each string with a unique string.

Hash The hash method masks each string in a repeatable manner; however, the hash method may not mask each string with a unique string.

Map The map method masks each string in a repeatable manner and with a unique string.

JavaScript policy syntax

This policy is available with the function `ScrambleMask.repeatableReplacementByRegularExpression()`.

```
ScrambleMask.repeatableReplacementByRegularExpression(record.getItem(
'<arg:inputAttribute>'), '<arg:regularExpression>',  '<arg:language>',
'<arg:scrambleType>')
```

For example, to mask the lowercase characters from a-h with characters from an English character set using the CRC method, enter the following syntax:

```
ScrambleMask.repeatableReplacementByRegularExpression(record.getItem('/DEMO/ORDERS/
ORDER_ID'), '([a-h]+)', 'English', 'CRC')
```

Argument	Description
inputAttribute	The attribute containing the string to mask.
regularExpression	A regular expression describing the characters to mask in the input string.
language	The language for the character set that provides the characters used for masking. If a character set is not specified, or if the character set is not supported, the English character set is used. For a list of supported character sets, see "Language character sets supported for scramble mask policies" on page 72.
scrambleType	The mask method: CRC, HASH, or MAP.

Creating a repeatable replacement by regular expression policy:

You can use the privacy policy editor to create a repeatable replacement by regular expression policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a repeatable replacement by regular expression policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Scramble Privacy Policies** and then click **Repeatable replacement by regular expression**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   - You must provide a regular expression that specifies the characters to mask.
   - You must select a character set that provides the characters used for masking.
   - You must select one of the following mask methods:

     CRC The cyclic redundancy check (CRC) method masks each string in a repeatable manner; however, the CRC method may not mask each string with a unique string.

     Hash The hash method masks each string in a repeatable manner; however, the hash method may not mask each string with a unique string.

     Map The map method masks each string in a repeatable manner and with a unique string.

Replace characters

The replace characters policy masks each character in a string with a randomly generated character that matches the type of character that is replaced. For example, numbers are replaced with numbers, and lowercase letters are replaced with lowercase letters. The characters used for masking are obtained from a specified character set. The policy masks characters that are part of the character set only.

JavaScript policy syntax

This policy is available with the function `ScrambleMask.replaceCharacters()`.

```
ScrambleMask.replaceCharacters(record.getItem('<arg:inputAttribute>'),'<arg:language>')
```

For example, to replace values in a string with characters from an English character set, enter the following syntax:

```
ScrambleMask.replaceCharacters(record.getItem('/DEMO/ORDERS/ORDER_ID'),'English')
```

Argument	Description
inputAttribute	The attribute containing the string to mask.
language	The language for the character set that provides the characters used for masking. If a character set is not specified, or if the character set is not supported, the English character set is used. For a list of supported character sets, see "Language character sets supported for scramble mask policies" on page 72.

Creating a replace characters policy:

You can use the privacy policy editor to create a replace characters policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a replace characters policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Scramble Privacy Policies** and then click **Replace characters**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   You must select a character set that provides the characters used for masking.

Replace characters by regular expression

The replace characters by regular expression policy masks each character in a string with a randomly generated character that matches the type of character that is replaced. For example, numbers are replaced with numbers, and lowercase letters are replaced with lowercase letters. The policy uses a regular expression to determine which characters to mask in the string. The characters used for masking are obtained from a specified character set. The policy masks characters that are part of the character set only.

JavaScript policy syntax

This policy is available with the function `ScrambleMask.replaceCharactersByRegularExpression()`.

```
ScrambleMask.replaceCharactersByRegularExpression(record.getItem(
'<arg:inputAttribute'), <arg:regularExpression>, <arg:language>)
```

For example, to replace the lowercase characters from a-h with characters from an English character set, enter the following syntax:

```
ScrambleMask.replaceCharactersByRegularExpression(record.getItem('/DEMO/ORDERS/
ORDER_ID'), '([a-h]+)', 'English')
```

Argument	Description
inputAttribute	The attribute containing the string to mask.
regularExpression	A regular expression describing the characters to mask in the input string.
language	The language for the character set that provides the characters used for masking. If a character set is not specified, or if the character set is not supported, the English character set is used. For a list of supported character sets, see "Language character sets supported for scramble mask policies" on page 72.

Creating a replace characters by regular expression policy:

You can use the privacy policy editor to create a replace characters by regular expression policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a replace characters by regular expression policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Scramble Privacy Policies** and then click **Replace characters by regular expression**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.

   You must provide a regular expression that specifies the characters to mask. You must select a character set that provides the characters used for masking.

Scramble characters

The scramble characters policy masks a string by randomly changing the order of the characters in the string.

JavaScript policy syntax

This policy is available with the function `ScrambleMask.scrambleCharacters()`.

```
ScrambleMask.scrambleCharacters(record.getItem('<arg.inputAttribute>'))
```

For example:

```
ScrambleMask.scrambleCharacters(record.getItem('/DEMO/ORDERS/ORDER_ID'))
```

Argument	Description
inputAttribute	The attribute containing the string to mask.

Creating a scramble characters policy:

You can use the privacy policy editor to create a scramble characters policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a scramble characters policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Scramble Privacy Policies** and then click **Scramble characters**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**.

Scramble characters by regular expression

The scramble characters by regular expression policy masks a string by randomly changing the order of the characters in the string. The policy uses a regular expression to determine which characters to mask in the string.

JavaScript policy syntax

This policy is available with the function `ScrambleMask.scrambleCharactersByRegularExpression()`.

```
ScrambleMask.scrambleCharactersByRegularExpression(record.getItem(
'<arg.inputAttribute'), <arg:regularExpression>)
```

For example, to swap the lowercase characters from a-h, enter the following syntax:

```
ScrambleMask.scrambleCharactersByRegularExpression(record.getItem('/DEMO/ORDERS/
ORDER_ID'), '([a-h]+)')
```

Argument	Description
inputAttribute	The attribute containing the string to mask.
regularExpression	A regular expression describing the characters to mask in the input string.

Creating a scramble characters by regular expression policy:

You can use the privacy policy editor to create a scramble characters by regular expression policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a scramble characters by regular expression policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Scramble Privacy Policies** and then click **Scramble characters by regular expression**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.

   You must provide a regular expression that specifies the characters to mask.

Simple scramble characters

The simple scramble characters policy masks a string by randomly swapping the characters in the string among themselves. The policy masks a string in a repeatable manner.

JavaScript policy syntax

This policy is available with the function `ScrambleMask.simpleScramble()`.

```
ScrambleMask.simpleScramble(record.getItem('<arg.inputAttribute>'))
```

For example:

```
ScrambleMask.simpleScramble(record.getItem('/DEMO/ORDERS/ORDER_ID'))
```

Argument	Description
inputAttribute	The attribute containing the string to mask.

Creating a simple scramble characters policy:

You can use the Add Policy wizard to create a simple scramble characters policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a simple scramble characters policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Scramble Privacy Policies** and then click **Simple scramble characters**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**.

Language character sets supported for scramble mask policies

The replace characters, replace characters by regular expression, repeatable replacement, and repeatable replacement by regular expression policies support multiple language character sets for masking.

When entering a character set in a JavaScript policy, use the following values.

English, Afrikaans, Albanian, Arabic, Armenian, Assamese, Azerbaijani, Belarusian, Bengali, Bulgaria, Catalan, Chinese (Simplified), Chinese (Traditional), Croatian, Czech, Danish, Dutch, Estonian, French, Finnish, German, Georgian, Gujarati, Indonesian, Hebrew, Hindi, Hungarian, Greek, Icelandic, Italian, Japanese, Kannada, Kazakh, Konkani, Korean, Latvian, Lithuanian, Macedonian, Malay, Malayalam, Maltese, Marathi, Nepali, Norwegian, Oriya, Portuguese (Brazil), Portuguese (Portugal), Punjabi, Polish, Romanian, Russian, Serbian (Cyrillic), Serbian (Latin), Sinhala, Slovak, Slovenian, Spanish, Swahili, Swedish, Tamil, Telugu, Thai, Turkish, Ukrainian, Urdu, Vietnamese, Welsh

Generic lookup privacy policies

Use the generic lookup privacy policies to select values from a lookup table that are used to populate a destination entity. You select the lookup table from a data source connection.

Before you can create a generic lookup privacy policy, you must define a lookup data source for the target platform of the policy.

Lookup policy

The lookup policy uses a lookup table to mask data according to values in a source attribute. The policy finds matching attribute values in the source data and lookup table, and using data from the lookup table row with the matched value, the policy masks the row containing the source value.

The policy can mask values in one or more attributes.

Creating a lookup policy:

You can use the privacy policy editor to create a lookup policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

Before you can create a lookup policy, you must define a lookup data source for the target platform selected for the policy.

To create a lookup policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Generic Lookup Privacy Policies** and then click **Lookup**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   - You must select a data source connection. You must also select the lookup table schema and name. You can enter a DB alias associated with the lookup table.
   - You must map columns in the lookup table to attributes in the source entity that will be masked.
   - You must select a column in the lookup table that contains values to match against the search value from the source attribute.

Hash lookup policy

The hash lookup policy uses a lookup table to mask data according to a hashed value derived from a source attribute.

In hash lookup processing, the replacement values are selected by hashing a source value and using the hashed value as an index to a row in the lookup table. A source column that is hashed does not need to be a column that will be replaced by lookup table values. The maximum length of the source and lookup columns is 256 characters.

You can use options to specify characters that will be trimmed from the source value and convert the value to upper case before it is hashed. You can also enter a seed value to vary the calculation performed by the hashing algorithm.

The lookup table must include a key column that contains sequential number values without any gaps, and the remaining columns contain replacement values. The key column must be a numeric data type. The lookup table is typically indexed. The function hashes a source attribute to derive sequential numbers from 1 to the maximum value in the key column of the lookup table. The hashed value from the source attributte is matched with the sequential numbers in the lookup table, and values from the corresponding lookup table row are inserted at the destination.

If a source column used to derive the hashed value contains certain values (NULL, spaces (for CHAR columns), and zero-length VARCHAR), the value is not hashed and the following reserved values are used as keys to the lookup table:

Source Value	Lookup Table Key
NULL	-1
spaces (CHAR or VARCHAR)	-2
zero-length VARCHAR	-3
multiple hash lookup columns where all values are one or more of the following values: NULL, spaces (CHAR or VARCHAR), or zero-length VARCHAR	-4

Creating a hash lookup policy:

You can use the privacy policy editor to create a hash lookup policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

Before you can create a hash lookup policy, you must define a lookup data source for the target platform selected for the policy.

To create a hash lookup policy:
1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Generic Lookup Privacy Policies** and then click **Hash Lookup**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.
   - You must select a data source connection and enter a DB alias associated with the lookup table. You must also select the lookup table schema and name.
   - You must map columns in the lookup table to attributes in the source entity that will be masked.
   - You must select an attribute that will provide the values to be hashed.
   - You can specify characters that will be trimmed from the source value before it is hashed.
   - You can enter a seed value to vary the calculation performed by the hashing algorithm.
   - You must select a column in the lookup table that contains values to match against the search value from the source attribute.

Random lookup policy

The random lookup policy uses a lookup table to mask data by selecting a random value.

The policy generates a random number between 1 and the limit or number of rows in the lookup table to use as a subscript into the table. The column value or values from the row that correspond to the subscript are inserted in the destination attribute.

You can set a limit on the number of rows from the lookup table used to select values for masking. Specify an integer, up to a maximum value of 2,000,000,000.

Creating a random lookup policy:

You can use the privacy policy editor to create a random lookup policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

Before you can create a random lookup policy, you must define a lookup data source for the target platform selected for the policy.

To create a random lookup policy:
1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, expand **Generic Lookup Privacy Policies** and then click **Random Lookup**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.

7. Complete the steps of the wizard.
   - You must select a data source connection and enter a DB alias associated with the lookup table. You must also select the lookup table schema and name.
   - You must map columns in the lookup table to attributes in the source entity that will be masked.
   - You can set a limit on the number of rows from the lookup table used to select values for masking. Specify an integer, up to a maximum value of 2,000,000,000.

Random shuffle function

The random shuffle function replaces a value with another value from the source attribute.

The source row and the row that contains the replacement value will never be the same, but depending on your data, source and replacement values can be identical. You can indicate the number of times the function will refetch a replacement value until a value that does not match the source value is found (a "retry"), or you can allow a replacement value to match the source. The default retry value is 10.

Creating a random shuffle function

You can use the privacy policy editor to create a random shuffle function policy in a data access plan.

You can add a policy to a data access plan that includes a selection policy.

To create a random shuffle function policy:

1. Expand a **Data Access Plans** folder for a logical data model package in the Data Project Explorer.
2. Double-click the data access plan to include the policy. The data access plan editor opens.
3. Click **Data Privacy**. The privacy policy editor opens.
4. From the **Policies** area, do the following:
   a. Select a **Platform** that will process the policy.
   b. From the **Policy** list, click **Random shuffle function**.
5. From the **Attributes** list, select the attribute to mask with the policy.
6. Click **Apply**. The Add Policy wizard opens.
7. Complete the steps of the wizard.

   You can enter the number of times the function should look for a replacement value that does not match the original value. Enter a value from 0-1000. The default is 10.

JavaScript policies

A JavaScript policy applies to source attributes in a data management service and is executed when the service is run. Use a service plan to add a JavaScript policy to a service.

A JavaScript policy includes JavaScript files that describe data transformations applied to attributes in a service. You can use JavaScript files with multiple attributes in an entity, but you can associate each attribute with only one file. A service can include multiple JavaScript policies. The JavaScript files are stored in the **Other Files** folder in the Data Project Explorer.

Use JavaScript to define a data transformation for an attribute. You can use JavaScript to mask numbers with random values, extract substrings, concatenate entity values, and perform other data transformations available by leveraging JavaScript. You can also use JavaScript functions to apply the date privacy, identity privacy, numeric privacy, and scramble privacy policies.

The transformations defined in a JavaScript policy occur after Optim performs any lookup or rule-based transformations on the source data.

Retrieving a source value

Use the "record" object to reference the source logical data model and the getItem() method to access the source attribute. For relational data, a source item is identified in the format ('*/schema/entity/attribute*').

For example, to retrieve the source ADDRESS attribute in the CUSTOMERS entity of the DEMO schema, use the following syntax:

```
record.getItem('/DEMO/CUSTOMERS/ADDRESS')
```

Concatenating strings

To concatenate strings, use the "+" operator instead of the concat() function.

Handling a source value

When a source value is processed by the record.getItem() method, the value is converted to a Java data type during JavaScript processing. After JavaScript processing, the value is converted to the target database's data type.

Use the following table to determine how source data types are converted during JavaScript processing.

Source data type	Java type
Character	java.lang.String
Character Varying	java.lang.String
National Character	java.lang.String
National Character Varying	java.lang.String
Character Large Object	byte[] (for IBM DB2, java.sql.Clob) (for Oracle, char[])
National Character Large Object	byte[](for IBM DB2, java.sql.Clob) (for Oracle, char[])
Binary	byte[]
Binary Varying	byte[]
Binary Large Object	java.sql.Blob
Boolean	java.lang.Boolean
Date	java.util.Calendar
Time	java.util.Calendar
Timestamp	java.sql.Timestamp (for Oracle, java.lang.Object)
Numeric	java.math.BigDecimal
Decimal	java.lang.String
Double Precision	java.lang.Double (for Oracle, java.lang.String)
Real	java.lang.Double
Float	java.lang.Double (for Oracle, java.math.BigDecimal)
Small Integer	java.lang.Short
Integer	java.lang.Integer
Big Integer	java.lang.Long
Interval	java.lang.Object
XML	java.lang.Object
Datalink	java.lang.Object

Creating a JavaScript policy

You can use the Add Policy wizard to create a JavaScript policy for a service plan.

To create a JavaScript policy:

1. From the Data Project Explorer, expand the **Services** folder and then open the service request that contains the service plan to which you will add the policy.
2. Double-click the **Service Plan** node. The service plan editor opens.
3. Click **Add Policy**. The Add Policy wizard opens.
4. Complete the steps of the wizard.

   Select **JavaScript Policy** on the Policy Selector page.

Adding a JavaScript file to a JavaScript policy

You can use the Add JavaScript File wizard to add a JavaScript file to a JavaScript policy. The JavaScript file will apply to an attribute in a source entity in the service. After you add the file, you can use an editor to add an expression to the file.

To add a JavaScript file to a JavaScript policy:

1. From the Data Project Explorer, expand the **Services** folder and then open the service request that contains the service plan with the policy.
2. Double-click the **Service Plan** node. The service plan editor opens.
3. In the **Target Policies** list, select the JavaScript policy. The JavaScript Policy editor opens.
4. Click **Add JavaScript**. The Add JavaScript File wizard opens.
5. Complete the steps of the wizard.

   You must select an attribute to which the JavaScript file will apply, and enter a file name. After you complete the wizard, an editor will open.
6. Use the editor to enter a JavaScript expression in the file.
7. Click **File** > **Save** to save the file. The file will be listed in the JavaScript Policy editor and stored in the **Other Files** folder in the Data Project Explorer.

Editing a JavaScript file in a JavaScript policy

You can edit a JavaScript file in a JavaScript policy.

To edit a JavaScript file in a JavaScript policy:

1. From the Data Project Explorer, expand the **Services** folder and then open the service request that contains the service plan with the policy.
2. Double-click the **Service Plan** node. The service plan editor opens.
3. In the **Target Policies** list, select the JavaScript policy. The JavaScript Policy editor opens.
4. Select the JavaScript file and click **Edit**. The JavaScript file opens in a editor.
5. Edit the file. Click **File** > **Save** to save the file.

JavaScript expression examples

The following examples illustrate common JavaScript expressions.

Substring

To extract a substring of the first 15 characters of the CITY attribute value, use the following syntax:

```
record.getItem('/DEMO/CUSTOMERS/CITY').substr(0,15)
```

Concatenate

To concatenate a value in the ADDRESS attribute with values in the CITY and STATE attributes, separating each value with a space, use the following syntax:

```
 record.getItem('/DEMO/CUSTOMERS/ADDRESS')+ ' ' +record.getItem('/DEMO/CUSTOMERS/CITY')+ ' '
+record.getItem('/DEMO/CUSTOMERS/STATE')
```

If-Else Statement

To prevent errors, use an if-else statement to ignore the substring method when the length of the attribute value is less than the length of the substring. In the following syntax, the substring method is not used if a value in CUSTNAME is less than or equal to 8 characters:

```
 var maxLength = 8 if ( record.getItem('/DEMO/CUSTOMERS/CUSTNAME').toString().length() >
maxLength ) { record.getItem('/DEMO/CUSTOMERS/CUSTNAME').substr( 0, maxLength ) }  else {
record.getItem('/DEMO/CUSTOMERS/CUSTNAME') }
```

Date Manipulation

To return a random date in YYYY-MM-DD format (for use with the java.sql.Date class), use the Date() method to obtain the current date and the setDate() method to add a random number of days (from 0 to 365) to the date. Then concatenate the values returned by the getFullYear(), getMonth(), and getDate() methods to return the new date in YYYY-MM-DD format. Use the following syntax:

```
 var dob=new Date(); dob.setDate(dob.getDate()+Math.floor(Math.random()*365))
dob.getFullYear()+'-'+dob.getMonth()+'-'+dob.getDate()
```

Data privacy compliance requirements

When you define a data access plan that includes entities with data privacy compliance requirements, data privacy policies are automatically created for these entities.

You can create atomic domains in a domain model and define data privacy compliance requirements for the domain. When the atomic domain is associated with a column data type in a physical model, the compliance information is also attached. When such a physical data model is transformed to an Optim logical data model, the compliance information is carried forward in the logical data model. When a data access plan includes an entity with a data privacy compliance requirement, data privacy policies are automatically created based on the compliance requirement.

Based on the data privacy policy assigned to an attribute as part of a compliance requirement, Optim Designer will define a policy for the entity that contains the attribute. Depending on the data privacy policy, you must use the data privacy editor to complete the policy by entering missing properties. A data privacy policy that has missing properties will display an error status in the **Data privacy policies in use** list of the data privacy editor.

Using the data privacy editor

You can use the data privacy editor to add or edit a data privacy policy.

Apply data privacy policies

Use the **Apply data privacy policies** area to define a data privacy policy.

To create a data privacy policy, select a policy, select an attribute to mask, and then click Apply. You can filter the Policies list by name and by the platform that will process the policy. The following platforms are available:

- **Distributed** - Optim and Optim Solutions
- **Executor** - Optim Data Privacy Solution
- **z/OS** - Optim for z/OS and Optim for z/OS Solutions

The **Attributes** list indicates if a data privacy enforcement requirement is defined for each attribute.

Data privacy policies in use

Use the **Data privacy policies in use** list to view data privacy policies applied to the data access plan. You can select a policy to view and edit properties in the policy.

The **Data privacy policies in use** list indicates if a data privacy enforcement requirement is defined for an attribute in the policy. The list also indicates if the policy is in compliance with the requirement. A policy is in compliance if it masks all attributes with a data privacy enforcement requirement.

If a policy is missing required property values, the policy will be in error status and display the following icon:  . You can use the policy properties area to enter missing values.

Policy properties

Use the policy properties area to view and edit properties in a data privacy policy. To view the properties for a policy, select the policy in the **Data privacy policies in use** list.

If a policy is missing required property values, the policy will be in error status and display the following icon on the tab that contains the missing values:  . You can use the policy properties area to enter the missing values.

The properties area includes tabs for properties that are not entered in the Add Policy wizard. The following tabs are included:

Preserve Options
: You can specify which values from the source data will not be masked.

Random
: You can provide a seed number from which the masked number is generated.

Hash Map
: You can enter the values to ignore when a hash value is generated.

Regular Expression
: You can edit the regular expressions that determine switch values.

Editing a data privacy policy

Use the properties area in the data privacy editor to edit a data privacy policy.

To edit a data privacy policy:

1. From the Data Project Explorer, expand a **Data Access Plans** folder in an Optim logical data model package.
2. Right-click a data access plan and click **Open**. The data access plan editor opens.
3. Click **Data Privacy**. The data privacy editor opens.

4. From the **Data privacy policies in use** area, select the data privacy policy you want to edit. The policy properties are displayed in the properties area.
5. Edit the policy.

   Select a tab to view or edit properties.
6. Click **File** > **Save**.

Chapter 6. Using Optim Designer with your Optim Solution

Optim Designer provides a single design interface for Optim data growth, data privacy, test data management, and application retirement solutions. Depending upon your processing and system requirements, the objects that you create and maintain from Optim Designer can be deployed to various components and repositories for processing.

Optim interoperability services allow you to deploy objects to process federated data across supported data sources and distributed platforms or to process DB2 data on a networkless z/OS platform. Native design interfaces on distributed and z/OS platforms allow you to refine the definitions and requests contained in interoperability models to meet your platform or data source requirements.

You can use Optim Designer to design and test Optim interoperability services and executor services. Both services allow you to use data privacy functions, while executor allows you to privatize data in place. You can also publish services to a centralized registry in the Optim Manager environment, where you schedule and run services for production processing.

Using Optim Designer with Optim interoperability services on a distributed platform

This tutorial teaches you how to use Optim Designer to create an Optim interoperability service based on an extract request. In this tutorial, you will use the Optim sample database to define data models. You will use the data models to define a data access plan that includes a selection policy, and for users with a data privacy license, a data privacy policy. You will create an Optim interoperability service based on a data model and a data access plan.

After completing this tutorial you will be able to create an extract request that you can run as an Optim interoperability service or import to an Optim directory.

Learning objectives

When you complete the exercises, you will know how to do the following tasks:

- Create a data design project to contain your data models and definitions
- Connect to the sample database
- Create a physical data model by reverse engineering a schema in the sample database
- Transform the physical data model to an Optim logical data model that can include a data access plan
- Create a data access plan and a selection policy
- Define selection criteria within the selection policy
- Define a data privacy policy to mask credit card information
- Create an Optim interoperability service that includes an extract request

Time required

This module should take approximately 60 minutes to complete.

Prerequisites

This tutorial uses the Optim sample database included with your Optim installation. Use the Optim Configuration program to install the database. For more information about the Optim Configuration program, refer to the *Optim Installation and Configuration Guide*.

This tutorial can be completed in the Optim Designer environment.

Creating a data design project

In this exercise, you will create a data design project in which to store your data models and definitions.

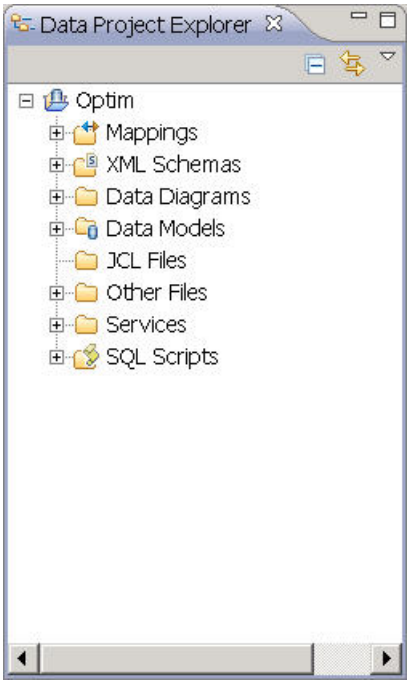
Before you create data models or other definitions, you must create a data design project in which to store your objects. You can store various types of objects in a data design project including data models, data management service definitions, and interoperability services.

To create a data design project:

1. From the main menu, click **File** > **New** > **Data Design Project**. The New Data Design Project wizard opens.
2. In the **Project Name** field, type `Optim`, then click **Finish**.

   If the Open Associated Perspective popup is displayed, click **No**. You will use the default Optim perspective.

   The Optim project is displayed in the Data Project Explorer.
3. Expand the Optim project in the Data Project Explorer to view the contents of the project.



Connecting to the Optim sample database

Optim Designer provides wizards that make it easy for you to connect to databases and to display the status of your connections. In this exercise, you will create a connection to the Optim sample database.

Note: This tutorial uses the Optim sample database included with your Optim installation. Use the Optim Configuration program to install the database. For more information about the Optim Configuration program, refer to the *Optim Installation and Configuration Guide*.

You can use the pages in the New Connection wizard to create a connection profile, allowing you to connect to a database.

You will use the sample database to define physical and logical data models upon which Optim processes are based.

To connect to the sample database:

1. In the Data Source Explorer view, right-click the **Database Connections** folder, and click **New**. The New Connection wizard opens.
2. On the Connection Parameters page, select a DBMS, a JDBC driver, and specify other connection details.
   a. In the **Connection identification** area, specify a connection name. The connection name is displayed in the Data Source Explorer after you create the connection.

      Use default naming convention
      Specifies that a connection name is generated based on the name of the database that you are connecting to.

      Connection name
      Type a name for the connection. Available only if **Use default naming convention** is not checked.

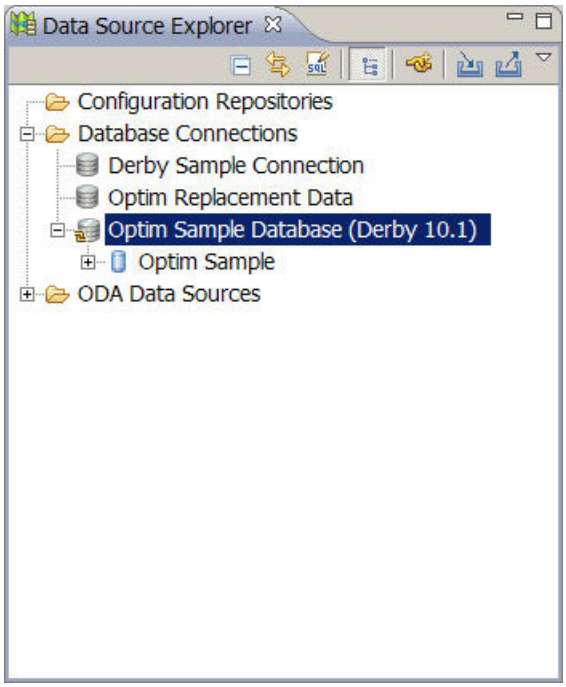
   b. In the **Local** tab, select the DBMS for your database.
   c. Select a driver in the **JDBC driver** list.

      If a driver that you want to use is not listed, but it is supported by the database manager, select **Other Driver Default** and provide the details.

      To modify the path to the JAR files for a JDBC driver, click the browse button (**...**) to open the Edit JAR List window. You can also use this window to view the names and typical locations for JDBC JAR files for each listed driver.

   d. In the **Properties** area, enter connection information for your database, based on the selected DBMS.
3. Click **Test Connection** to verify the connection.
4. Click **OK** to save the connection profile.
5. Right-click the database connection profile and click **Connect**.

The connection definition will display the database type, and open to display the database.



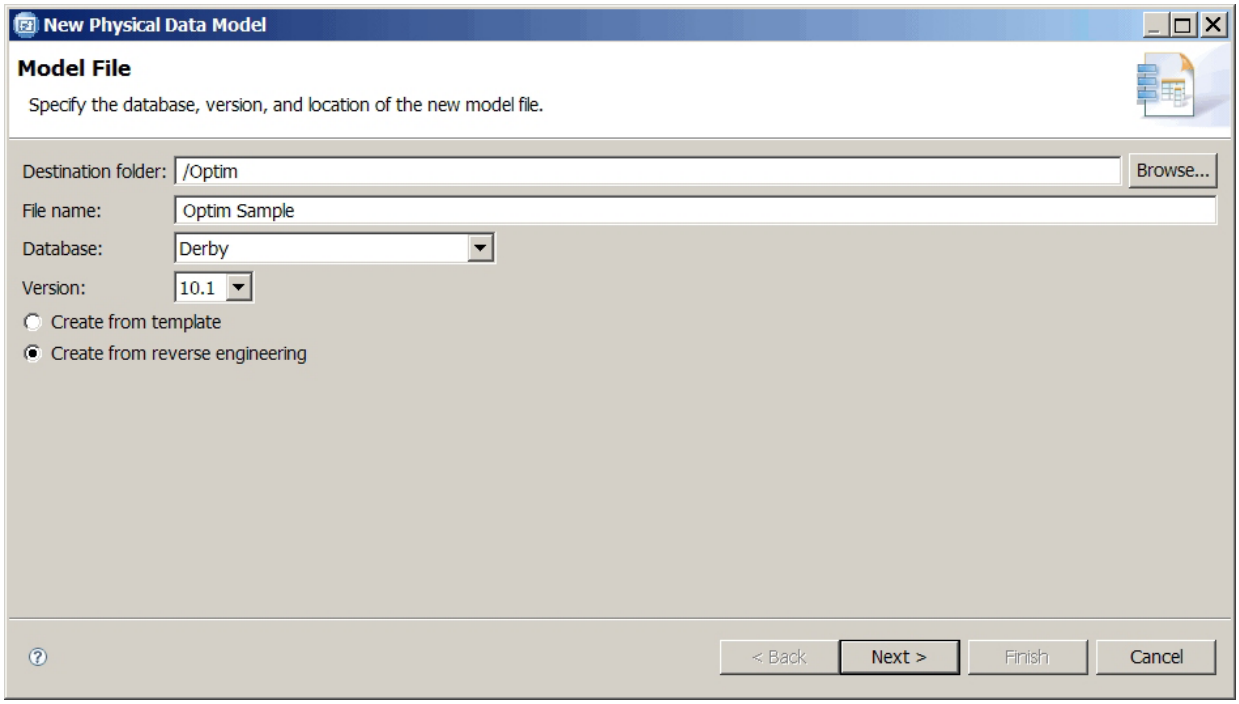
Creating a physical data model based on reverse engineering

In this exercise, you will create a physical data model for the Optim sample database. Physical data models are database-specific models that represent relational data objects (for example, tables, columns, primary keys, and foreign keys) and their relationships. A physical data model based on reverse engineering is created using the metadata in a source database.

Use physical data models to create Optim logical data models, which describe the data used with Optim data management services and processes.

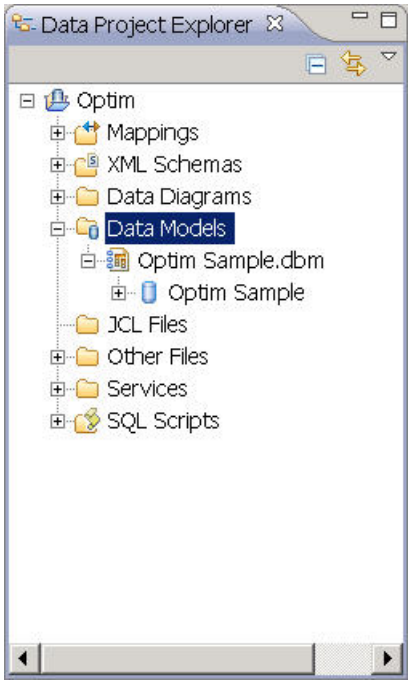
To create a physical data model based on reverse engineering:

1. On the Data Project Explorer view, right-click the **Data Models** folder and click **New** > **Physical Data Model**. The New Physical Data Model wizard opens.
2. In the Model File page, do the following:
   a. In **File Name**, type `Optim Sample`.
   b. From the **Database** list, select the DBMS that contains the Optim sample database.
   c. From the **Version** list, select the version of the DBMS.
   d. Select **Create from reverse engineering**.
   e. Click **Next**.



3. On the Source page, select **Database**.
4. On the Select Connection page, from the **Connections** area, select the connection for the Optim sample database. Click **Next**.
5. On the Select Objects page, from the **Select objects** area, select the schema name for the Optim sample database. Click **Finish**.

The new physical data model, Optim Sample.dbm, will appear under the **Data Models** folder.

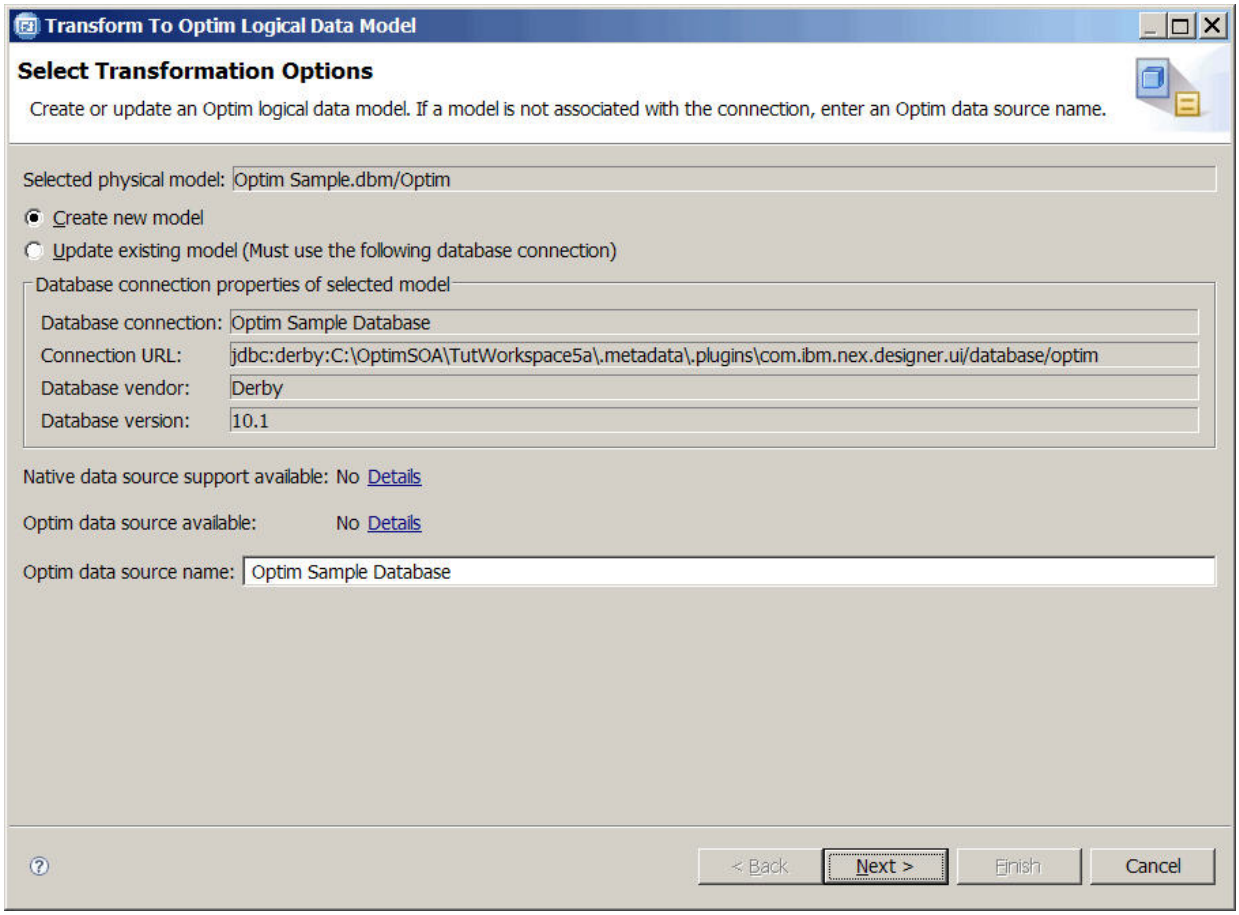


Transforming a schema in a physical data model to an Optim logical data model

In this exercise, you will create an Optim logical data model from a schema in a physical data model. Logical data models are not specific to a database and describe the data used with Optim data management services. An Optim logical data model is a logical data model that includes a data access plan, which contains policies that determine the data to copy or transform from a source logical data model used in an Optim data management service.

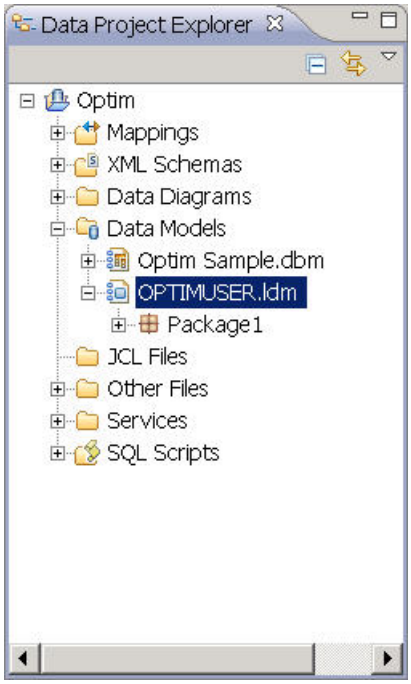
To transform a schema in a physical data model to an Optim logical data model:

1. In the Data Project Explorer, expand the **Data Models** folder, and expand the Optim Sample physical data model to display the schema for the Optim sample database.
2. Right-click the schema, and click **Transform to Optim Logical Data Model**. The Transform to Optim Logical Data Model wizard opens.
3. On the Select Transformation Options page, select **Create new model** and use the default value for the **Optim data source name**, *Optim Sample Database*. Click **Next**.



4. On the Native Data Source Access page, clear the **Use native data source connection as the default for services** check box. A native data source connection is not required for this tutorial. Click **Next**.
5. On the Enter Model Name and Project Folder page, type `OPTIMUSER` in **Name**. Click **Next**.
6. On the Transformation Results page, review the results of the transformation, and click **Finish**. The new logical data model, OPTIMUSER.ldm, will appear under the **Data Models** folder.

You have created a new Optim logical data model, OPTIMUSER.

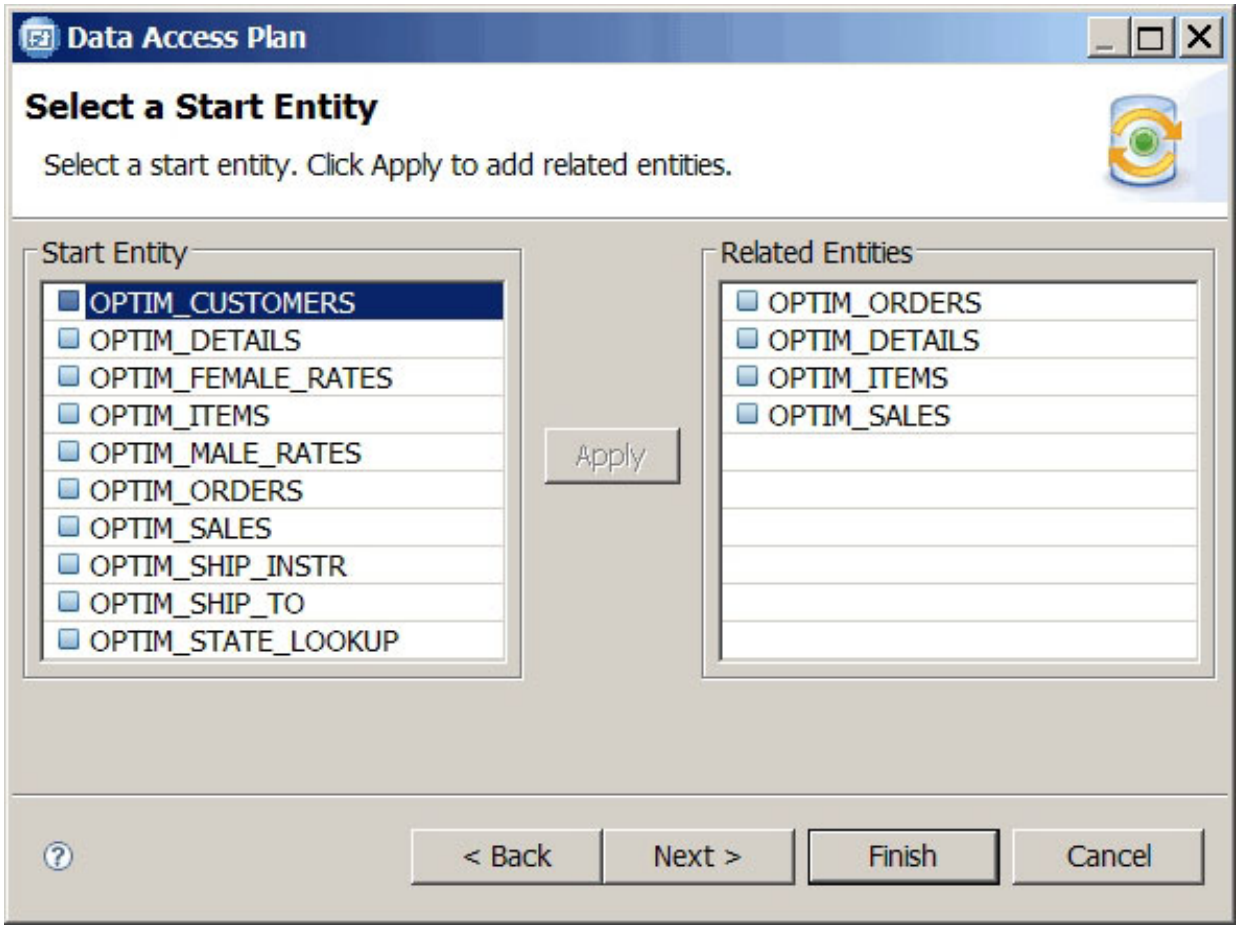


Creating a data access plan and a selection policy

In this exercise, you will create a data access plan and a selection policy. A data access plan contains policies that determine which data to copy or transform from a source logical data model in an Optim data management service or process. A selection policy specifies the entities and attributes to use in an Optim data management service or process.

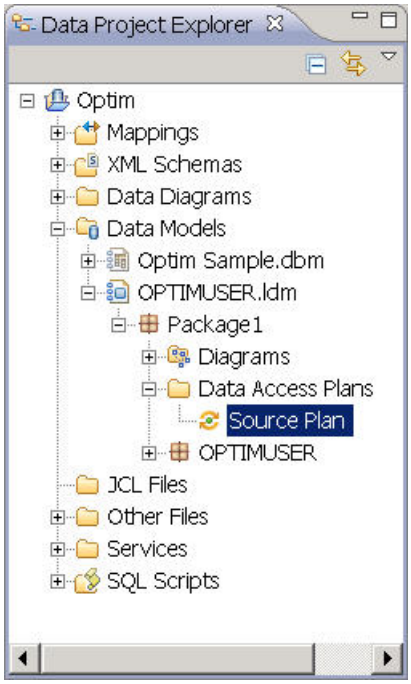
To create a data access plan and a selection policy:

1. In the Data Project Explorer, expand the **Data Models** folder, expand the OPTIMUSER logical data model to open the model, and expand the Package1 node to display the **Data Access Plans** folder.
2. Right-click the **Data Access Plans** folder and click **New** > **Data Access Plan**. The Data Access Plan wizard opens.
3. On the Data Access Plan Name page, type `Source Plan` in the **Name** field. Click **Next**.
4. On the Select a Package page, select the package with the schema name of the Optim sample database. Click **Next**.
5. On the Select Entity Options page, select **Select entities based on relationships with a start entity**. Click **Next**.
6. On the Select a Start Entity page, select **OPTIM_CUSTOMERS** from the **Start Entity** area, and click **Apply** to add the related tables to the **Related Entities** area. Click **Next**.



7. On the Select Reference Entities page, click **Finish**.
8. From the main menu, click **File** > **Save All**.

You have created a data access plan, Source Plan, which contains a selection policy that specifies OPTIM_CUSTOMERS as the start table and includes related tables in the OPTIMUSER schema.

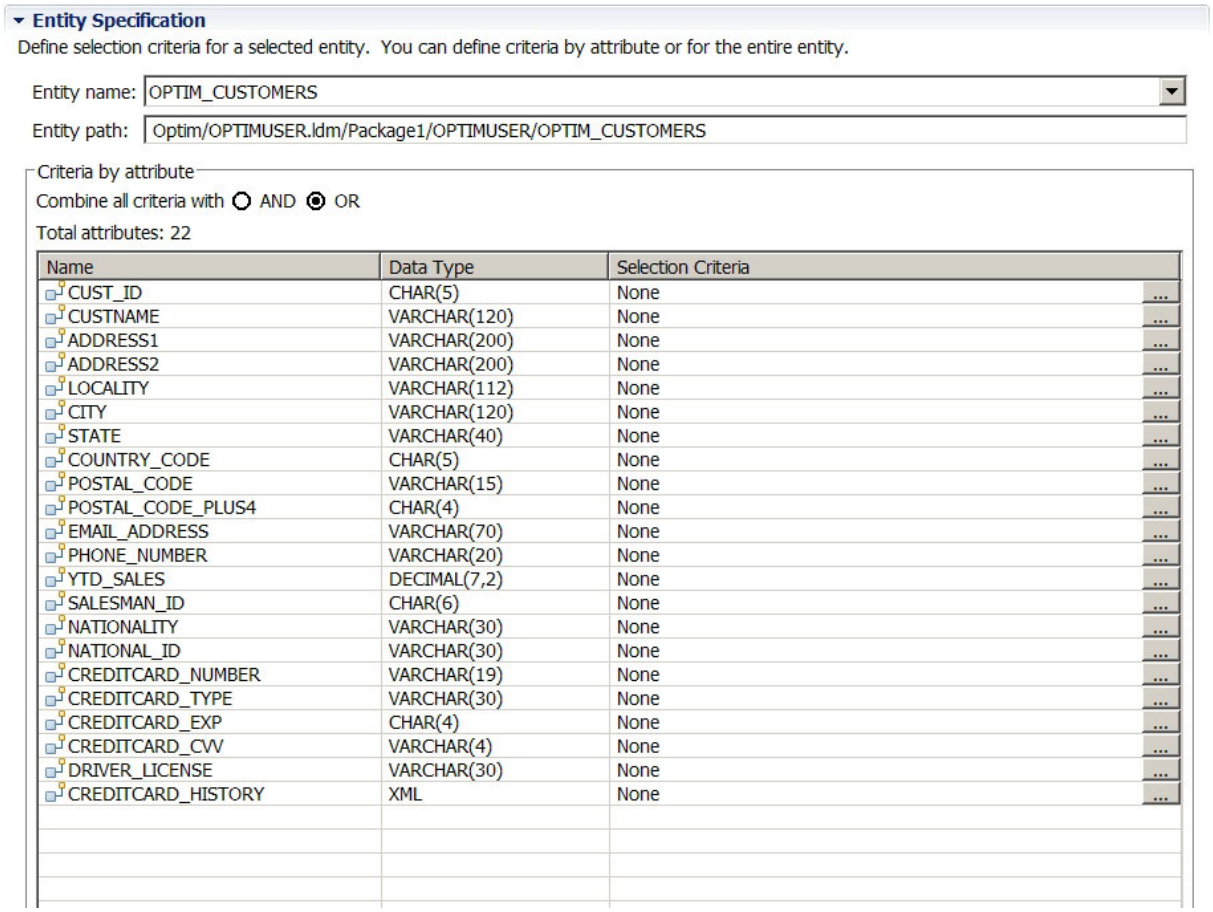


Defining selection criteria

In this exercise, you will define selection criteria for the selection policy in the Source Plan data access plan. Selection criteria allow you to pinpoint the data you want to use in an Optim data management service or process. You can select data according to values in one or more columns. Selection criteria must conform to SQL syntax and include relational or logical operators.

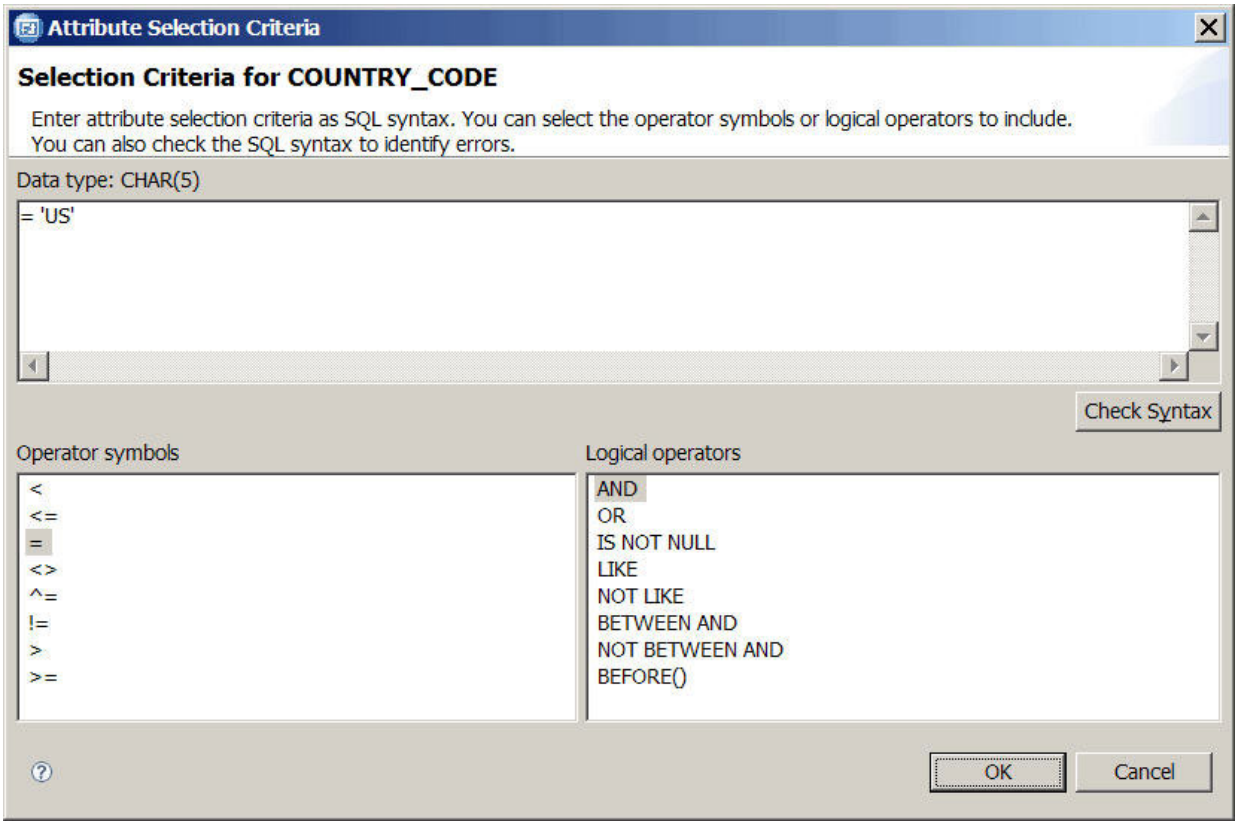
To define selection criteria:

1. In the Data Project Explorer, expand the **Data Models** folder, double-click the OPTIMUSER logical data model to open the model, expand the Package1 node, and expand the **Data Access Plans** folder.
2. Double-click **Source Plan**. The data access plan editor opens.
3. Click **Selection**. The selection policy editor opens.
4. In the **Entity Specification** area, select OPTIM_CUSTOMERS from the **Entity name** list.



The attributes for the OPTIM_CUSTOMERS entity are listed in the **Criteria by attribute** area.

5. Click the browse button in the **Selection Criteria** cell for the COUNTRY_CODE attribute. The Attribute Selection Criteria window opens.
6. Do the following in the Attribute Selection Criteria window:
   a. From the **Operator symbols** list, double-click **=**.
   b. In the editor area, type `'US'`. The following syntax should be entered: `='US'`.
   c. Click **OK** to return to the Selection Policy editor.



7. From the main menu, click **File** > **Save**.

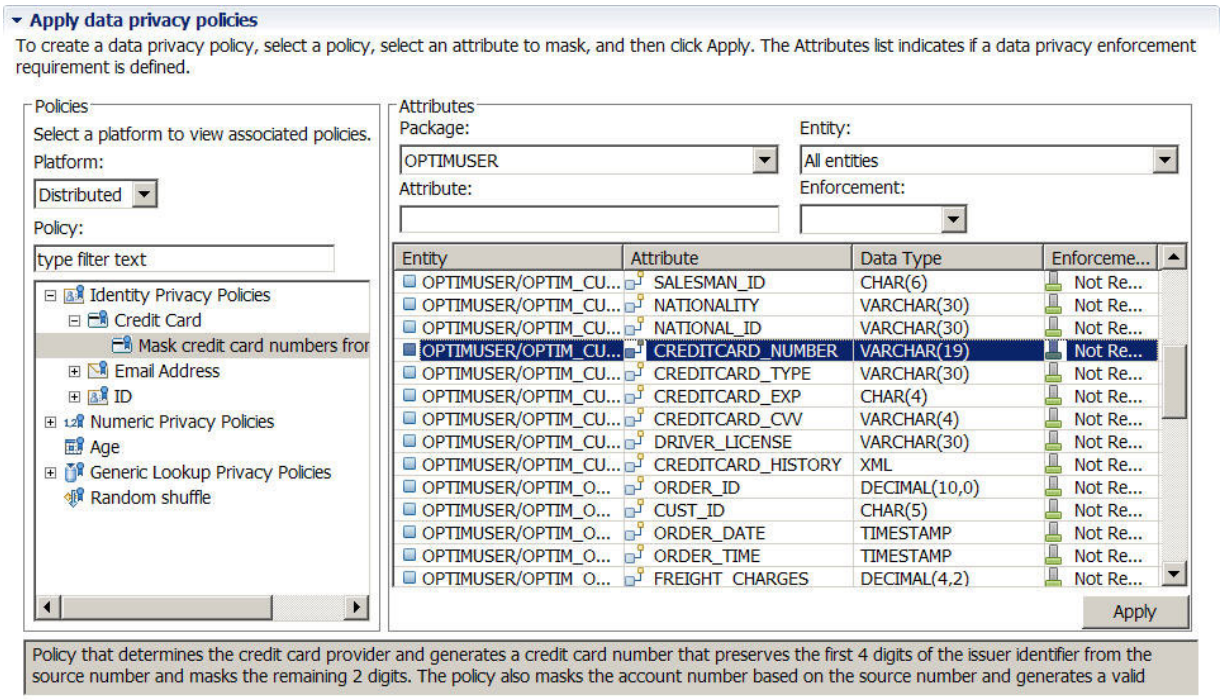
You have defined selection criteria that will only select rows from the OPTIM_CUSTOMERS entity in which the value of the COUNTRY_CODE attribute is 'US'.

Defining a data privacy policy to mask credit card numbers

This is an optional exercise intended for Optim Solution users with a data privacy license. In this exercise, you will define a data privacy policy to mask credit card numbers. The policy will be added to the Source Plan data access plan.

To define a data privacy policy to mask credit card numbers:

1. In the Data Project Explorer, expand the **Data Models** folder, double-click the OPTIMUSER logical data model to open the model, expand the Package1 node, and expand the **Data Access Plans** folder.
2. Double-click **Source Plan**. The data access plan editor opens.
3. Click **Data Privacy**. The data privacy editor opens.
4. From the **Policies** area, do the following:
   a. Select **Distributed** from the **Platform** list.
   b. Expand **Identity Privacy Policies**, expand **Credit Card**, and select **Mask credit card numbers from all providers**.
5. In the **Attributes** area, select **OPTIM_CUSTOMERS** from the **Entity** list. The attributes in the OPTIM_CUSTOMERS entity are listed.
6. From the **Attributes** list, select **CREDITCARD_NUMBER**.



7. Click **Apply**. The new privacy policy, OPTIM_CUSTOMERS, will display in the **Data privacy policies in use** area.
8. From the **Data privacy policies in use** area, select OPTIM_CUSTOMERS. The properties for the policy will display below the **Data privacy policies in use** area.
9. In the properties area, select the **Credit Card Policy Option** tab.
10. Select **Mask credit card issuer?**.
11. From the main menu, click **File** > **Save**.

You have defined a privacy policy that will mask credit card numbers from all supported issuers in the CREDITCARD_NUMBER attribute of the OPTIM_CUSTOMERS entity.

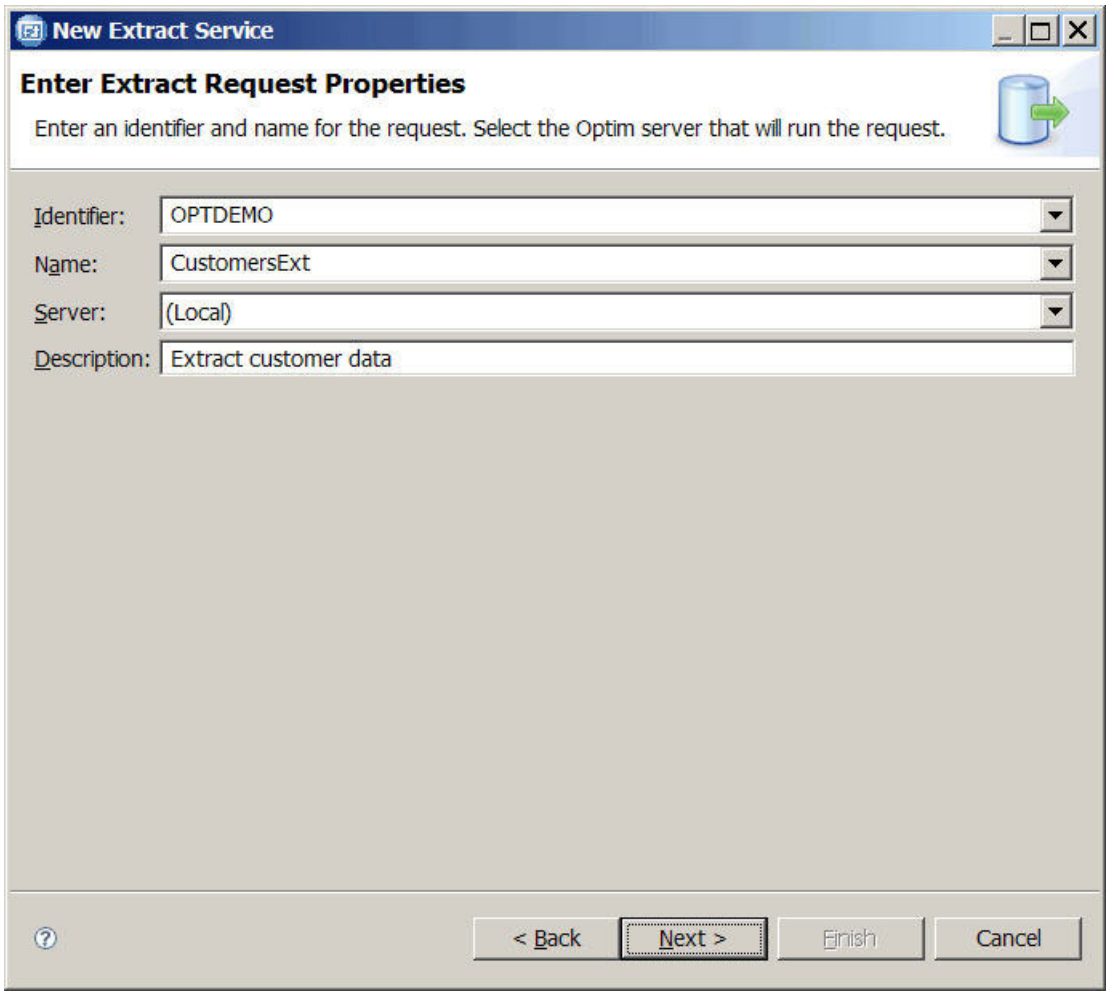
Creating an Optim interoperability service

In this exercise, you will create an Optim interoperability service that includes an extract request. The service allows you to execute an Optim request from the Optim Manager environment.

To create an Optim interoperability service:

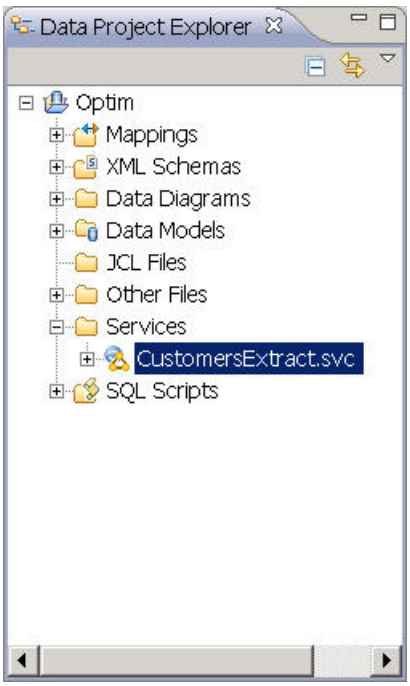
1. In the Data Project Explorer view, right-click the **Services** folder and click **New** > **Distributed Service** > **Distributed Extract**. The New Extract Service wizard opens.
2. On the Enter Extract Service Properties page, type `CustomersExtract` in **Extract service name**. Click **Next**.
3. On the Select an Optim Data Source page, select the data source that contains your Optim sample data. Click **Next**.
4. On the Select a Logical Data Model page, select **OPTIMUSER.ldm**. Click **Next**.
5. On the Select a Data Access Plan page, select **Source Plan**. Click **Next**.
6. On the Enter Extract Request Properties page, do the following:
   a. In **Identifier**, type `OPTDEMO`.
   b. In **Name**, type `CustomersExt`.
   c. In **Server**, use **(Local)**.
   d. In **Description**, type `Extract customer data`.

e. Click **Next**.



7. On the Enter Access Definition Properties page, type the name of the DB alias for your data source in **DB Alias**, and enter the ID for the data source in **Creator ID**. Click **Next**.
8. On the Enter Extract Process Properties and Options page, type `CustomersExtract.xf` in **Extract file name** and accept the default options. Click **Next**.
9. On the Select Objects to Extract page, accept the defaults. Click **Next**.
10. On the Enter Group Selection Options page, accept the defaults. Click **Finish**.

The new Optim interoperability service, CustomersExtract, will appear under the **Services** folder. The service will extract data defined in the source OPTIMUSER logical data model and store it in the CustomersExtract.xf extract file. The service will use the Source Plan data access plan to determine which data to select from the OPTIMUSER logical data model.



Using Optim Designer with Optim interoperability services on a z/OS platform

This tutorial teaches you how to use Optim Designer to create an Optim interoperability services that includes an extract request. In this tutorial, you will use the Optim sample database to define data models. You will use the data models to define a data access plan that includes a selection policy, and for users with a data privacy license, a data privacy policy. You will create an Optim interoperability service based on a data model and a data access plan.

After completing this tutorial you will be able to create a process request that you can include in an Optim interoperability service.

Learning objectives

When you complete the exercises, you will know how to do the following tasks:

- Create a data design project to contain your data models and definitions
- Connect to the sample database
- Create a physical data model by reverse engineering a schema in the sample database
- Transform the physical data model to a logical data model that can include a data access plan
- Create a data access plan and a selection policy
- Define selection criteria within the selection policy
- Define a data privacy policy to mask credit card information
- Create an Optim interoperability service that includes an extract request

Time required

This module should take approximately 60 minutes to complete.

Prerequisites

This tutorial uses the Optim sample database included with your Optim installation. For more information about installing the Optim sample database, refer to the *IBM InfoSphere Optim for DB2 for z/OS Customization Guide*.

This tutorial can be completed in the Optim Designer environment.

Creating a data design project

In this exercise, you will create a data design project in which to store your data models and definitions.

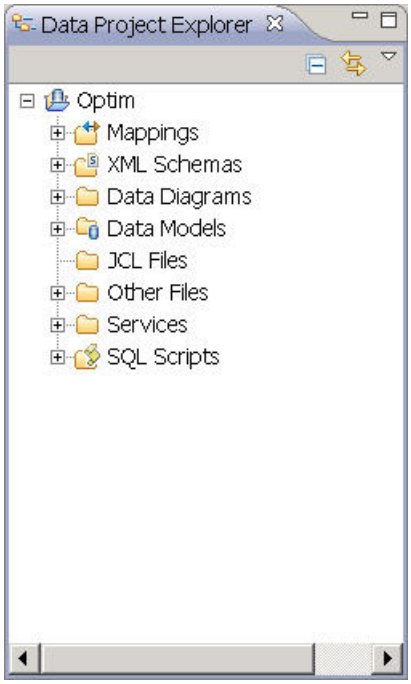
Before you create data models or other definitions, you must create a data design project in which to store your objects. You can store various types of objects in a data design project including data models, data management service definitions, and interoperability services.

To create a data design project:

1. From the main menu, click **File** > **New** > **Data Design Project**. The New Data Design Project wizard opens.
2. In the **Project Name** field, type `Optim`, then click **Finish**.

   If the Open Associated Perspective popup is displayed, click **No**. You will use the default Optim perspective.

   The Optim project is displayed in the Data Project Explorer.
3. Expand the Optim project in the Data Project Explorer to view the contents of the project.



Connecting to the Optim sample database

Optim Designer provides wizards that make it easy for you to connect to databases and to display the status of your connections. In this exercise, you will create a connection to the Optim sample database.

Note: This tutorial uses the Optim sample database included with your Optim installation. For more information about installing the Optim sample database, refer to the *IBM InfoSphere Optim for DB2 for z/OS Customization Guide*.

You can use the pages in the New Connection wizard to create a connection profile, so that you can connect to a database.

You will use the sample database to define physical and logical data models upon which Optim processes are based.

To connect to the sample database:

1. In the Data Source Explorer view, right-click the **Database Connections** folder, and click **New Connection**. The New Connection wizard opens.
2. On the Connection Parameters page, select a DBMS, a JDBC driver, and specify other connection details.
   a. In the **Connection identification** area, specify a connection name. The connection name is displayed in the Data Source Explorer after you create the connection.

      Use default naming convention
      Specifies that a connection name is generated based on the name of the database that you are connecting to.

      Connection name
      Type a name for the connection. Available only if **Use default naming convention** is not checked.

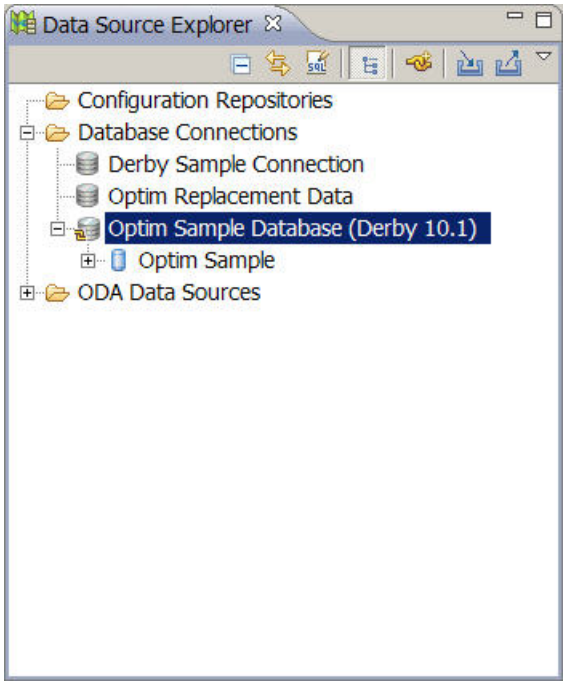
   b. In the **Local** tab, select the DBMS for your database.
   c. Select a driver in the **JDBC driver** list.

      If a driver that you want to use is not listed, but it is supported by the database manager, select **Other Driver Default** and provide the details.

      To modify the path to the JAR files for a JDBC driver, click the browse button (**...**) to open the Edit JAR List window. You can also use this window to view the names and typical locations for JDBC JAR files for each listed driver.

   d. In the **Properties** area, enter connection information for your database, based on the selected DBMS.
3. Click **Test Connection** to verify the connection.
4. Click **OK** to save the connection profile.
5. Right-click the database connection profile and click **Connect**.

The connection definition will display the database type, and open to display the database.



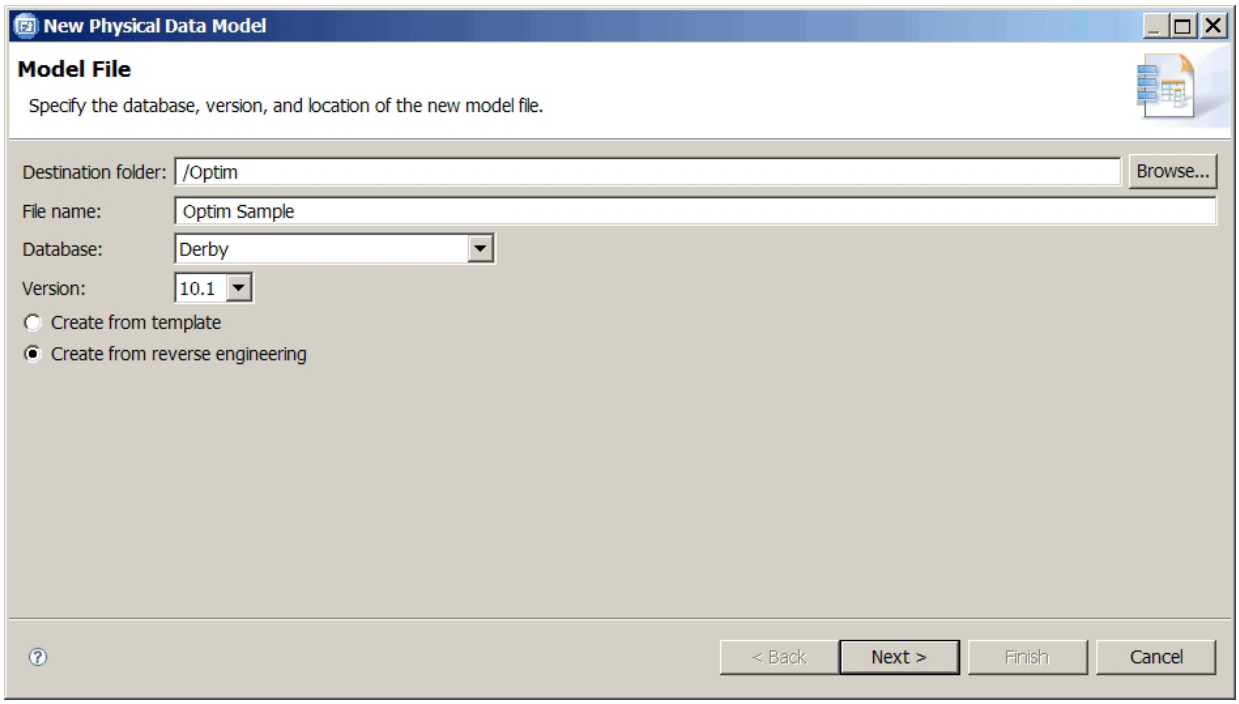
Creating a physical data model based on reverse engineering

In this exercise, you will create a physical data model for the Optim sample database. Physical data models are database-specific models that represent relational data objects (for example, tables, columns, primary keys, and foreign keys) and their relationships. A physical data model based on reverse engineering is created using the metadata in a source database.

Use physical data models to create Optim logical data models, which describe the data used with Optim data management services and processes.

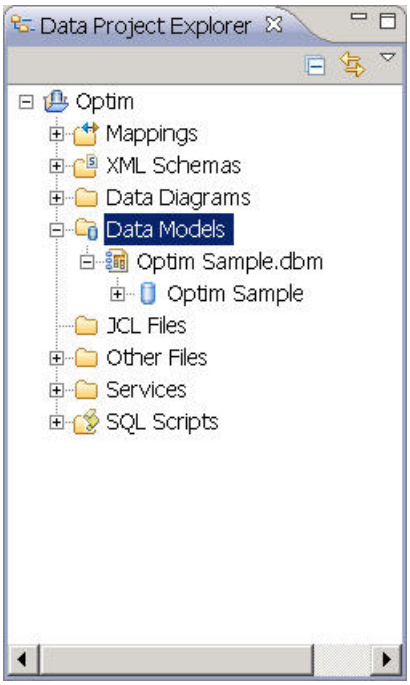
To create a physical data model based on reverse engineering:

1. On the Data Project Explorer view, right-click the **Data Models** folder and click **New** > **Physical Data Model**. The New Physical Data Model wizard opens.
2. In the Model File page, do the following:
   a. In **File Name**, type `Optim Sample`.
   b. From the **Database** list, select the DBMS that contains the Optim sample database.
   c. From the **Version** list, select the version of the DBMS.
   d. Select **Create from reverse engineering**.
   e. Click **Next**.



3. On the Source page, select **Database**.
4. On the Select Connection page, from the **Connections** area, select the connection for the Optim sample database. Click **Next**.
5. On the Select Objects page, from the **Select objects** area, select the schema name for the Optim sample database. Click **Finish**.

The new physical data model, Optim Sample.dbm, will appear under the **Data Models** folder.

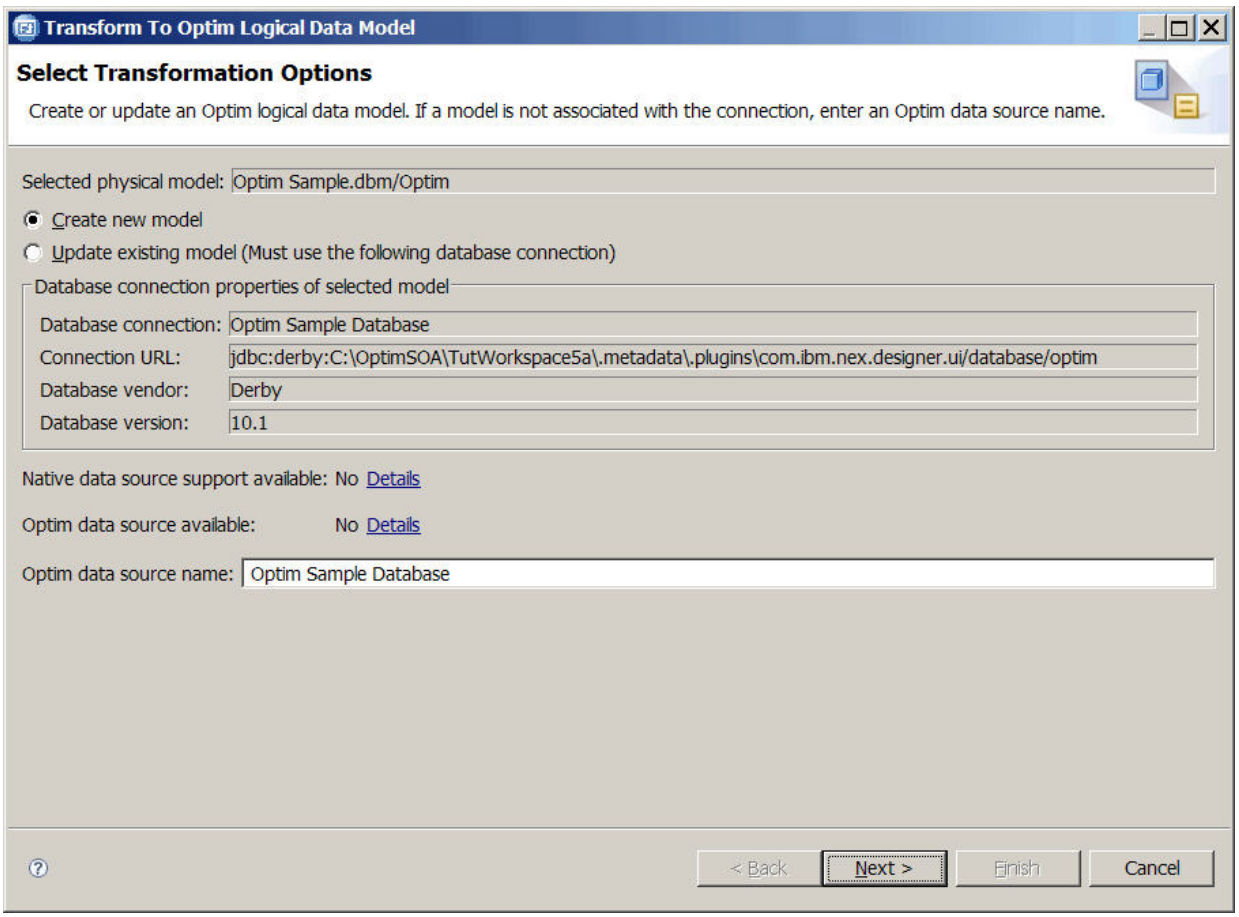


Transforming a schema in a physical data model to an Optim logical data model

In this exercise, you will create an Optim logical data model from a schema in a physical data model. Logical data models are not specific to a database and describe the data used with Optim data management services. An Optim logical data model is a logical data model that includes a data access plan, which contains policies that determine the data to copy or transform from a source logical data model used in an Optim data management service.

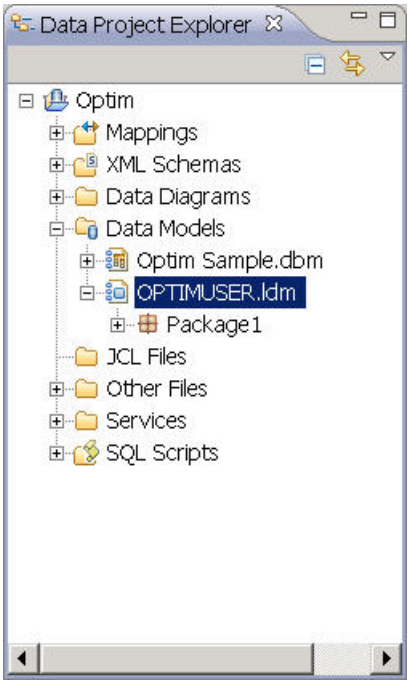
To transform a schema in a physical data model to an Optim logical data model:

1. In the Data Project Explorer, expand the **Data Models** folder, and expand the Optim Sample physical data model to display the schema for the Optim sample database.
2. Right-click the schema, and click **Transform to Optim Logical Data Model**. The Transform to Optim Logical Data Model wizard opens.
3. On the Select Transformation Options page, select **Create new model** and use the default value for the **Optim data source name**, *Optim Sample Database*. Click **Next**.



4. On the Native Data Source Access page, clear the **Use native data source connection as the default for services** check box. A native data source connection is not required for this tutorial. Click **Next**.
5. On the Enter Model Name and Project Folder page, type `OPTIMUSER` in **Name**. Click **Next**.
6. On the Transformation Results page, review the results of the transformation, and click **Finish**. The new logical data model, OPTIMUSER.ldm, will appear under the **Data Models** folder.

You have created a new Optim logical data model, OPTIMUSER.

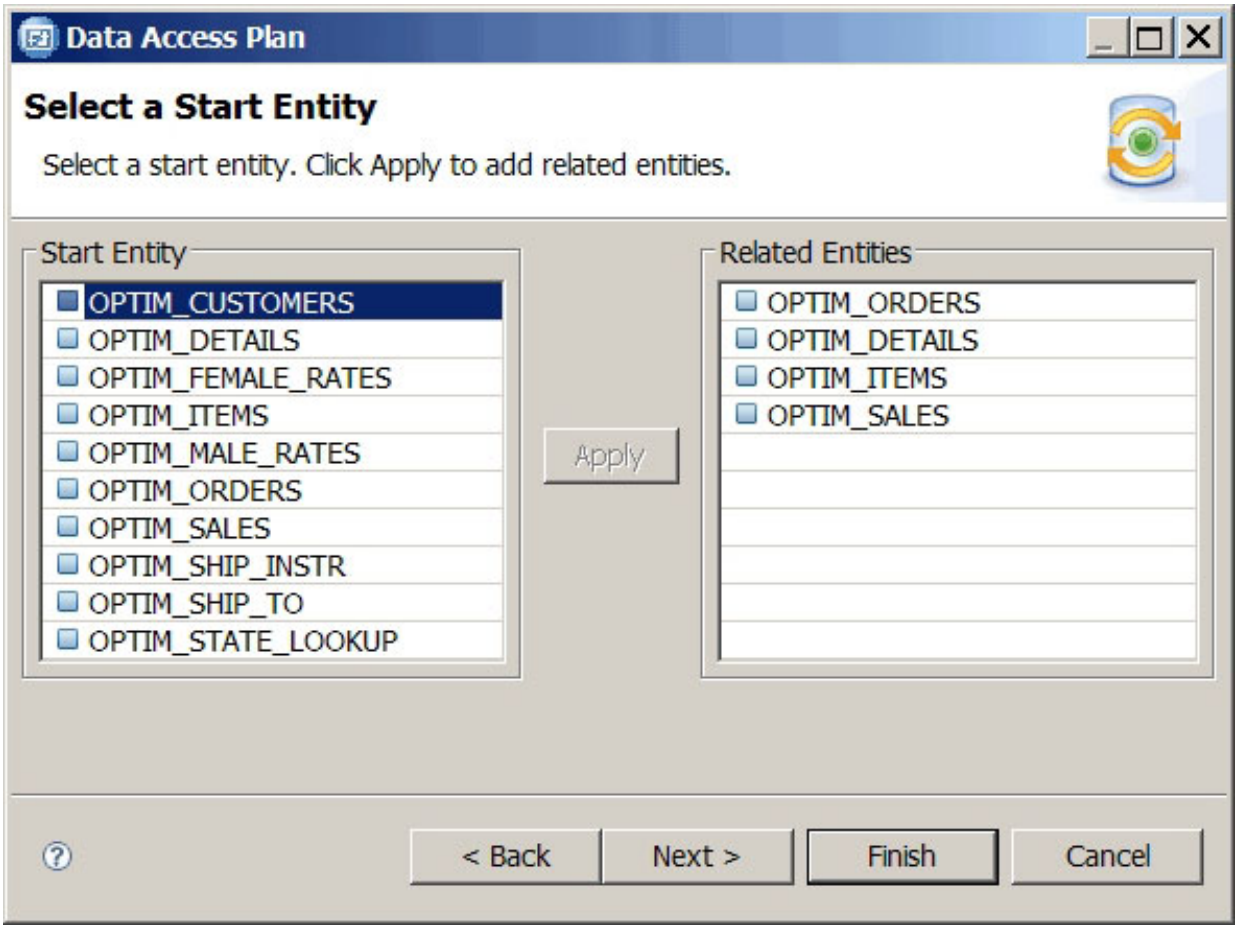


Creating a data access plan and a selection policy

In this exercise, you will create a data access plan and a selection policy. A data access plan contains policies that determine which data to copy or transform from a source logical data model in an Optim data management service or process. A selection policy specifies the entities and attributes to use in an Optim data management service or process.

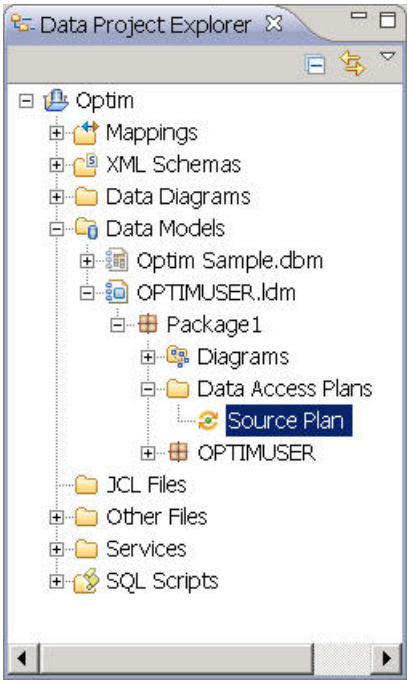
To create a data access plan and a selection policy:

1. In the Data Project Explorer, expand the **Data Models** folder, expand the OPTIMUSER logical data model to open the model, and expand the Package1 node to display the **Data Access Plans** folder.
2. Right-click the **Data Access Plans** folder and click **New** > **Data Access Plan**. The Data Access Plan wizard opens.
3. On the Data Access Plan Name page, type `Source Plan` in the **Name** field. Click **Next**.
4. On the Select a Package page, select the package with the schema name of the Optim sample database. Click **Next**.
5. On the Select Entity Options page, select **Select entities based on relationships with a start entity**. Click **Next**.
6. On the Select a Start Entity page, select **OPTIM_CUSTOMERS** from the **Start Entity** area, and click **Apply** to add the related tables to the **Related Entities** area. Click **Next**.



7. On the Select Reference Entities page, click **Finish**.
8. From the main menu, click **File** > **Save All**.

You have created a data access plan, Source Plan, which contains a selection policy that specifies OPTIM_CUSTOMERS as the start table and includes related tables in the OPTIMUSER schema.

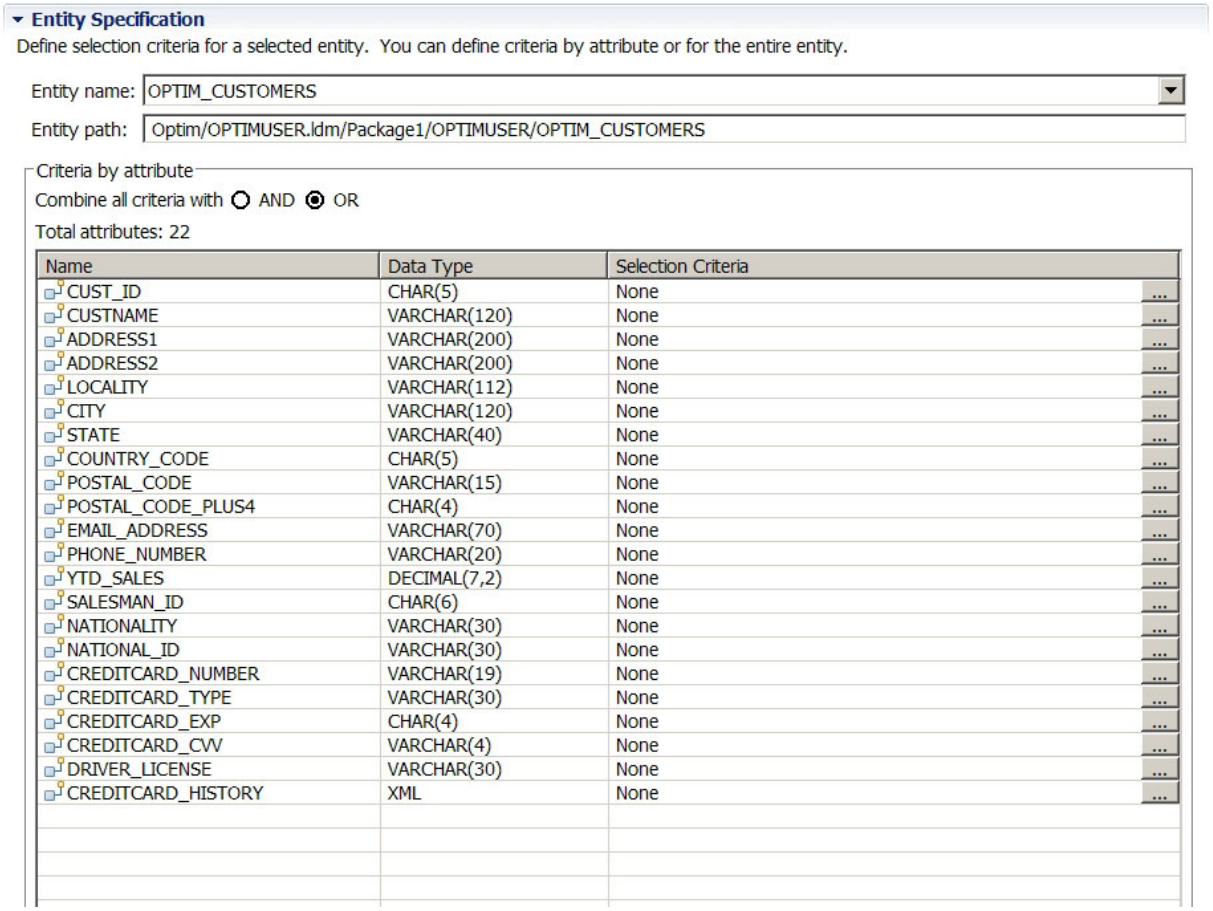


Defining selection criteria

In this exercise, you will define selection criteria for the selection policy in the Source Plan data access plan. Selection criteria allow you to pinpoint the data you want to use in an Optim data management service or process. You can select data according to values in one or more columns. Selection criteria must conform to SQL syntax and include relational or logical operators.

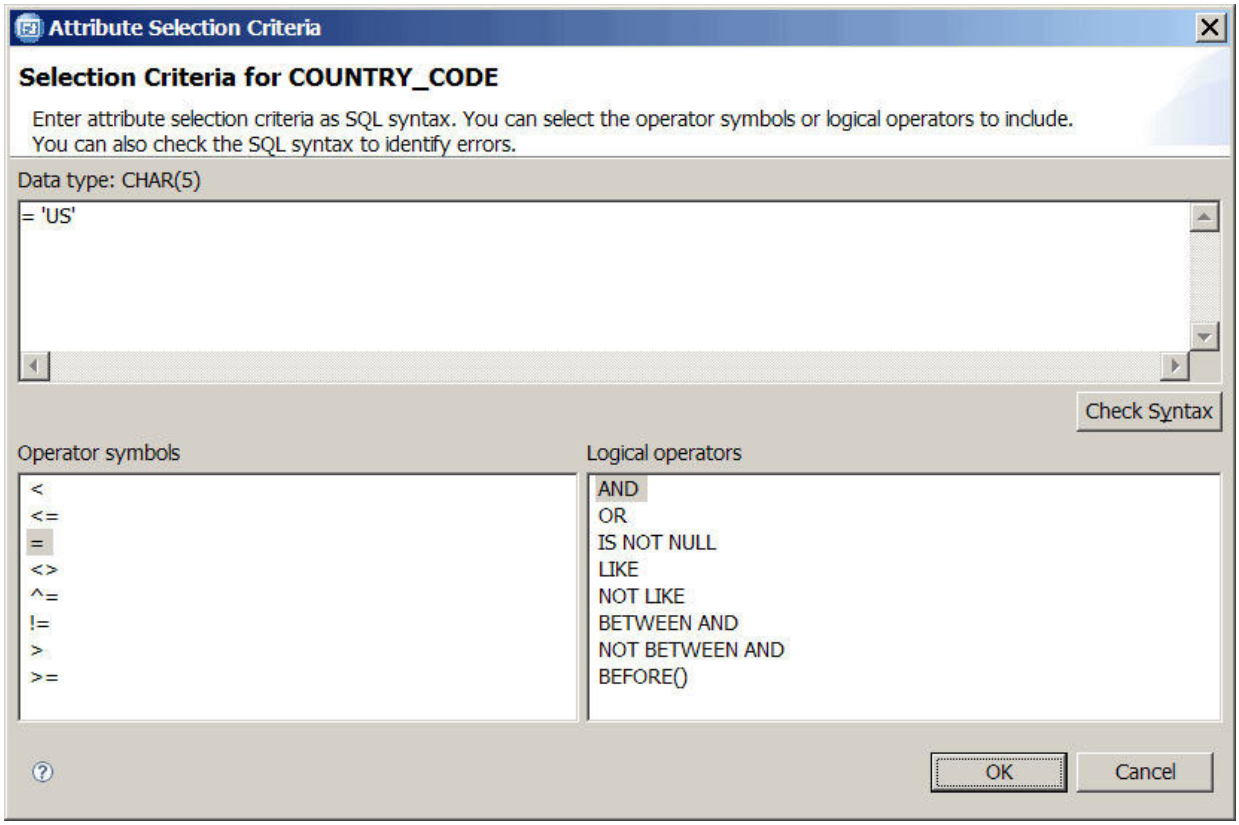
To define selection criteria:

1. In the Data Project Explorer, expand the **Data Models** folder, double-click the OPTIMUSER logical data model to open the model, expand the Package1 node, and expand the **Data Access Plans** folder.
2. Double-click **Source Plan**. The data access plan editor opens.
3. Click **Selection**. The selection policy editor opens.
4. In the **Entity Specification** area, select OPTIM_CUSTOMERS from the **Entity name** list.



The attributes for the OPTIM_CUSTOMERS entity are listed in the **Criteria by attribute** area.

5. Click the browse button in the **Selection Criteria** cell for the COUNTRY_CODE attribute. The Attribute Selection Criteria window opens.
6. Do the following in the Attribute Selection Criteria window:
   a. From the **Operator symbols** list, double-click **=**.
   b. In the editor area, type `'US'`. The following syntax should be entered: `='US'`.
   c. Click **OK** to return to the Selection Policy editor.



7. From the main menu, click **File** > **Save**.

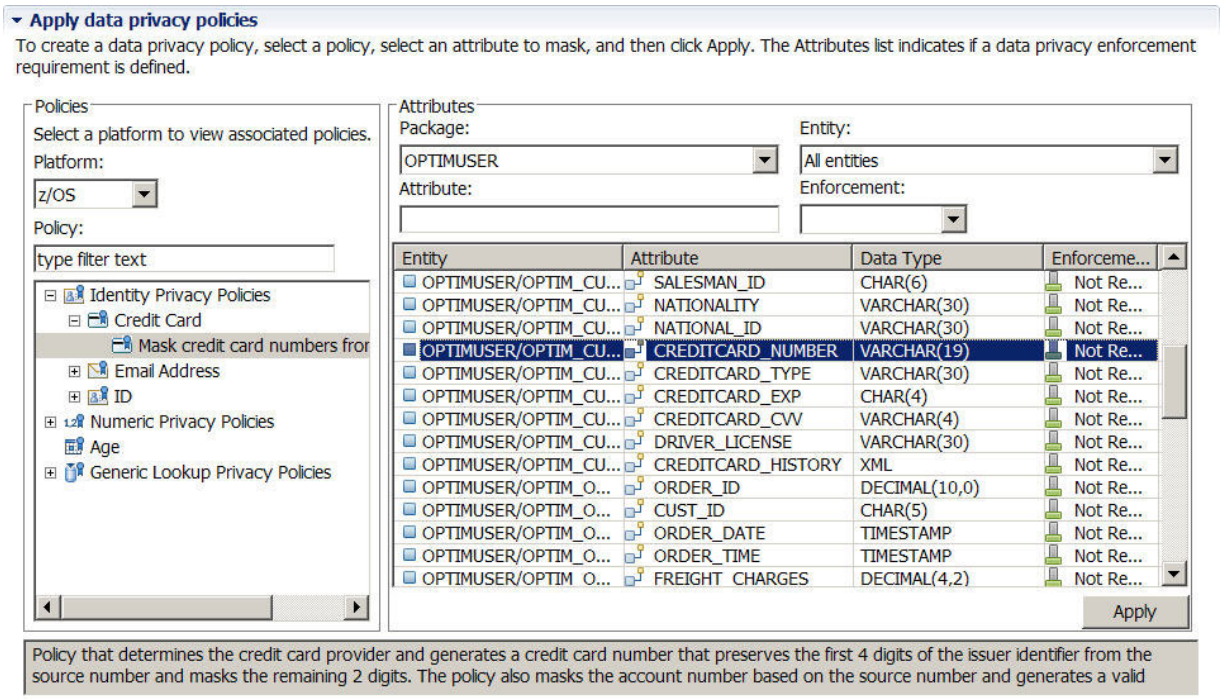
You have defined selection criteria that will only select rows from the OPTIM_CUSTOMERS entity in which the value of the COUNTRY_CODE attribute is 'US'.

Defining a data privacy policy to mask credit card numbers

This is an optional exercise intended for Optim Solution for z/OS users with a data privacy license. In this exercise, you will define a data privacy policy to mask credit card numbers. The policy will be added to the Source Plan data access plan.

To define a data privacy policy to mask credit card numbers:

1. In the Data Project Explorer, expand the **Data Models** folder, double-click the OPTIMUSER logical data model to open the model, expand the Package1 node, and expand the **Data Access Plans** folder.
2. Double-click **Source Plan**. The data access plan editor opens.
3. Click **Data Privacy**. The data privacy editor opens.
4. From the **Policies** area, do the following:
   a. Select **z/OS** from the **Platform** list.
   b. Expand **Identity Privacy Policies**, expand **Credit Card**, and select **Mask credit card numbers from all providers**.
5. In the **Attributes** area, select **OPTIM_CUSTOMERS** from the **Entity** list. The attributes in the OPTIM_CUSTOMERS entity are listed.
6. From the **Attributes** list, select **CREDITCARD_NUMBER**.



7. Click **Apply**. The new privacy policy, OPTIM_CUSTOMERS, will display in the **Data privacy policies in use** area.
8. From the **Data privacy policies in use** area, select OPTIM_CUSTOMERS. The properties for the policy will display below the **Data privacy policies in use** area.
9. In the properties area, select the **Credit Card Policy Option** tab.
10. Select **Mask credit card issuer?**.
11. From the main menu, click **File** > **Save**.

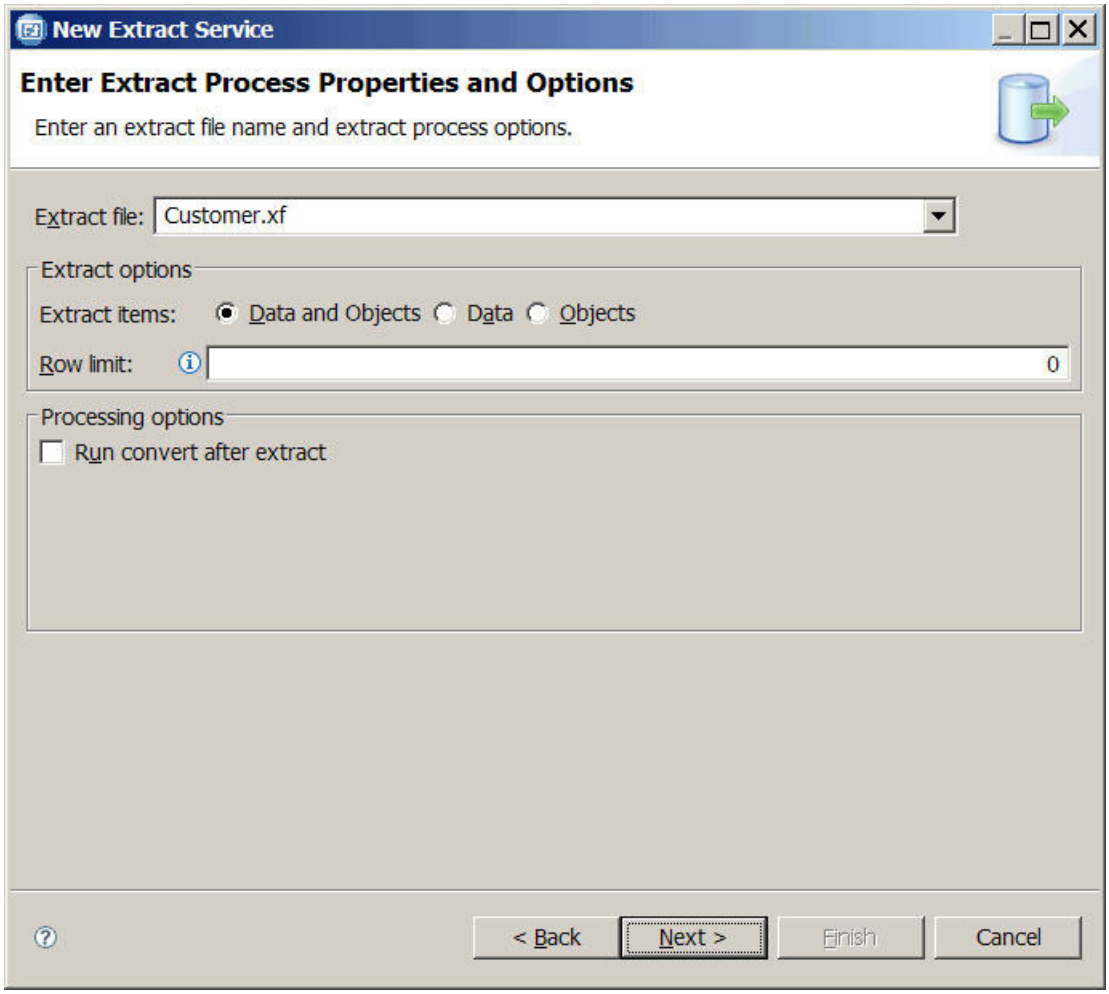
You have defined a privacy policy that will mask credit card numbers from all supported issuers in the CREDITCARD_NUMBER attribute of the OPTIM_CUSTOMERS entity.

Creating an Optim interoperability service

In this exercise, you will create an Optim interoperability service from a request in an Optim interoperability model. An Optim interoperability service is based on the data source and processing information in an Optim interoperability request. The service allows you to execute an Optim interoperability request from the Optim Manager environment.

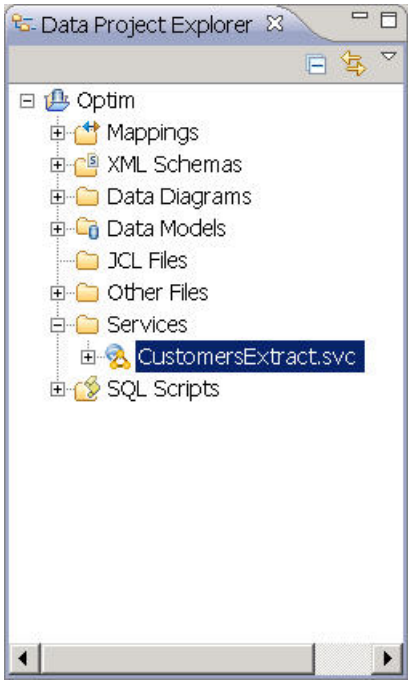
To create an Optim interoperability service:

1. In the Data Project Explorer view, right-click the **Services** folder and click **New** > **z/OS Service** > **z/OS Extract**. The New Extract Service wizard opens.
2. On the Enter Extract Service Properties page, type `CustExt` in **Extract service name**. Click **Next**.
3. On the Select an Optim Data Source page, select the data source that contains your Optim sample data. Click **Next**.
4. On the Select a Logical Data Model page, select **OPTIMUSER.ldm**. Click **Next**.
5. On the Select a Data Access Plan page, select **Source Plan**. Click **Next**.
6. On the Enter Access Definition Properties page, type `Optim.User.Customers` in **Access definition**, and enter the ID for the data source in **Creator ID**. Click **Next**.
7. On the Enter Extract Properties and Options page, type `Customer.xf` in **Extract file** and accept the default options. Click **Next**.



8. On the Select Objects to Extract page, accept the defaults. Click **Next**.
9. On the Enter Group Selection Options page, accept the defaults. Click **Finish**.

The new Optim interoperability service, CustomersExtract, will appear under the **Services** folder. The service will extract data defined in the source OPTIMUSER logical data model and store it in the Customer.xf extract file. The service will use the Source Plan data access plan to determine which data to select from the OPTIMUSER logical data model.



Using Optim Designer with Optim Data Masking Solution

This tutorial teaches you how to use Optim Designer to create an Optim data management service that will copy data defined in one logical data model to another logical data model. In this tutorial, you will use the Optim sample database to define data models. You will use the data models to define a data access plan that includes polices for data selection and data privacy.

Learning objectives

When you complete the exercises, you will know how to do the following tasks:

- Create a data design project to contain your data models and definitions
- Connect to the sample database
- Create physical data models by reverse engineering schemas in the sample database
- Transform the physical data models to logical data models that can include a data access plan
- Create a data access plan and a selection policy
- Define selection criteria within the selection policy
- Define a data privacy policy to mask credit card information
- Define a data privacy policy to mask numeric data
- Create a data management service to copy and mask data

Time required

This module should take approximately 60 minutes to complete.

Prerequisites

This tutorial can be completed in the Optim Designer environment.

Creating a data design project

In this exercise, you will create a data design project in which to store your data models and definitions.

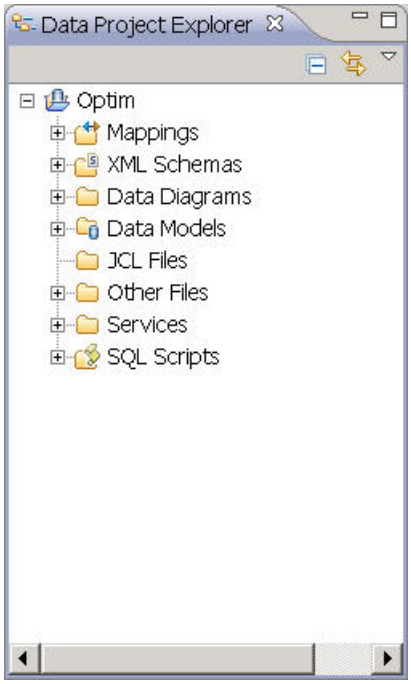
Before you create data models or other definitions, you must create a data design project in which to store your objects. You can store various types of objects in a data design project including data models, data management service definitions, and interoperability services.

To create a data design project:

1. From the main menu, click **File** > **New** > **Data Design Project**. The New Data Design Project wizard opens.
2. In the **Project Name** field, type `Optim`, then click **Finish**.

   If the Open Associated Perspective popup is displayed, click **No**. You will use the default Optim perspective.

   The Optim project is displayed in the Data Project Explorer.
3. Expand the Optim project in the Data Project Explorer to view the contents of the project.



Connecting to the Optim sample database

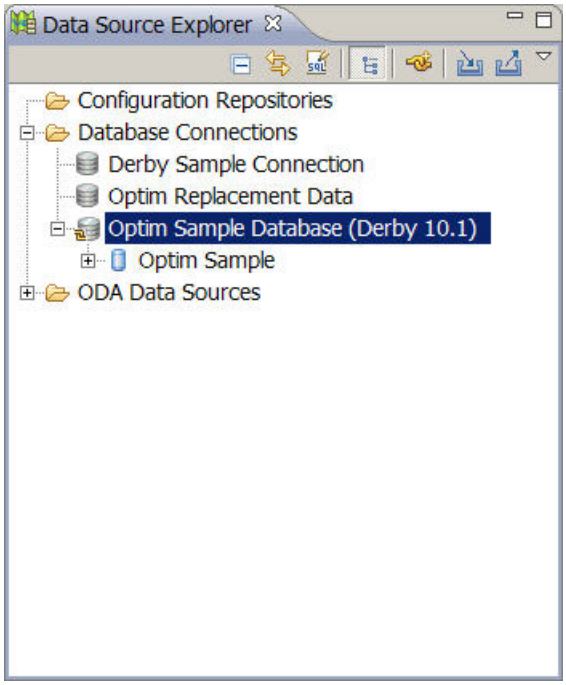
Optim Designer provides wizards that make it easy for you to connect to databases and to display the status of your connections. In this exercise, you will connect to the Optim sample database.

You will use the sample database to define physical and logical data models upon which Optim processes are based.

To connect to the sample database:

1. In the Data Source Explorer view, expand the **Database Connections** folder.
2. Right-click the Optim Sample Database connection definition and select **Connect**.

The connection definition will display the database type, Derby, and open to display the Optim Sample database.



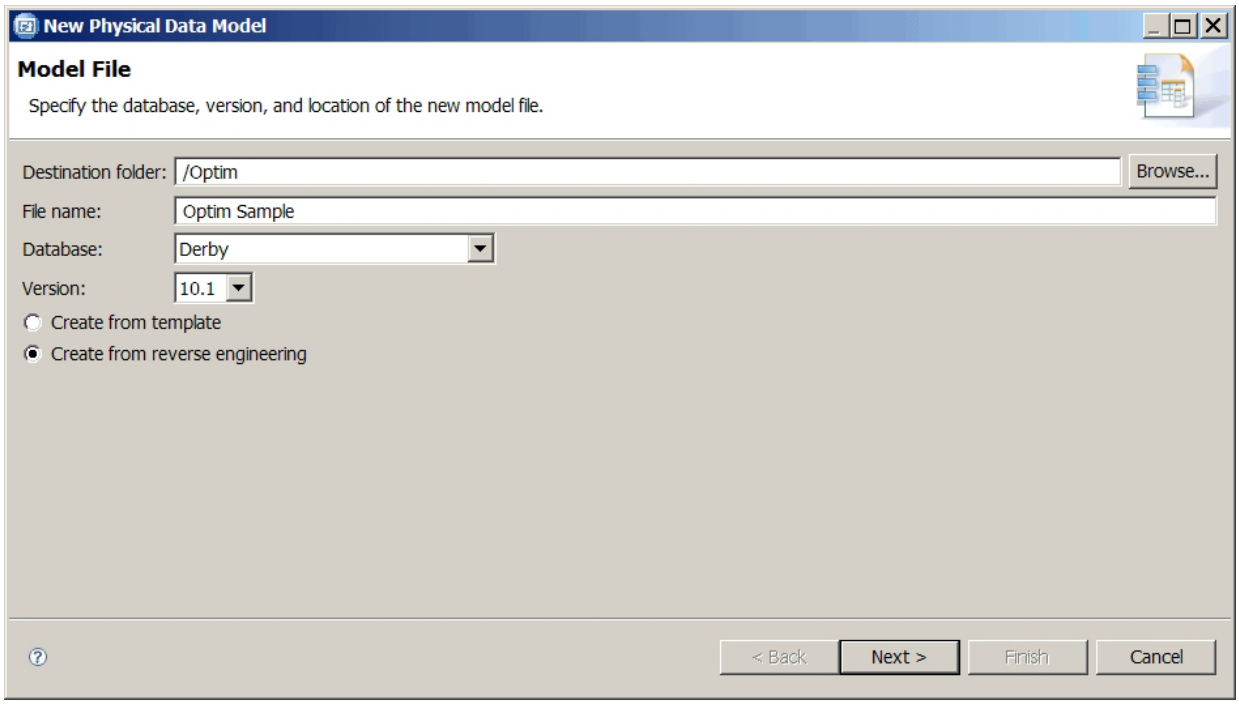
Creating a physical data model based on reverse engineering

In this exercise, you will create a physical data model. Physical data models are database-specific models that represents relational data objects (for example, tables, columns, primary keys, and foreign keys) and their relationships. A physical data model based on reverse engineering is created using the metadata in a source database.

You use physical data models to create logical data models, which describe the data used with Optim data management services and processes.

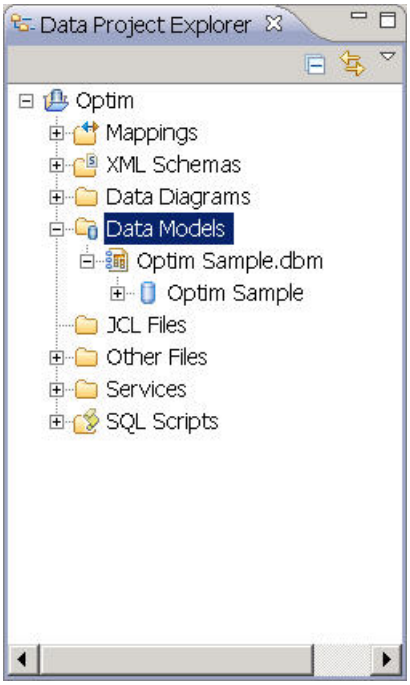
To create a physical data model based on reverse engineering:

1. In the Data Source Explorer view, right-click the **Data Models** folder and click **New** > **Physical Data Model**. The New Physical Data Model wizard opens.
2. On the Model File page, do the following:
   a. In **File Name**, type `Optim Sample`.
   b. From the **Database** list, select **Derby**.
   c. From the **Version** list, select **10.1**.
   d. Select **Create from reverse engineering**.
   e. Click **Next**.



3. In the Select Connection page, from the **Connections** area, select **Optim Sample Database**. Click **Next**.
4. In the Select Objects page, from the **Select objects** area, select **OPTIMUSER** and **OPTIMUSER2**. Click **Finish**.

The new physical data model, Optim Sample.dbm, will appear under the **Data Models** folder. The model will include the OPTIMUSER and OPTIMUSER2 schemas from the Optim Sample database.

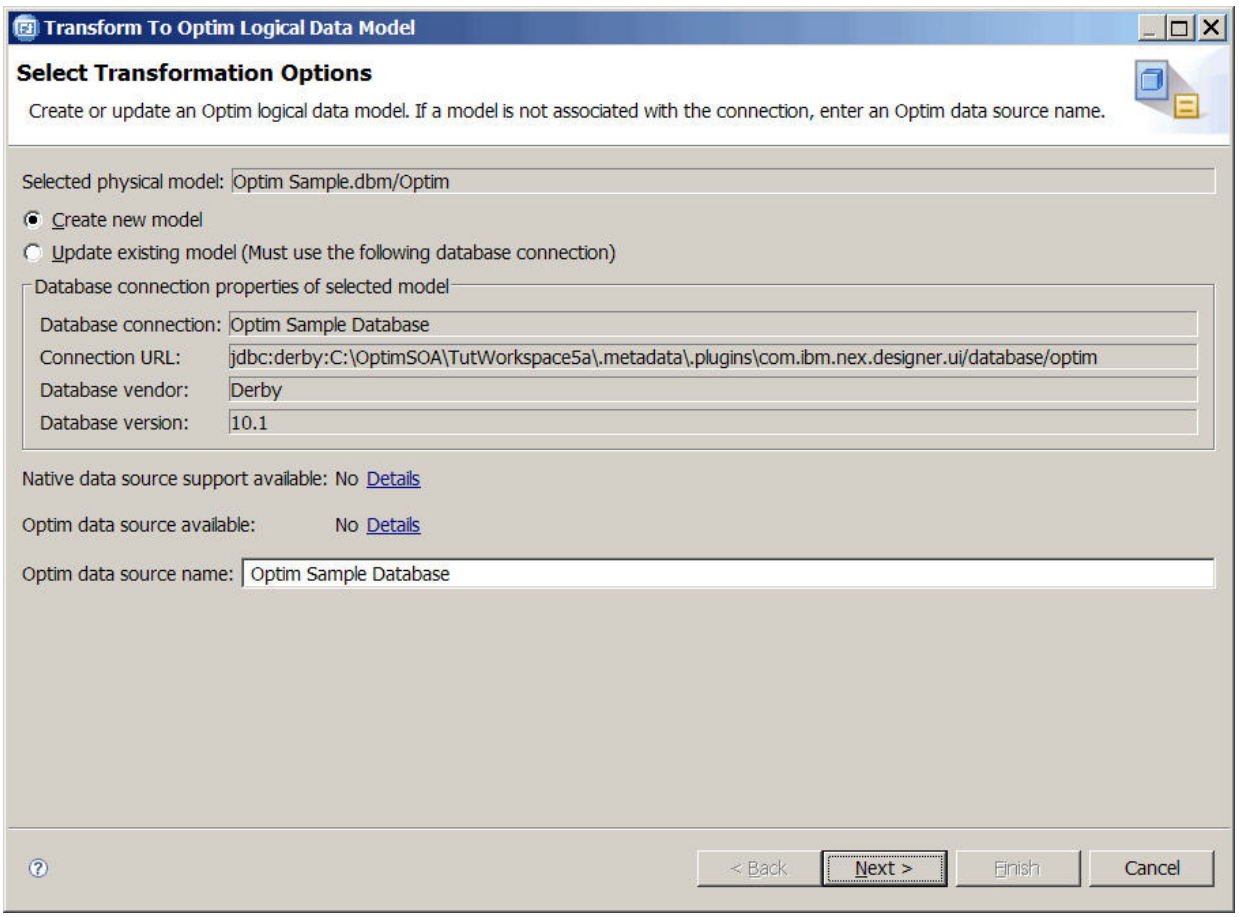


Transforming schemas in a physical data model to an Optim logical data model

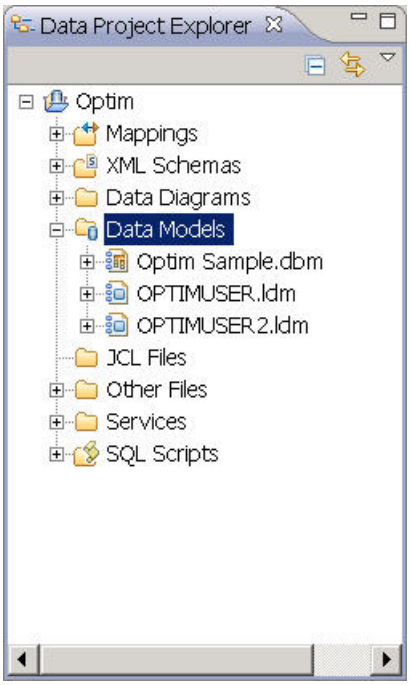
In this exercise, you will create logical data models from schemas in a physical data model. Logical data models are not specific to a database and describe the data used with Optim data management services and processes. An Optim logical data model is a logical data model that includes a data access plan, which contains policies that determine the data to copy or transform from a source logical data model used in an Optim data management service or process.

To transform schemas in a physical data model to an Optim logical data model:

1. In the Data Project Explorer, expand the **Data Models** folder, and expand the Optim Sample physical data model to to display the OPTIMUSER and OPTIMUSER2 schemas.
2. Right-click the OPTIMUSER schema, and click **Transform to Optim Logical Data Model**. The Transform to Optim Logical Data Model wizard opens.
3. On the Select Transformation Options page, select **Create new model** and use the default value for the **Optim Data Source**, *Optim Sample Database*. Click **Next**.



4. On the Enter Model Name and Project Folder page, type `OPTIMUSER` in **Name**. Click **Next**.
5. On the Transformation Results page, review the results of the transformation, and click **Finish**. The new logical data model, OPTIMUSER.ldm, will appear under the **Data Models** folder.
6. From the Optim Sample physical data model, right-click the OPTIMUSER2 schema, and click **Transform to Optim Logical Data Model**. The Transform to Optim Logical Data Model wizard opens.
7. On the Select Transformation Options page, select **Create new model**. Click **Next**.
8. On the Enter Model Name and Project Folder page, type `OPTIMUSER2` in **Name**. Click **Next**.
9. On the Transformation Results page, review the results of the transformation, and click **Finish**.



The new logical data model, OPTIMUSER2.ldm, will appear under the **Data Models** folder.

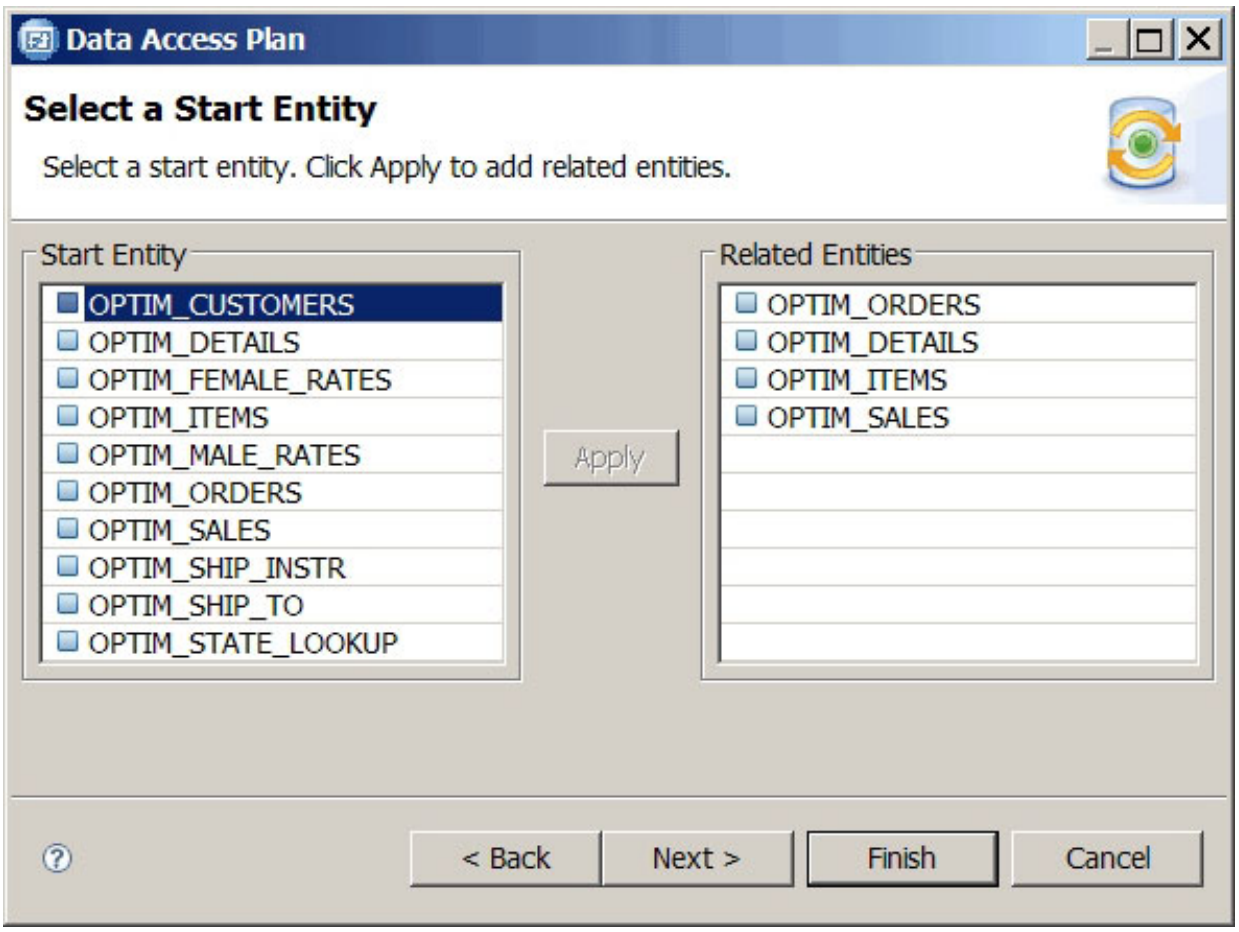
You have created two new Optim logical data models, OPTIMUSER and OPTIMUSER2.

Creating a data access plan and a selection policy

In this exercise, you will create a data access plan and a selection policy. A data access plan contains policies that determine which data to copy or transform from a source logical data model in an Optim data management service or process. A selection policy specifies the entities and attributes to use in an Optim data management service or process.

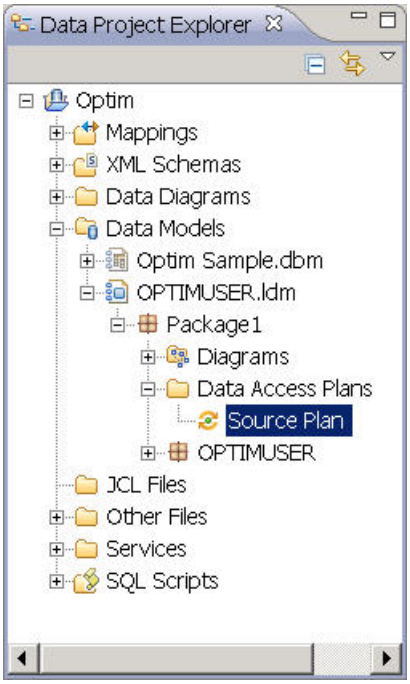
To create a data access plan and a selection policy:

1. In the Data Project Explorer, expand the **Data Models** folder, expand the OPTIMUSER logical data model to open the model, and expand the Package1 node to display the **Data Access Plans** folder.
2. Right-click the **Data Access Plans** folder and click **New** > **Data Access Plan**. The Data Access Plan wizard opens.
3. On the Data Access Plan Name page, type `Source Plan` in the **Name** field. Click **Next**.
4. On the Select a Package page, select the package with the schema name of the Optim sample database. Click **Next**.
5. On the Select Entity Options page, select **Select entities based on relationships with a start entity**. Click **Next**.
6. On the Select a Start Entity page, select **OPTIM_CUSTOMERS** from the **Start Entity** area, and click **Apply** to add the related tables to the **Related Entities** area. Click **Next**.



7. On the Select Reference Entities page, click **Finish**.
8. From the main menu, click **File** > **Save All**.

You have created a data access plan, Source Plan, which contains a selection policy that specifies OPTIM_CUSTOMERS as the start table and includes related tables in the OPTIMUSER schema.

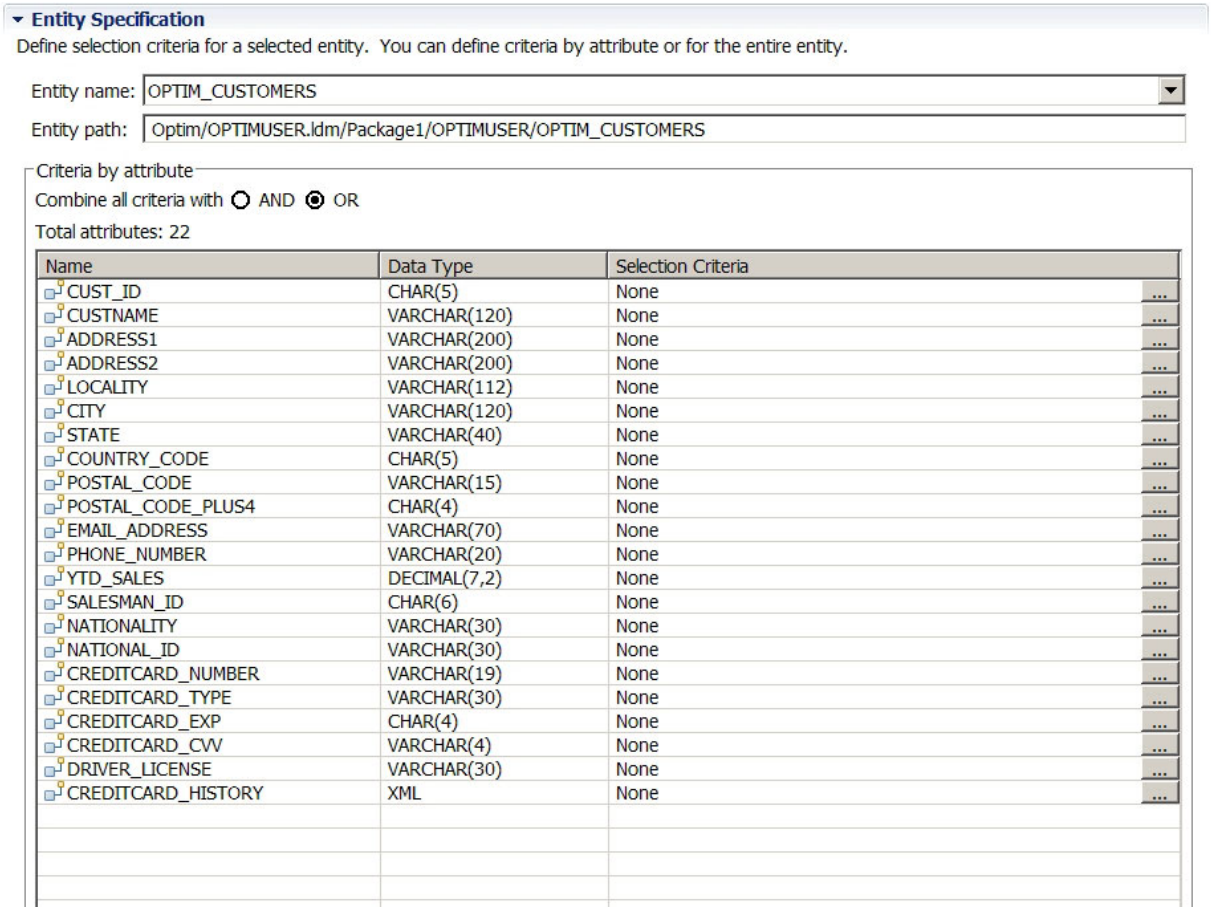


Defining selection criteria

In this exercise, you will define selection criteria for the selection policy in the Source Plan data access plan. Selection criteria allow you to pinpoint the data you want to use in an Optim data management service or process. You can select data according to values in one or more columns. Selection criteria must conform to SQL syntax and include relational or logical operators.

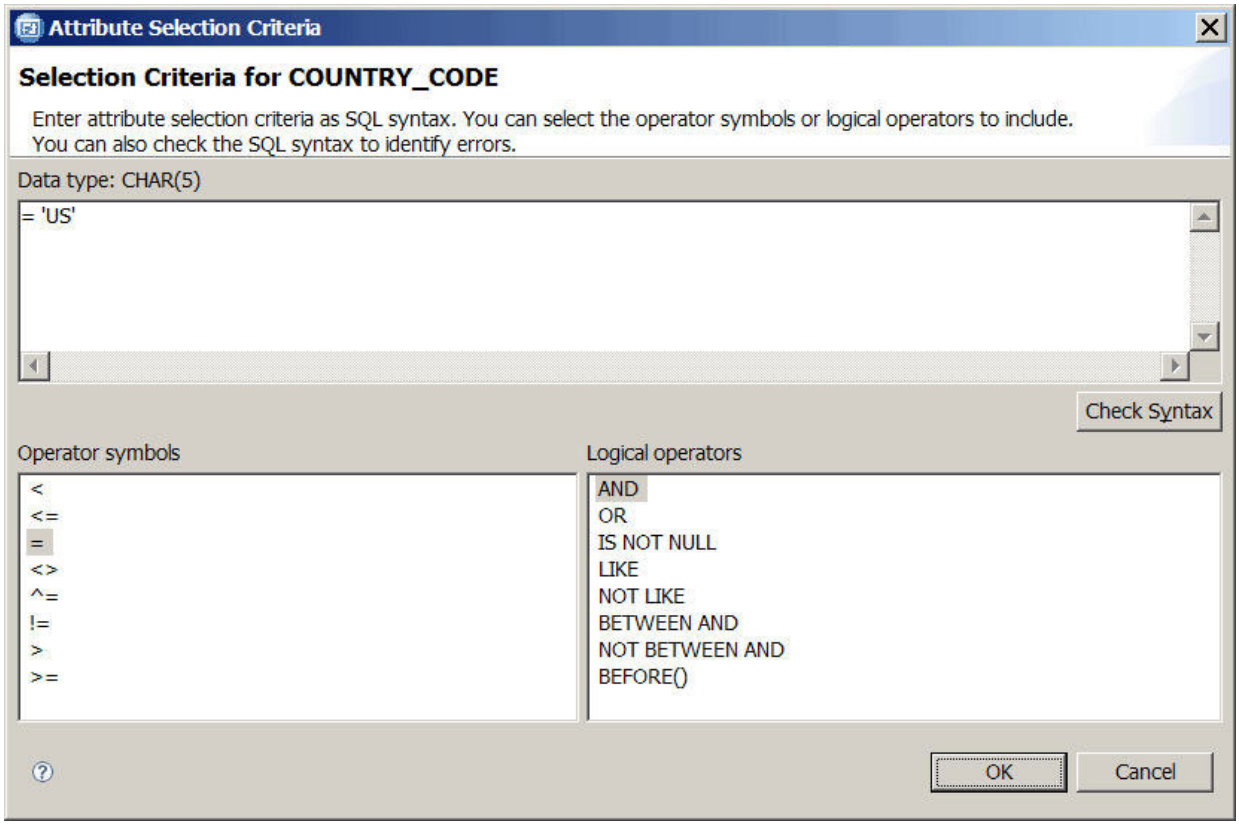
To define selection criteria:

1. In the Data Project Explorer, expand the **Data Models** folder, double-click the OPTIMUSER logical data model to open the model, expand the Package1 node, and expand the **Data Access Plans** folder.
2. Double-click **Source Plan**. The data access plan editor opens.
3. Click **Selection**. The selection policy editor opens.
4. In the **Entity Specification** area, select OPTIM_CUSTOMERS from the **Entity name** list.



The attributes for the OPTIM_CUSTOMERS entity are listed in the **Criteria by attribute** area.

5. Click the browse button in the **Selection Criteria** cell for the COUNTRY_CODE attribute. The Attribute Selection Criteria window opens.
6. Do the following in the Attribute Selection Criteria window:
   a. From the **Operator symbols** list, double-click **=**.
   b. In the editor area, type `'US'`. The following syntax should be entered: `='US'`.
   c. Click **OK** to return to the Selection Policy editor.



7. From the main menu, click **File** > **Save**.

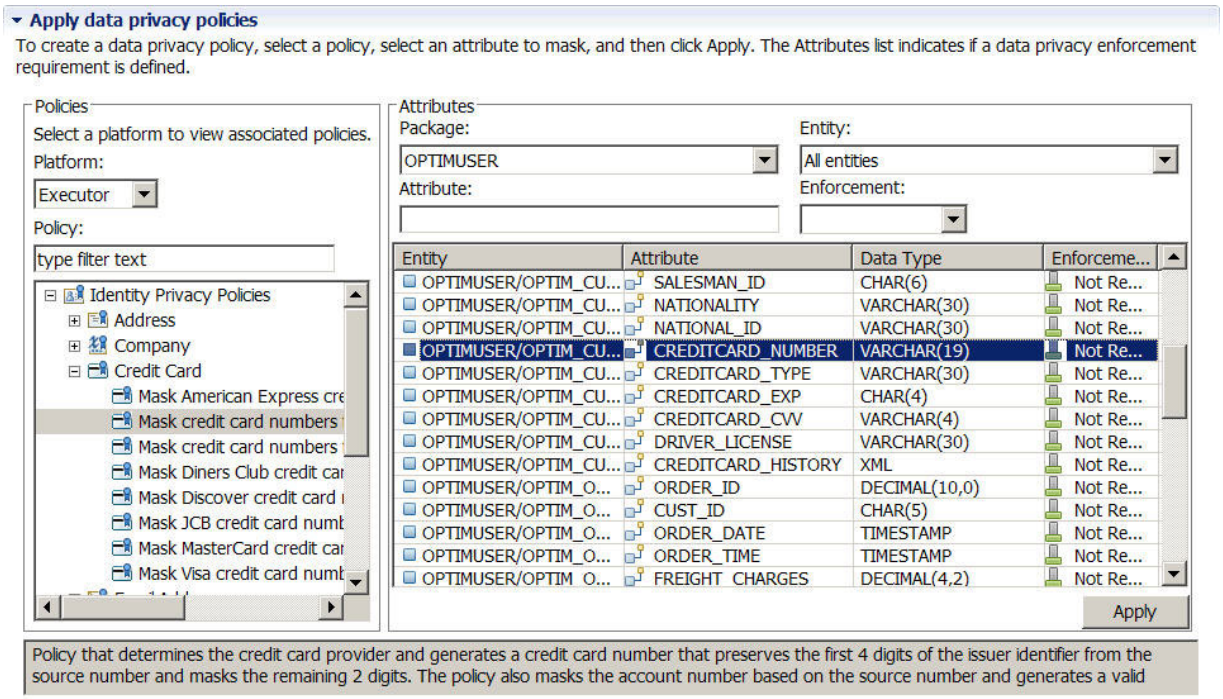
You have defined selection criteria that will only select rows from the OPTIM_CUSTOMERS entity in which the value of the COUNTRY_CODE attribute is 'US'.

Defining a data privacy policy to mask credit card numbers

In this exercise, you will define a data privacy policy to mask credit card numbers. The policy will be added to the Source Plan data access plan.

To define a data privacy policy to mask credit card numbers:

1. In the Data Project Explorer, expand the **Data Models** folder, double-click the OPTIMUSER logical data model to open the model, expand the Package1 node, and expand the **Data Access Plans** folder.
2. Double-click **Source Plan**. The data access plan editor opens.
3. Click **Data Privacy**. The data privacy editor opens.
4. From the **Policies** area, do the following:
   a. Select **Executor** from the **Platform** list.
   b. Expand **Identity Privacy Policies**, expand **Credit Card**, and select **Mask credit card numbers from all providers**.
5. In the **Attributes** area, select **OPTIM_CUSTOMERS** from the **Entity** list. The attributes in the OPTIM_CUSTOMERS entity are listed.
6. From the **Attributes** list, select **CREDITCARD_NUMBER**.



7. Click **Apply**. The new privacy policy, OPTIM_CUSTOMERS, will display in the **Data privacy policies in use** area.
8. From the **Data privacy policies in use** area, select OPTIM_CUSTOMERS. The properties for the policy will display below the **Data privacy policies in use** area.
9. In the properties area, select the **Credit Card Policy Option** tab.
10. Select **Mask credit card issuer?**.
11. From the main menu, click **File** > **Save**.

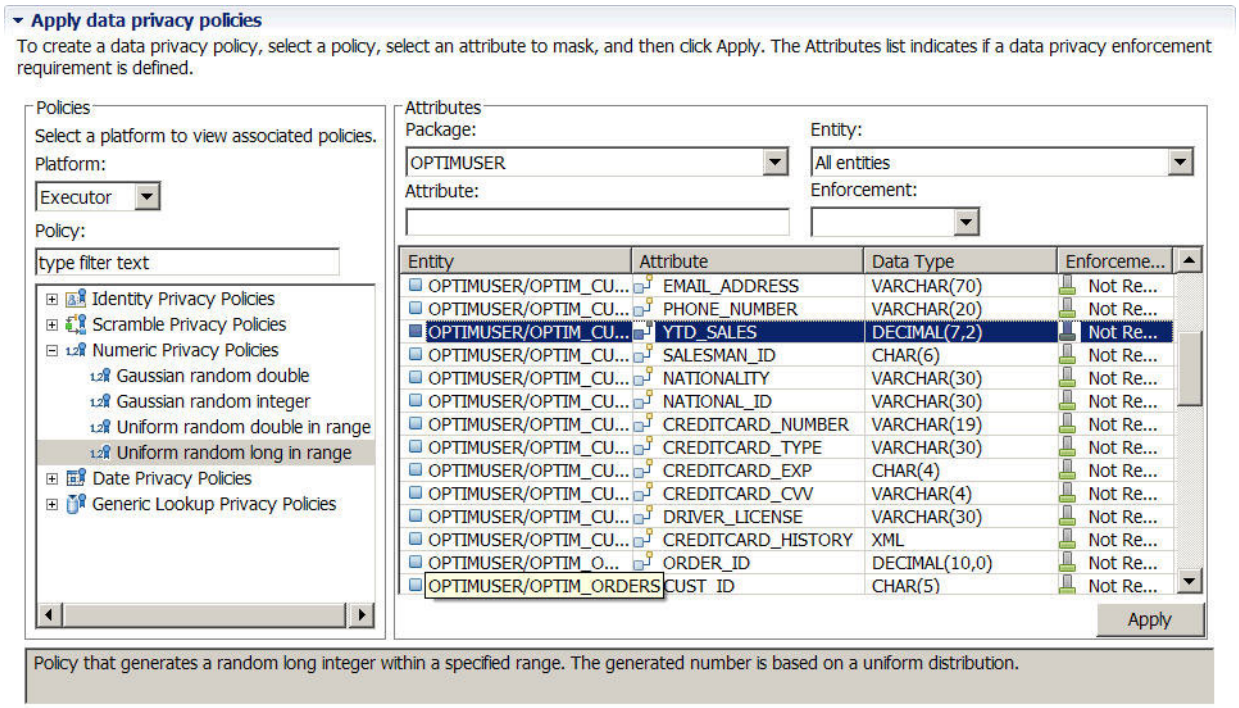
You have defined a privacy policy that will mask credit card numbers from all supported issuers in the CREDITCARD_NUMBER attribute of the OPTIM_CUSTOMERS entity.

Defining a data privacy policy to mask numeric data

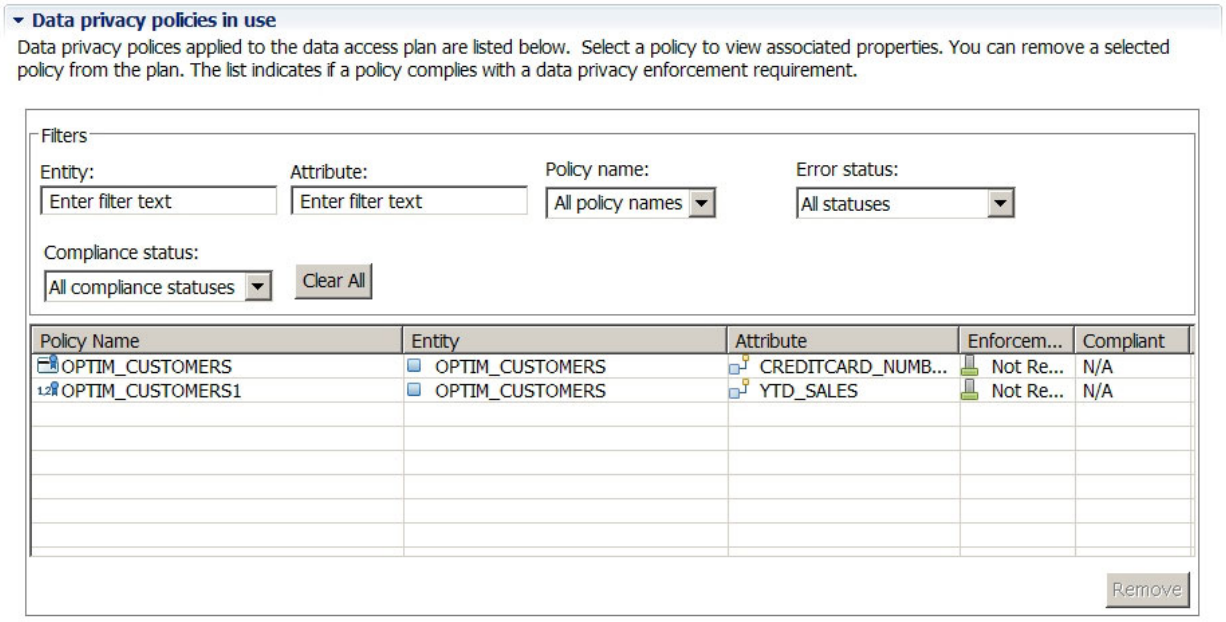
This is an optional exercise intended for Optim Data Privacy Solution users. In this exercise, you will define a data privacy policy to mask year-to-date sales numbers. The policy will be added to the Source Plan data access plan.

To define a data privacy policy to mask numeric data:

1. In the Data Project Explorer, expand the **Data Models** folder, double-click the OPTIMUSER logical data model to open the model, expand the Package1 node, and expand the **Data Access Plans** folder.
2. Double-click **Source Plan**. The data access plan editor opens.
3. Click **Data Privacy**. The data privacy editor opens.
4. From the **Policies** area, do the following:
   a. Select **Executor** from the **Platform** list.
   b. Expand **Identity Privacy Policies**, expand **Numeric Privacy Policies** and select **Uniform random long in range**.
5. In the **Attributes** area, select **OPTIM_CUSTOMERS** from the **Entity** list. The attributes in the OPTIM_CUSTOMERS entity are listed.
6. From the **Attributes** list, select **YTD_SALES**.



7. Click **Apply**. The Add Policy wizard opens.
8. On the Uniform Random Long In Range Options page, do the following:
   a. In **Range lower bound**, type `1000`.
   b. In **Range upper bound**, type `10000`.
   c. Click **Finish**.



The new privacy policy, OPTIM_CUSTOMERS1, will display in the **Data privacy policies in use** area.

9. From the main menu, click **File** > **Save**.

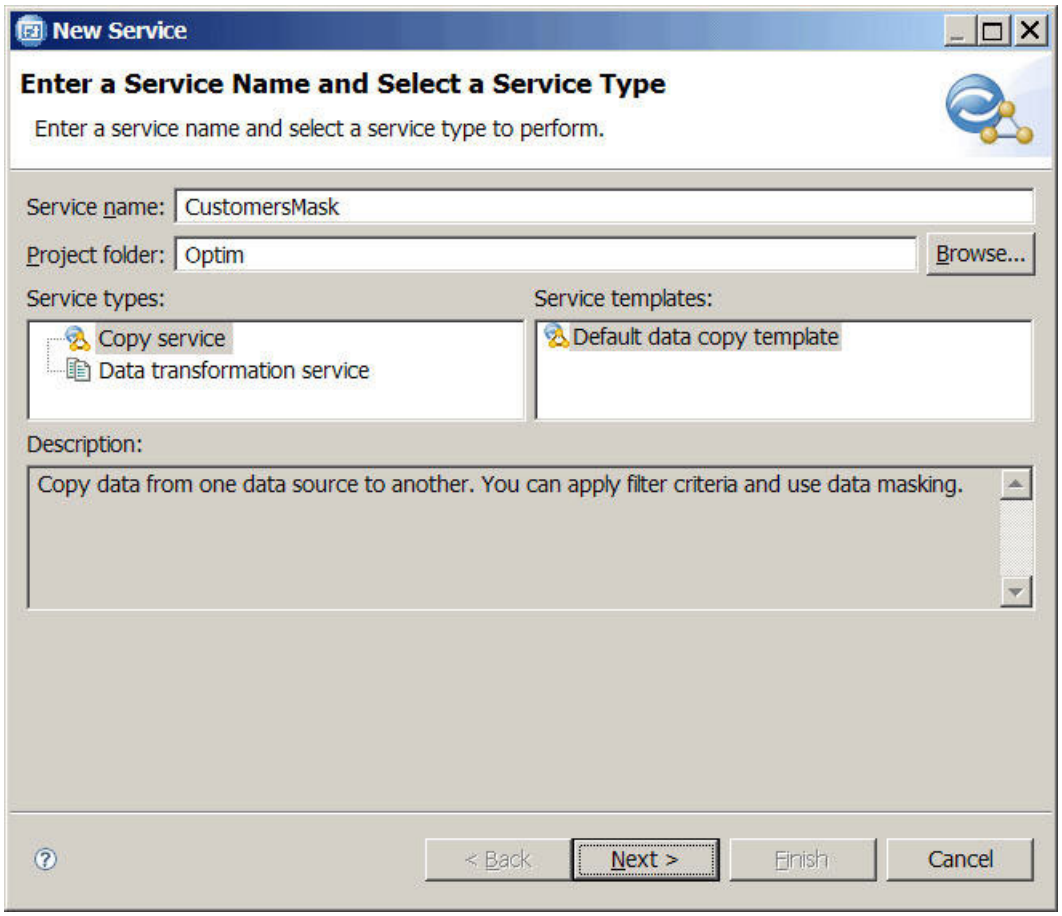
You have defined a privacy policy that will mask numbers by generating a random long integer within a specified range, 1000-10000, for the YTD_SALES attribute of the OPTIM_CUSTOMERS entity.

Defining an executor service to copy and transform data

In this exercise, you will define an executor service to copy and transform data defined in a source Optim logical data model.

To define an executor service to copy and transform data:

1. In the Data Source Explorer view, right-click the **Services** folder and click **New** > **Executor Service**. The New Service wizard opens.
2. On the Enter a Service Name and Select a Service Type page, do the following:
   a. In **Service name**, type `CustomersMask`.
   b. In the **Service Types** area, select **Copy service**.
   c. Click **Next**.

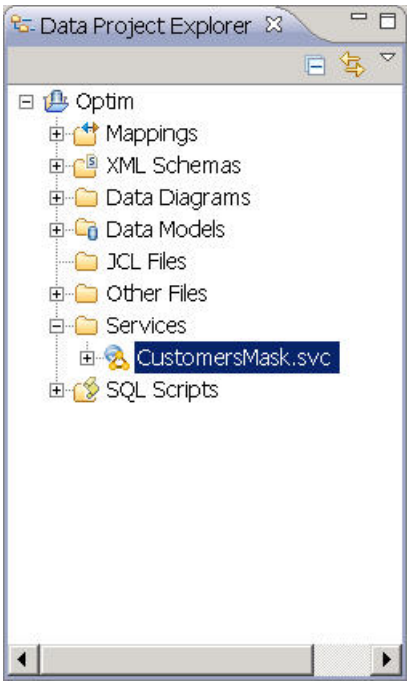


3. On the Select a Source Optim Data Source page, select **Optim Sample Database**. Click **Next**.
4. On the Select a Source Logical Data Model page, select **OPTIMUSER.ldm**. Click **Next**.
5. On the Select a Data Access Plan page, select **Source Plan**. Click **Next**.
6. On the Target Model Options page, select **Select a target model and use an auto mapping of entities**. Click **Next**.
7. On the Select a Target Optim Data Source page, select **Optim Sample Database**. Click **Next**.
8. On the Select a Target Optim Logical Data Model and Operation page, do the following
   a. From the **Target Operation Type** area, select **Insert**.

b. Select the **OPTIMUSER2.ldm** Optim logical data model.
c. Click **Next**.

9. On the Match Schema page, accept the default mapping. Click **Next.**.
10. On the Auto Map Results page, review the results of the auto mapping. Click **Finish**.

The new data management service, CustomersMask, will appear under the **Services** folder. The service will copy data defined in the source OPTIMUSER Optim logical data model and insert it into the target defined in the OPTIMUSER2 Optim logical data model. The service will use the Source Plan data access plan to determine which data to select and transform from the OPTIMUSER Optim logical data model. The service definition automatically created a mapping between entities in the source Optim logical data model and the target Optim logical data model.



Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM
IBM logo
DB2
AIX
Informix
InfoSphere
Optim

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Apache Derby is a trademark of The Apache Software Foundation.

Eclipse is a trademark of Eclipse Foundation, Inc.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Index

C

credit card policies
  credit card specific 50
  mask American Express credit card numbers 50
  mask credit card numbers from all providers 51
    creating 51
  mask credit card numbers from all providers based on provider name 52
    creating 52
  mask Diners Club credit card numbers 50
  mask Discover credit card numbers 50
  mask JCB credit card numbers 50
  mask MasterCard credit card numbers 50
  mask VISA credit card numbers 50
  overview 50

D

data access plans
  adding 18
  data sources 18
  editing 18
  overview 18
  selection policy 18
data management services
  exporting 43
  Optim registry 42
  overview 23
data models
  data access plans 18
  Optim Database Relationship Analyzer 15
  Optim logical data model 17
  overview 15
  physical data model based on reverse engineering 15
data privacy compliance requirements 78
data privacy policies
  date privacy policies 45
  editing 78, 79
  generic lookup privacy policies 72
  identity privacy 49
  JavaScript policy 75
  lookup 57
  overview 45
  random shuffle function 75
database lookup policies
  overview 72
Database Relationship Analyzer
  physical model 16
date privacy policies
  age 45
    creating 46
date privacy policies *(continued)*
  overview 45
  random date in range 46
    creating 46
  round date to month 47
    creating 47
  round date to year 48
    creating 48
disable constraints policy
  creating 26
  enabling or disabling all constraints 26
  enabling or disabling selected constraints 26
  overview 26

E

e-mail address policies
  auto-generated e-mail name 49
    creating 49
  formatted e-mail name 49
    creating 49
  overview 49
executor service
  configuring Optim executor 28
  creating 23
  creating a copy service 23
  creating a data transformation service 24
  editing 24
  executing 28
  managing the Optim license 28
  overview 23
  testing overview 28

G

generic lookup privacy policies
  hash lookup policy 73
    creating 73
  lookup policy 72
    creating 72
  random lookup policy 74
    creating 74

I

identity privacy lookup policies
  address information 59
  creating 59
  given name information 60
  hash lookup 57
  mask a company name 61
  overview 57
  personal information 60
  random lookup 57
  surname information 61
identity privacy policies
  credit card policies 50
identity privacy policies *(continued)*
  e-mail address policies 49
  national ID policies 52
  numeric privacy policies 61
  overview 49
  scramble privacy policies 66

J

JavaScript policy
  adding a JavaScript file 77
  creating 77
  editing a JavaScript file 77
  examples 77
  overview 75

M

migrating Optim Designer workspaces 6

N

national ID policies
  country specific national ID policies 52
    creating 52
  mask Canadian Social Insurance Numbers 53
  mask French National Institute for Statistics and Economic Studies Numbers 53
  mask Italian Fiscal Code Numbers 54
  mask national ID numbers based on the country name or country code 56
    creating 57
  mask Spanish Fiscal Identification Numbers 54
  mask United Kingdom National Insurance Numbers 55
  mask United States Social Security Numbers 56
  overview 52
native data source connections 11
numeric privacy policies
  Gaussian random double 61
    creating 62
  Gaussian random integer 62
    creating 63
  overview 61
  random number function 63
    creating 63
  sequential number function 64
    creating 64
  uniform random double in range 64
    creating 65
  uniform random long in range 65
    creating 65

O

Optim data sources 11
Optim Database Relationship Analyzer
  defining a connection profile 16
  setting up 16
  using 15
Optim Designer
  accessibility features 9
  database connections 11
  database support 8
  masking data 4
  overview 1
Optim for z/OS interoperability services
  Add Host window 39
  overview 36
  z/OS host configuration 39
Optim interoperability services
  column maps 41
  defining a DB alias 33
  distributed archive service 29
  distributed convert service 30
  distributed delete service 30
  distributed extract service 31
  distributed insert service 31
  distributed load service 31
  distributed restore service 32
  editing 40
  exporting definitions from services to OEF 34
  exporting requests 34
  exporting requests from Optim directory to OEF 35
  exporting requests to Optim directory 34
  importing requests 35
  importing requests from OEF to Optim directory 35
  importing requests from OEF to services 35
  Linux, UNIX, and Windows 29
  Optim directory 32
  Optim for z/OS services 36
  Optim request definitions 40
  Optim server name 33
  overview 29
  pr0cmnd location 32
  testing 41
  transforming request to Optim interoperability service 36
  z/OS archive service 36
  z/OS convert 36
  z/OS delete service 37
  z/OS extract service 37
  z/OS insert service 37
  z/OS load service 38
  z/OS restore service 38
Optim license
  defining location 29
  managing 28
  trial license 29
Optim perspective
  Data Project Explorer 5
  Data Source Explorer 6
  overview 5
Optim registry
  entering default location 42
  overview 42
Optim registry *(continued)*
  publishing a service 42
  secure connection 43

P

physical data model based on reverse engineering
  defining 15
  overview 15

R

random shuffle function 75
  creating 75

S

sample data
  EXTENDED_LOOKUP schema 7
  lookup tables 7
  Optim Replacement Data 7
  Optim Sample Database 7
  overview 7
scramble privacy policies
  overview 66
  repeatable replacement 66
    creating 66
  repeatable replacement by regular expression 67
    creating 68
  replace characters 68
    creating 69
  replace characters by regular expression 69
    creating 70
  scramble characters 70
    creating 70
  scramble characters by regular expression 71
    creating 71
  simple scramble characters 71
    creating 72
  supported character sets 72
selection criteria
  defining 20
selection policy
  adding an entity 19
  changing entity selection 19
  changing start entity 19
  defining related and reference entities 19
  editing 18
  removing an entity 20
  selection criteria 20
service diagnostics policy
  creating 27
  overview 27
service plans
  disable constraints policy 26
  editing 25
  editing update policy 27
  overview 24
  service diagnostics policy 27
  source to target mapping 25
source to target mapping
  adding an entity 25
  changing a target Optim logical data model 25
  overview 25
  removing an entity 25
  restoring auto mapping 26

Printed in USA