HP Test Data Management

Software version: 1.10

Tutorial

Document release date: October 2010
Software release date: October 2010

Legal notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted rights legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Licensing

The use of HP products is governed by the terms and conditions of the applicable End User License Agreement (EULA).

Copyright notices

Trademark notices

Microsoft®, Windows®, Windows NT®, Windows XP®, and Windows Vista® are U.S. registered trademarks of Microsoft Corporation.

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Contents

About this document

HP Test Data Management provides powerful tools to design a test data management solution that copies data out of your production database for upload into a test database.

This tutorial is designed to help you get started with HP Test Data Management.

This guide provides information about:

- an example test data management solution
- steps for designing the example solution
- deployment of the example solution
- execution of the example solution

Intended audience

This guide is intended for:

- Test data developers

Prerequisites

Prerequisites for using this product include:

- Knowledge of the operating system
- Database knowledge
- Application knowledge

Related documentation

In addition to this guide, please refer to other documents for this product:

- *HP Test Data Management Installation Guide*

  Explains how to use the Installer to install the product.
- *HP Test Data Management Concepts Guide*

Explains the major concepts of test data management in general and HP Test Data Management in particular.

- *HP Test Data Management Designer User Guide*

  Explains how to use the Designer component to design, build, test, and deploy your test data management projects.

- *HP Test Data Management Web Console and Query Server User Guide*

  Explains how to use the Web Console component to run, monitor, and administer the business flows that copy data to and from the database. This guide also explains how to install, configure, and use the query server to access data that has been extracted from the database.

- *HP Test Data Management Troubleshooting Guide*

  Explains how to diagnose and resolve errors, and provides a list of common errors and solutions.

- *HP Test Data Management Release Notes*

  Lists any items of importance that were not captured in the regular documentation.

The latest documentation for the most recent HP Test Data Management release can be found on:

http://support.openview.hp.com/selfsolve/manuals

Document conventions and symbols

Convention	Element
`[ ]`	Indicates that the enclosed element is optional and may be left out.
`{ }`	Indicates that you must specify one of the listed options.
|	Separates alternatives.

Convention	Element
<parameter_name>	You must supply a value for a variable parameter.
. . .	• Indicates a repetition of the preceding parameter. • Example continues after omitted lines.
Medium blue text: Figure 1	Cross-reference links and e-mail addresses
Medium blue, underlined text (http://www.hp.com)	Web site addresses
Bold	• Key names • GUI elements that are clicked or selected, such as menu and list items, buttons, and check boxes
Italics	Text *emphasis*
Monospace	• File and directory names • Text displayed on the screen, such as system output and application messages • Code syntax
Monospace, italic	You must supply a value. • Code variables • Command-line variables

△ CAUTION Indicates that failure to follow directions could result in damage to equipment or loss of data.

NOTE Provides additional information.

TIP Provides helpful hints and shortcuts.

RECOMMENDATION Provides guidance from HP for a best practice or for optimum performance.

Documentation updates

For documentation for all versions of HP Test Data Management, you can go to:

http://support.openview.hp.com/selfsolve/manuals

NOTE This documentation is written to the latest patch version. If you have not installed the latest patch, there may be items in this documentation that do not apply to your environment.

Subscription service

HP strongly recommends that customers sign up online using the Subscriber's choice web site:

http://www.hp.com/go/e-updates

- Subscribing to this service provides you with e-mail updates on the latest product enhancements, versions of drivers, and firmware documentation updates as well as instant access to numerous other product resources.
- After signing up, you can quickly locate your products under Product Category.

Support

You can visit the HP Software Support web site at:

http://www.hp.com/go/hpsoftwaresupport

HP Software Support Online provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels and register for HP Passport, go to:

http://support.openview.hp.com/new_access_levels.jsp

1 Introduction

This chapter provides you with an overview of the test data management solution building process and the tutorial itself.

This chapter includes:

Tutorial overview

This tutorial is designed to introduce you to HP Test Data Management. The tutorial walks you through the following high level tasks using the sample Demarc database objects:

- Creating a model-based solution.
  - Designing a model of database objects to be extracted
  - Defining rules on tables in the model to subset the data
  - Designing an extraction cartridge that applies the data model and its rules
  - Previewing your model and cartridge
  - Creating business flows to call the extraction cartridge
- Deploying and running your business flows.
  - Creating an environment.
  - Creating Web Console users.
  - Deploying the business flow.
  - Running the business flow from the Web Console.
  - Confirming the results of the business flow.
  - Changing the business flow and redeploying it.
- Creating a spreadsheet from the extracted data for test input.
- Creating a schema-based cartridge (optional).
  - Designing a schema-based, extraction/upload cartridge.

- Defining rules on tables in the cartridge to subset the data.
- Creating a business flow to call the extraction/upload cartridge.
- Applying data masks to columns

Tutorial prerequisites

Before starting this tutorial, ensure all of the following have been completed:

- You have installed HP Test Data Management 1.10 on your computer. For details, see the *HP Test Data Management Installation Guide*.
- You have installed a database that is supported by HP Test Data Management 1.10. For details, see the *HP Test Data Management Installation Guide*.

  The steps and screen images in this tutorial assume an Oracle, SQL Server, or DB2 database. If you use another database, you can still follow the steps, but you may encounter some small differences in the steps and/or the appearance of the product.
- You have run the Web Console to set up a repository.

For details See Chapter 5, Starting and configuring the Web Console, of the *HP Test Data Management Web Console and Query Server User Guide*.

- You have noted all of the following:
  - database administrator user name
  - database administrator password
  - HP Test Data Management repository user name (for example, obt_rep)
  - HP Test Data Management repository password
  - HP Test Data Management encryption key

Test data management concepts

For conceptual information about Test Data Management, refer to the *HP Test Data Management Concepts Guide*.

Planning your test data management solution

For the purposes of this tutorial, the scenario is provided to you, which obviates the need for any planning activities. When you come to build your own solution, you will need to spend considerable time analyzing the data, applications, and environment before you start building your project in Designer. For information about how to plan for your test data project, refer to *HP Test Data Management Concepts Guide*.

Designer overview

Most of the work of developing a test data management solution is performed in Designer. Designer is a powerful graphical development environment used to:

- Model data.
- Apply rules.
- Design cartridges.
- Design business flows that employ cartridges and implement additional logic.
- Preview models and cartridges for testing purposes.
- Deploy cartridges and business flows to a specified environment (local or remote).

Designer drastically improves the productivity of test data developers. Developers no longer need to spend hours writing and debugging complex SQL. They simply drag and drop on the editor to include tables and relate them to other tables. Once the model is defined, they can point and click to define rules on tables as desired. When the business flows are ready, the developer can deploy them to any supported environment, local or remote, from Designer.

Navigation and interaction with the various components within Designer is consistent, and once familiar with the patterns in one portion of Designer, you should be able to work with any part of Designer.

This section contains:

Designer main window

After you have created a project with a connection, the Designer main window is displayed. You use the main window to define your project. The following figure shows an example of the main window and descriptions of its editors.

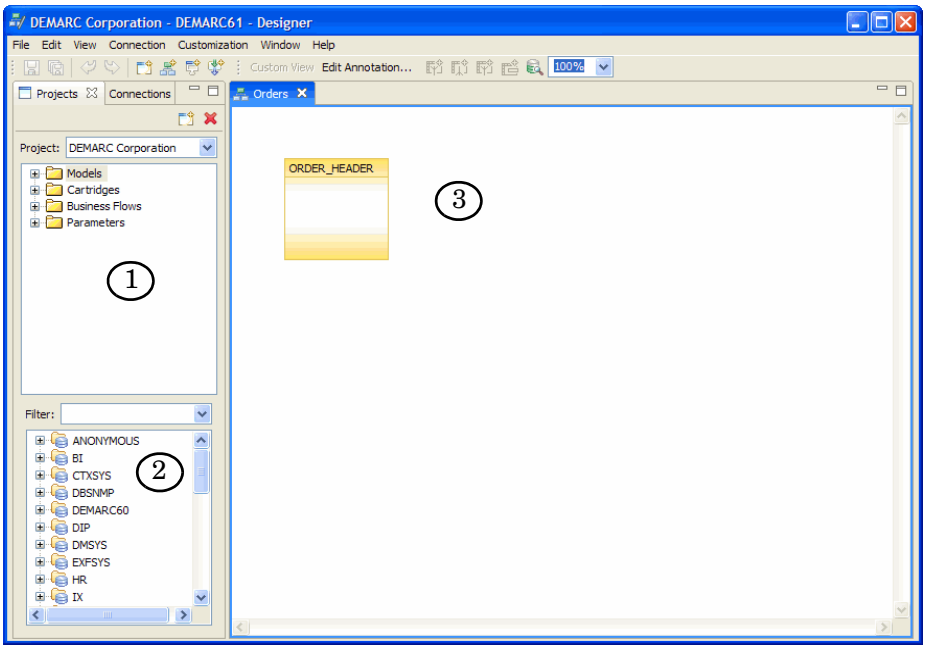


Table 1 Designer main window

Legend	Name	Description
1	Project Navigator	Shows the files contained within the project.
2	Database Navigator	Shows the content of the database or local cache for your open project.
3	Editor	The functional areas or work spaces used to define model, cartridge, and business flow components, to view preview data, and to view database table data.

Toolbar buttons

These are the main toolbar buttons used in Designer. As you use Designer, you will find that the available tools may change to match the mode of the tool and the selected object. The purpose of this section is to introduce you to the most common toolbar buttons.

Table 2 Toolbar buttons

Icon	Name	Description
	Save	Save the active window.
	Save All	Save work in all currently open windows and tabs.
	Undo	Undo the last change.
	Redo	Redo the last undone action.
	Create New Project	Enables you to start creating a new project.
	Create New Model	Enables you to start creating a new model.
	Create New Cartridge	Enables you to create a new cartridge, if you have models created.
	Create New Business Flow	Enables you to create a new business flow, if you have cartridges created.
	Custom View	Toggles custom view mode on or off. Custom view provides visual cues for locked projects that alert you to the customizations you are making and whether they are supported.
	Add Chaining Table	Enables you to add a chaining table to a model.
	Add Transactional Table	Enables you to add a transactional table to a model.
	Add Lookup Table	Enables you to add a lookup table to a model.

Table 2 Toolbar buttons

Icon	Name	Description
	Add Rule	Enables you to add an eligibility rule to a model object.
	Preview	Enables you to access the preview functionality of Designer.
	Refresh	Refreshes the data in preview from the database.

Summary and next steps

In this chapter you learned about:

- the overall structure of the tutorial
- the prerequisites for building the tutorial
- the basic concepts of HP Test Data Management
- the interface of Designer

You are now ready to begin building the example test data management solution.

2 Configuring the Demarc data

To follow the instructions in this tutorial, you must have the sample Demarc data set loaded in your database.

This chapter explains how to obtain and load the Demarc data.

This chapter includes:

Loading the Demarc data

The example in this tutorial is based upon the Demarc data set. You must install this schema and populate it before you can start the tutorial.

To load the data into your database, perform the following steps:

1. Install and configure the database of your choice. See Tutorial prerequisites (page 10) for information about which databases you can use.

   NOTE The demo data loader does not support Oracle RAC or SQL Server Windows authentication.

2. Install and configure HP Test Data Management as described in the *HP Test Data Management Installation Guide*.
3. Open a command window. On MS Windows, select **Start > Run**, type **`cmd`**, and click **OK**.
4. Change to the `bin` directory where you installed HP Test Data Management. For example, on MS Windows:

   ```
   cd c:\Program Files\HPTDM\obt\bin
   ```

   or on Unix:

   ```
   cd /home/HPTDM/obt/bin
   ```

5. Type the command appropriate for your database:

   On MS Windows:

   ```
   load_demo oracle
   load_demo sqlserver
   load_demo sybase
   load_demo db2
   load_demo generic
   ```

On Unix:

```
./load_demo.sh oracle
./load_demo.sh sqlserver
./load_demo.sh sybase
./load_demo.sh db2
./load_demo.sh generic
```

NOTE The generic option is for JDBC/ODBC data sources.

6 Respond to the prompts. Default values are displayed next to the prompts inside of square brackets [ ]. It may take a few minutes for the scripts to complete running.

TIP If you want to use a schema name other than DEMARC, enter the desired name when prompted for demo schema/username.

TIP Note that SQL Server database names are case sensitive. Hence, if you accept the default name, you must use DEMARC when referencing it.

7 To confirm that the scripts executed properly, check the log files located in *<install_dir>*\obt\demo for any errors:

- ngfloadlog.log for DB2
- ogfloadlog.log for Oracle
- sgfloadlog.log for SQL Server
- ygfloadlog.log for Sybase

8 If you have an ad hoc query tool, you can use it to confirm the presence of DEMARC and its tables.

Obtaining the tutorial solution

A completed version of the tutorial is shipped with HP Test Data Management for your reference. You may wish to open this project and review it after you have attempted to complete the tutorial yourself.

To obtain the tutorial solution project:

1 Launch Designer.

- On MS Windows, from the Start menu, open Designer by selecting **HP Test Data Management > Designer**.
- On Linux, use the designer desktop link, if you opted to create the links at install time. Otherwise, navigate to <*install dir*>/obt/bin and type:

  ```
  ./designer.sh
  ```

When you start Designer for the very first time, you are automatically prompted to create a new project.

The first field is the project Name. The second field is the Database connection used to get the table definitions.

TIP If you are not prompted to create a new project, select **File > New Project**. Or, you can click the New Project icon.

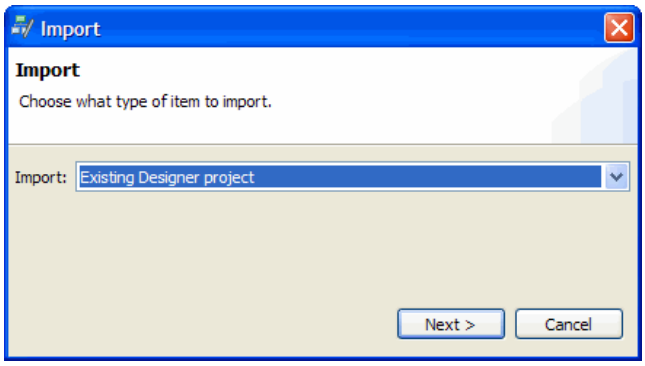
In the Name field, type **`DEMARC Orders App v1 soln`** as the name of your new project.

2 For Database, if you already created a connection to the database with DEMARC schema, choose the connection from the pull-down list. Otherwise, click **New** to set up such a database connection.

For more information on creating projects and database connections, refer to Creating a schema-based cartridge (page 85).

3 Once the New Project dialog box is filled out, click **OK**.

4 Select **File > Import**. The Import dialog box displays.



5 Choose **Existing Designer project** from the list.

6 Click **Next**. The Import Existing Project dialog box appears.

7 Browse to the location of the tutorial solution project. On MS Windows, it can be found in <*install dir*>`\obt\demo\project`. On Unix, it can be found in <*install dir*>`/obt/demo/project`.

8 Select `tutorial_soln_<db_type>.hdp`, where `<db_type>` is your database type (`oracle`, `sqlserver`, or `db2`).

9 Click **Open**. You should now have a complete, working version of the tutorial solution project to which you can compare your own solution.

Summary and next steps

In this chapter, you learned:

- how to run the scripts to load the sample data into the database you plan to use for the tutorial.
- how to obtain a completed version of the tutorial

The next step in building your test data management solution is to create a project with a schema-based cartridge, rules, and masking, and test it by previewing the data.

3 Creating a subset of data

One approach to creating test data is to build a data model in Designer that reflects the relationships among the tables from which you want to extract data. This chapter walks you through developing a model and then building a cartridge based on that model.

This chapter includes:

Creating a project

Projects provide you with a way to organize your test data management definitions. For example, you might collect in a single project all of the models, cartridges, business flows, and parameters required to extract the test data associated with a particular version of a particular application.

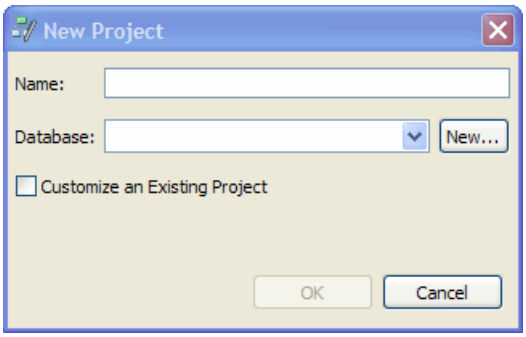
1 Launch Designer.

— On MS Windows, from the Start menu, open Designer by selecting **HP Test Data Management > Designer**.

— On Linux, use the designer desktop link, if you opted to create the links at install time. Otherwise, navigate to <*install dir*>`/obt/bin` and type:

```
./designer.sh
```

When you start Designer for the very first time, you are automatically prompted to create a new project.

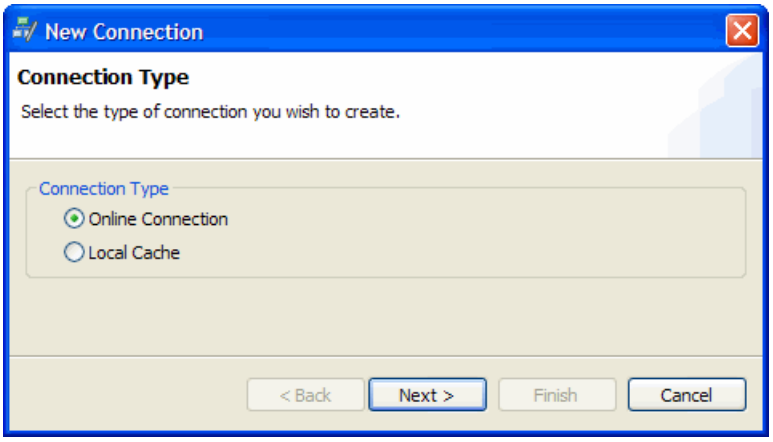


The first field is the project Name. The second field is the Database connection used to get the table definitions.

TIP If you are not prompted to create a new project, select **File > New Project**. Or, you can click the New Project icon.

In the Name field, type **`DEMARC Orders App v1`** as the name of your new project.

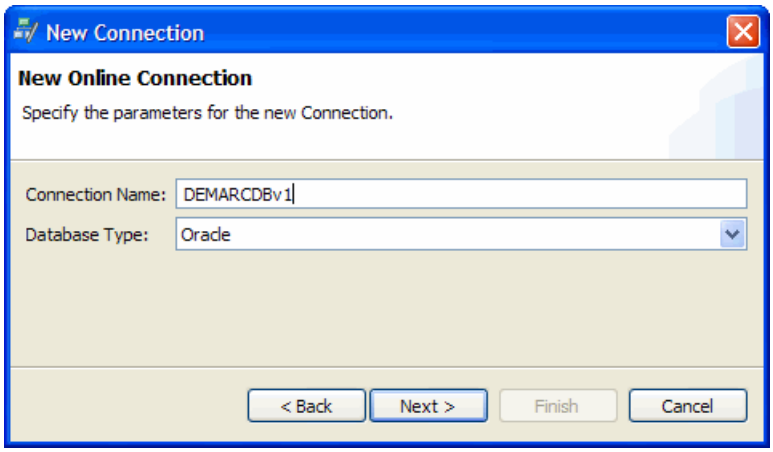
2 Leave the Database field blank and click **New** to set up a database connection.



3 Select **Online Connection** and click **Next** to create a connection to a database.

NOTE The Local Cache enables you to pick a local cache of previously stored (database) metadata. This option is useful if, for example, you are working and do not have access to the database over a network, or if you are dealing with a large number of database objects.

4 Type **`DEMARCDBv1`** in the Connection Name field.



5 Select your database type from the drop-down list.

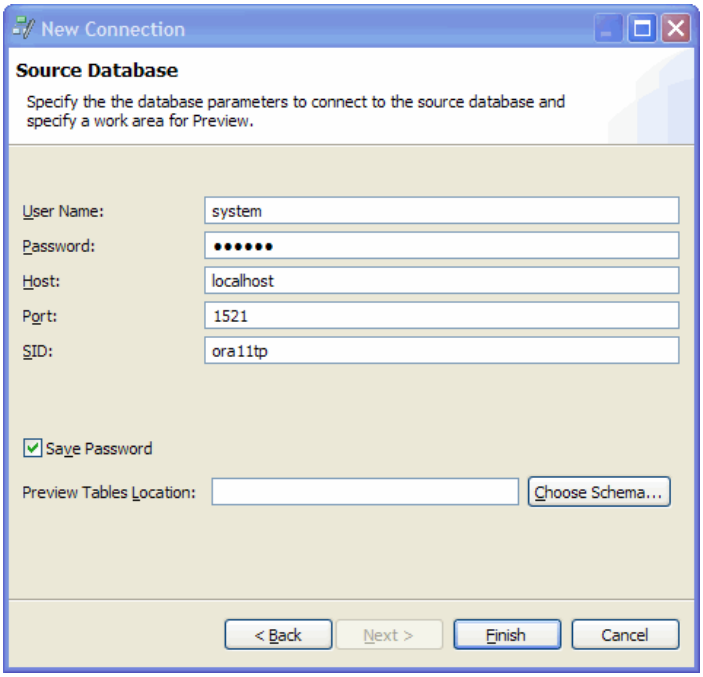
6 Click **Next.**

7 Type the connection information in the Source Database page.

NOTE Throughout the tutorial, the steps and screen images assume Oracle, SQL Server, or DB2. If you are using another supported database, you will notice some small differences in the steps and appearance of the product.

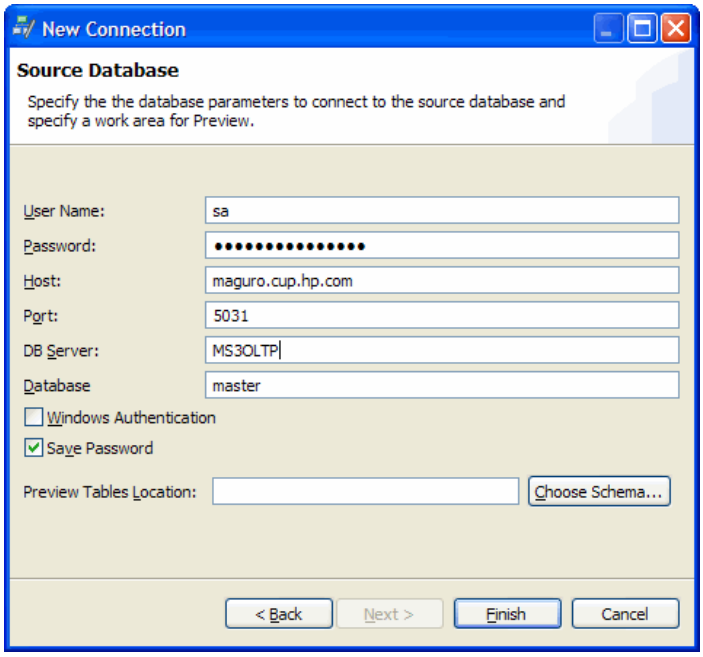
For Oracle

An example for Oracle, the last field on this panel prompts you for the SID (or service name).



For SQL Server

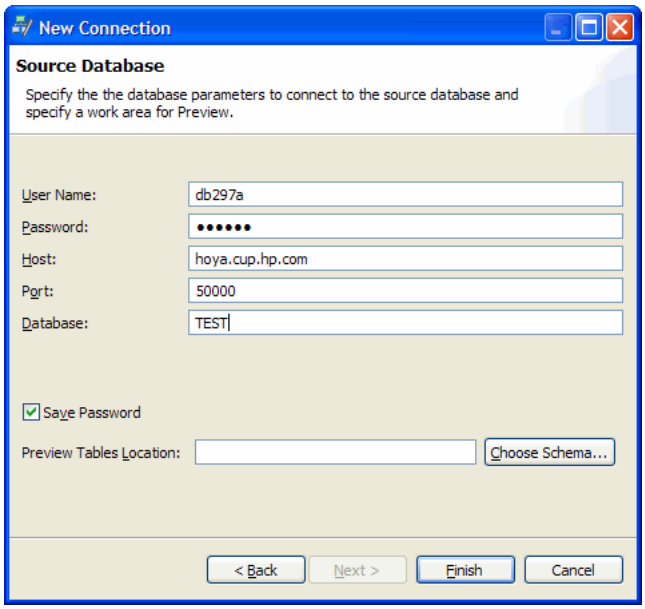
An example for SQL Server 2005, you are prompted for the DB Server and Database.



TIP For SQL Server, Windows Authentication indicates that the operating system login for the machine is the same as the SQL Server login, and once you are logged into the machine, you need not authenticate again for the SQL Server instance. If you do not select this option, the SQL Server login is distinct from the operating system login for the machine, and logging into the machine does not imply that you are authenticated for the SQL Server instance as well.

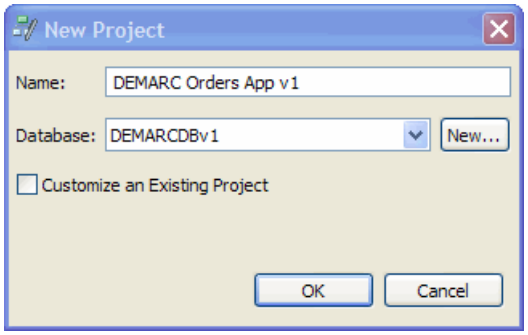
For DB2

An example for DB2, you are prompted for the database.



8 Click **Finish** to close the New Connection window.

9 Click **OK** to save and close the New Project window.



The Designer main window appears and you are ready to begin creating your definitions. For a general overview of Designer, refer to Designer overview (page 11).

Creating a model

Unlike a schema-based cartridge, a model-based approach enables you to capture the relationships among the tables. While adding tables is more complex than in a schema-based cartridge, building a data model has significant advantages. It simplifies the management of eligibility rules and the upload process. The first step is to create a data model that defines the tables and their relationships to one another.

For more information about schema-based cartridges, refer to Chapter 7, Creating a schema-based cartridge (optional).

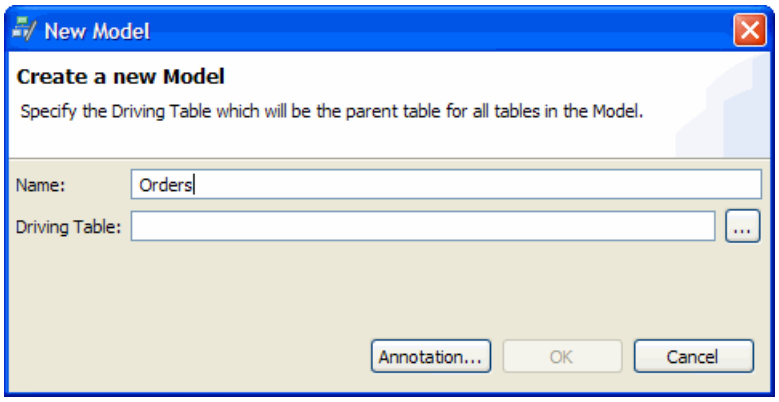
This section includes:

Creating a new model

1 Right-click the **Models** folder in the Project Navigator and select **New Model** from the pop-up menu to create a model.

TIP You can also start creating a new model by selecting **File > New Model**. Or, you can click the Create a New Model icon.

2 In the dialog box, type **`Orders`** for the name.

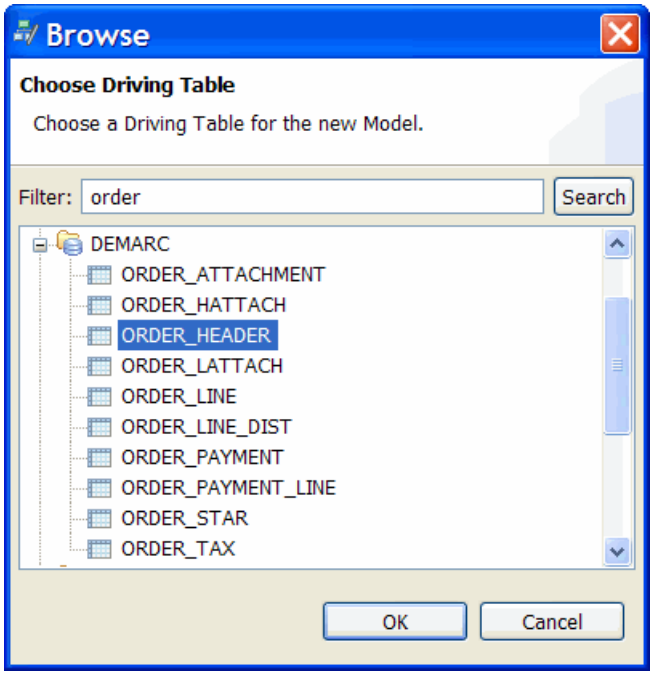


3 Click **Browse** (...) next to the Driving Table field. The Browse dialog appears allowing you to find the driving table. A driving table is the main table that drives the transaction you are extracting. Most driving tables (with a normalized schema) have subordinate tables.

4 Type **`order`** in the Filter Text field and click **Search**.

For Oracle

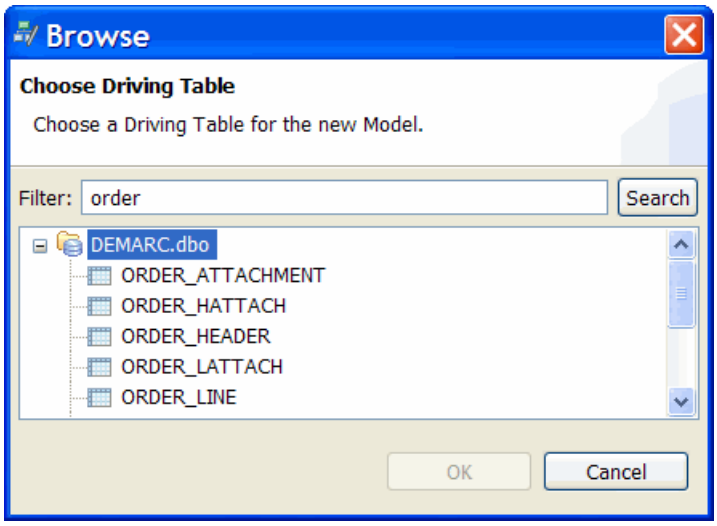
An example for Oracle, your results should be similar to these:



TIP For Oracle, if there are other schemas with table names starting with "order", you can type DEMARC.ORD to retrieve only the tables in the DEMARC schema containing 'ORD' in the name.

For SQL Server

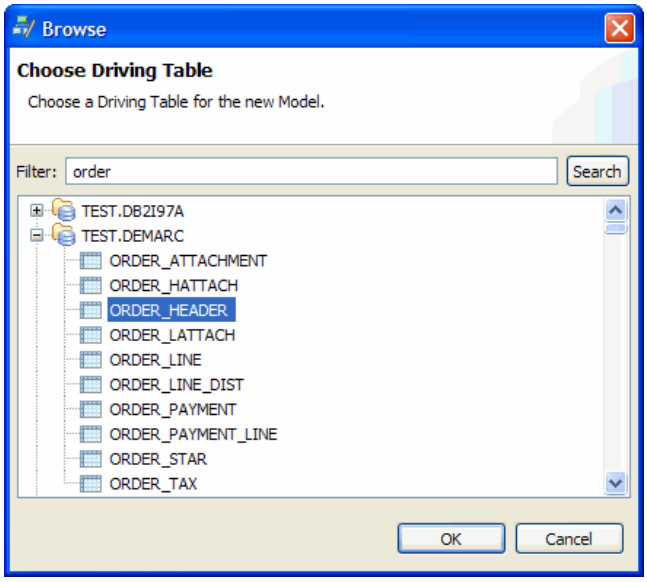
An example for SQL Server, your results should be similar to these:



TIP For SQL Server, if there are other schemas with table names starting with "order", you can type DEMARC.dbo.ORD to retrieve only the tables in the DEMARC schema containing 'ORD' in the name.

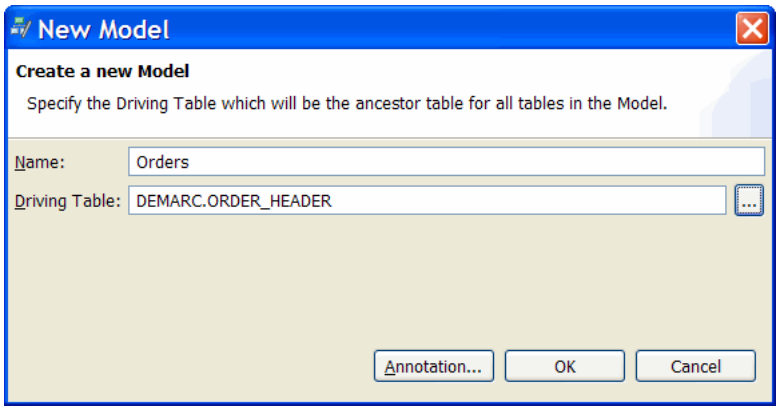
For DB2

An example for DB2, your results should be similar to these:



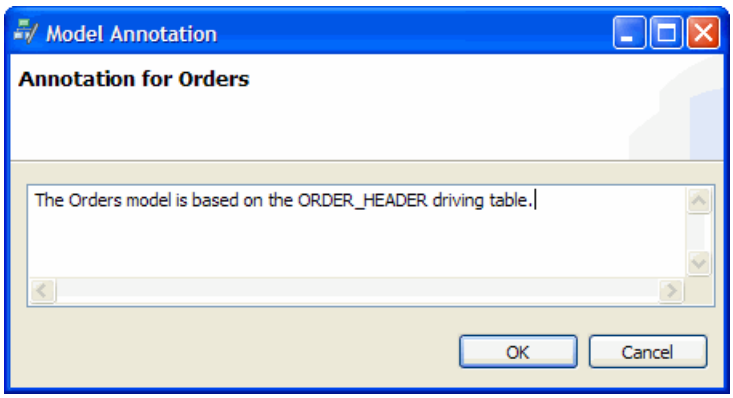
Notice how all the tables containing the search string appear in the list.

5 Double-click the **ORDER_HEADER** table under the appropriate schema name for your database to continue.



6 Click **Annotation**.

7 Type the following comment in the dialog box: **`The Orders model is based on the ORDER_HEADER driving table.`**

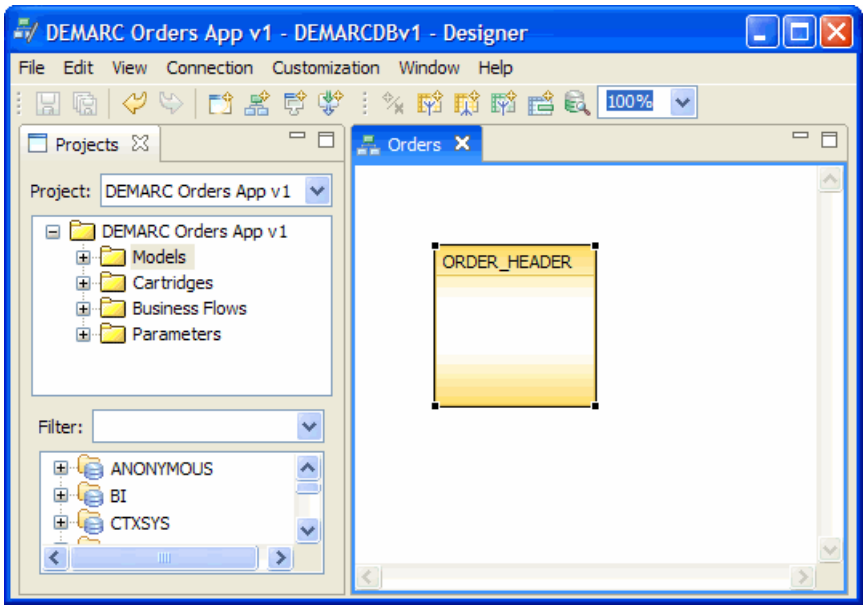


8 Click **OK** to close the Annotation dialog box.

TIP The Annotation button appears in many places within the Designer interface. Wherever it appears, you can use it to attach comments to that specific object.

9 Click **OK** to create a model.

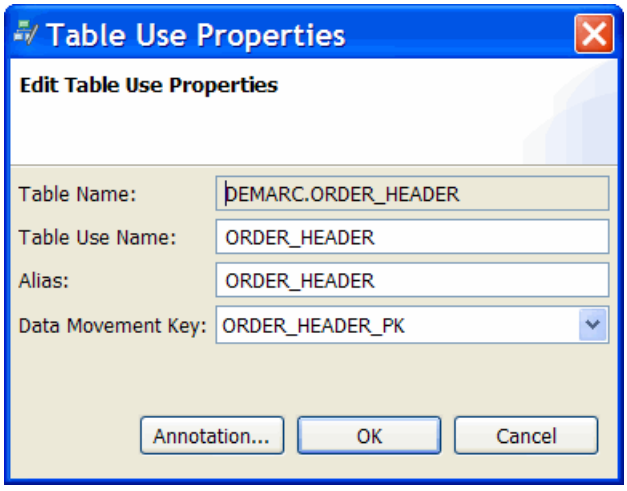
The Orders model editor appears with a simple graphical representation of the ORDER_HEADER table.



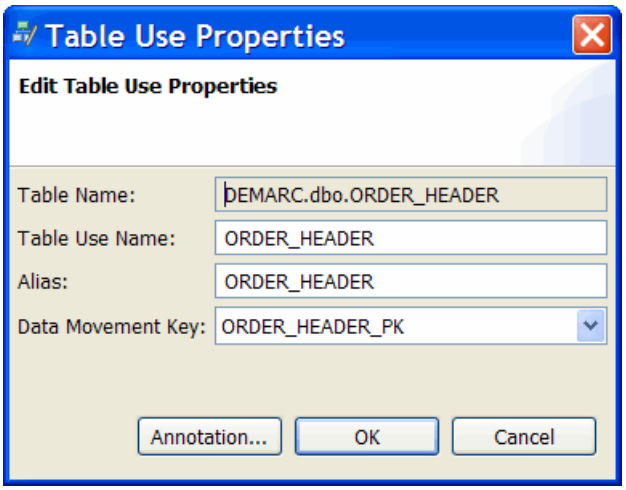
Viewing model properties

1 Double-click the **ORDER_HEADER** table in the main window to see the properties of the table.

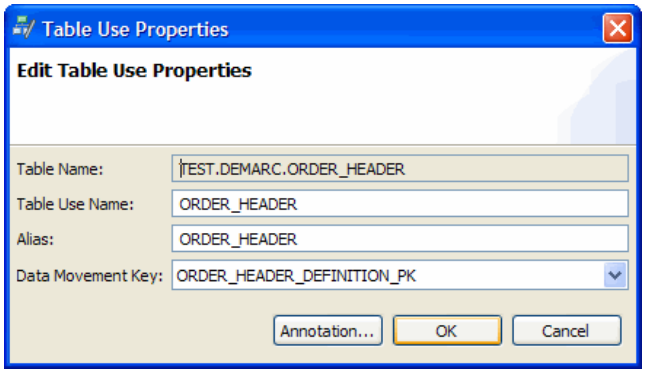
For Oracle An example for Oracle:



For SQL Server An example for SQL Server:



For DB2 An example for DB2:



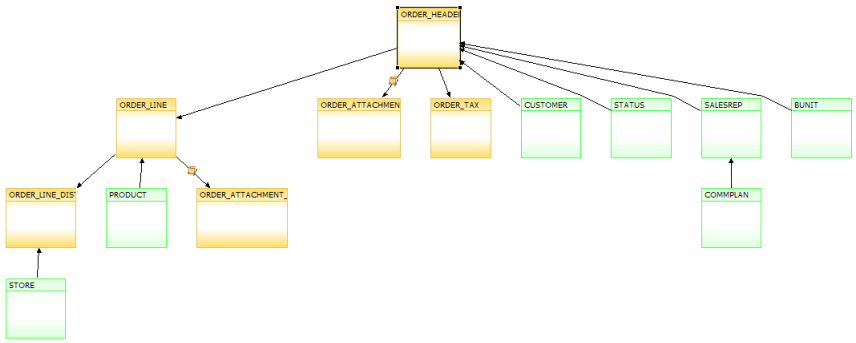
2 Click **OK**.

Adding tables to the model

Once you have created your model, you can begin to add tables to it. The process of adding tables varies depending upon the type of table:

- **Transactional tables** contain information about the business transaction. For example, a transactional table might contain detailed tax or payment information related to each business transaction.
- **Lookup tables** contain helpful, nontransaction information. You might need these lookup values present for the purposes of a rule or for the sake of making an extract file more complete. For example, nontransaction information could be status definitions, or the names of the sales representatives.

The steps of this section take you through adding the various types of tables. At the end, your model will look something like this example:

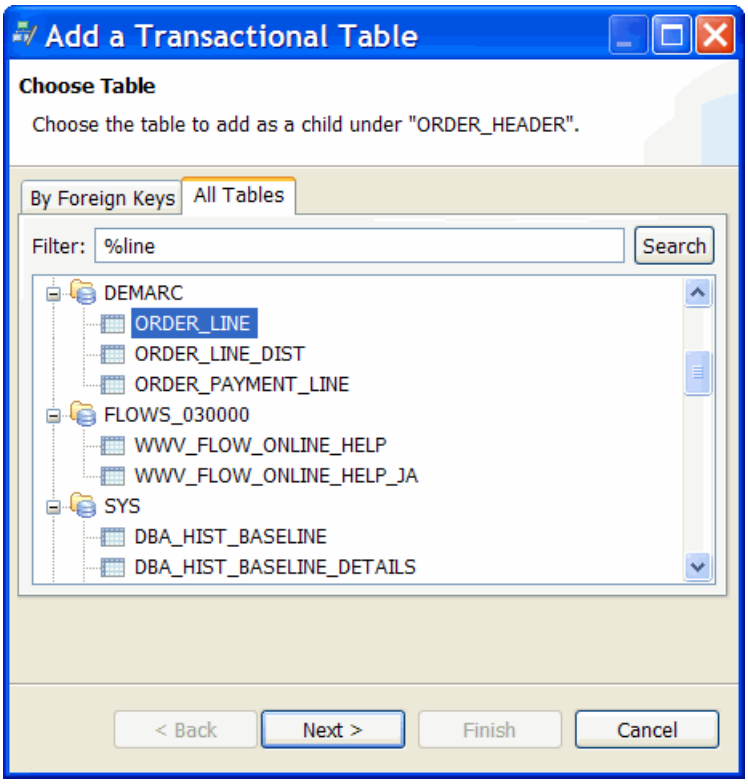


- Adding transactional and lookup tables (page 29)
- Adding multiple table uses with condition relationships (page 36)
- Completing the Orders data model (page 39)

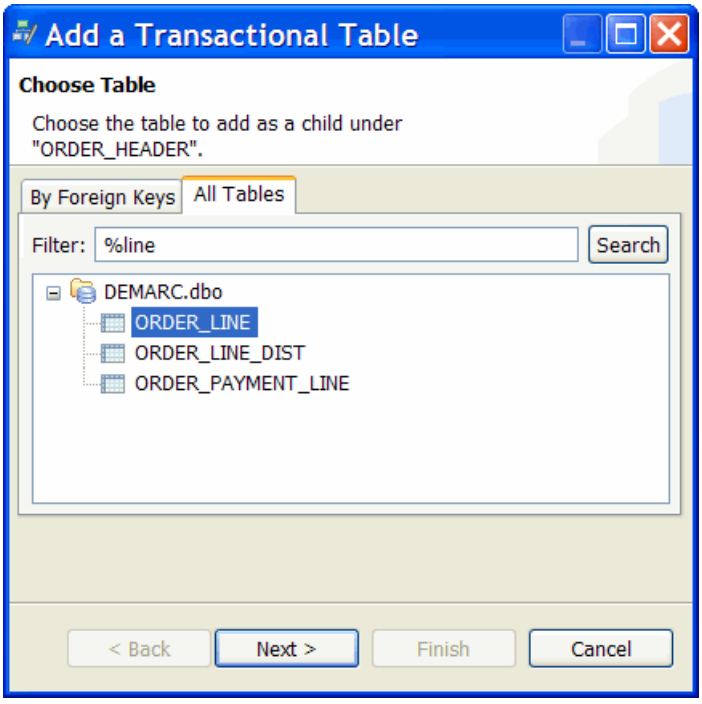
Adding transactional and lookup tables

1. Right-click the driving table and select **Add Transactional Table** from the pop-up menu. The Add Transactional Table wizard appears to help you add a subordinate table to the model.
2. Select the **All Tables** tab, if not already selected.
3. Type **`%line`** in the Filter field and click **Search**.
4. Select the **ORDER_LINE** table under the appropriate schema for your database.

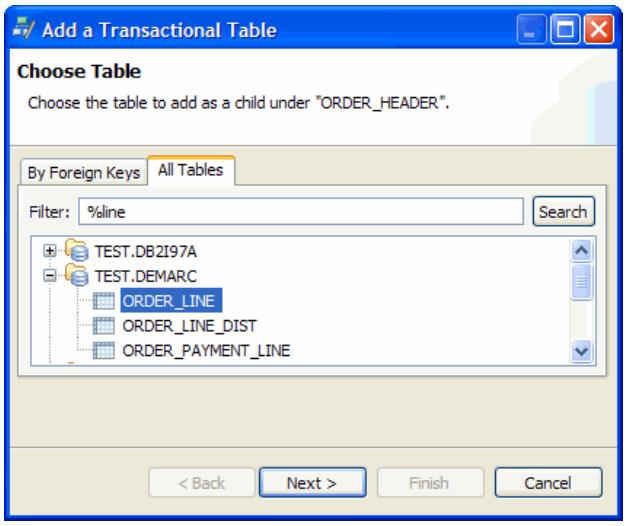
For Oracle An example for Oracle:



For SQL Server An example for SQL Server:



For DB2 An example for DB2:



5 Click **Next**.

When specifying a model, you must explain to Designer how the tables relate to each other. Because this sample schema contains primary keys but not foreign keys, you must indicate how those tables are related.

6 In the Specify How to Use or Build Keys page of the wizard, you can:

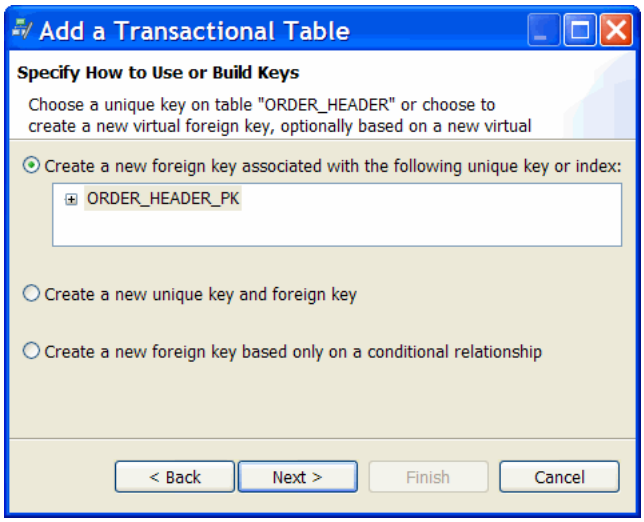
— **Create a new foreign key associated with the following unique key or index**. Designer discovers any existing unique keys that could serve as a virtual foreign key. Choosing this radio button enables you to choose from these discovered keys. This virtual foreign key is not created in the database, but it complements the knowledge Designer holds about your table relationships.

In this case, Designer found an existing primary key defined on ORDER_HEADER (ORDER_HEADER_PK). Choose this radio button and expand ORDER_HEADER_PK to see what columns participate in that key (only ORDERID in this instance). For the purposes of this tutorial, attach your virtual foreign key to the ORDER_HEADER_PK primary key.

— **Create a new unique key and foreign key**. Choosing this radio button indicates that you wish to create a new virtual unique key on ORDER_HEADER and attach a new virtual foreign key to it. These keys are not created in the database. They are virtual keys employed only in Designer to relate the tables.

— **Create a new foreign key based only on a conditional relationship**. Choosing this radio button indicates that you wish to create a foreign key against a conditional relationship.

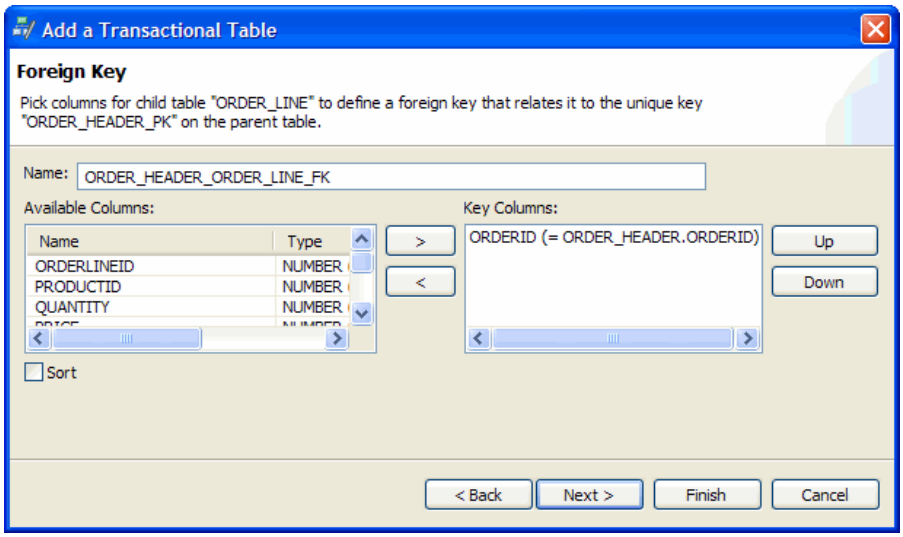
If you have gone through this process before on this table, Designer remembers the previously specified key and this dialog may appear differently. Ensure you make the selection described in this step for the purposes of this example.



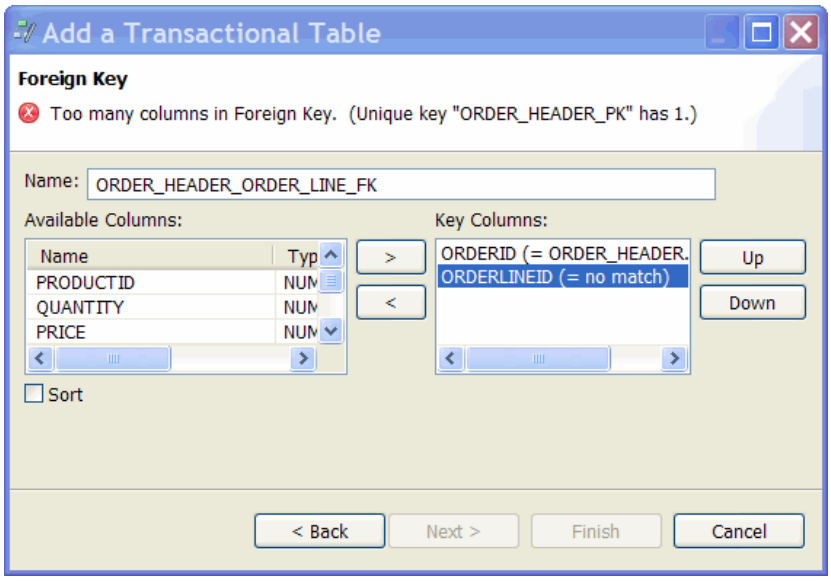
7 Click **Next**.

The Foreign Key page appears showing the available columns in the child table (ORDER_LINE) on the left. The list of columns to include in the virtual foreign key is on the right.

NOTE The wizard has already found the column to include by matching the column names in both tables.

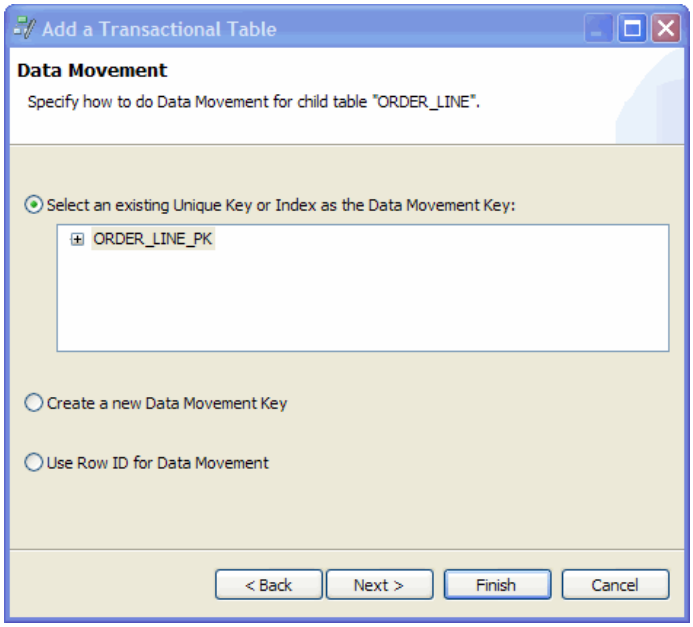


8 As an experiment, add another column using the arrow (>) or remove the ORDERID column. A message at the top of the dialog informs you the column count does not match, and both the Next and Finish buttons are disabled.



9 Undo whatever change you made such that the ORDERID column is the only Key Column and click **Next**. The Data Movement Key page of the wizard appears.

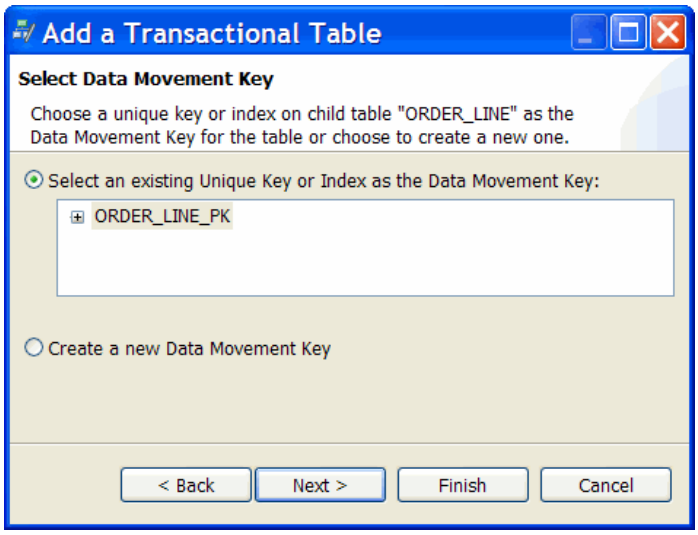
For Oracle An example for Oracle.



TIP While you can choose to use ROWID for data movement for Oracle, it is generally recommended that you use an existing key or index whenever possible.

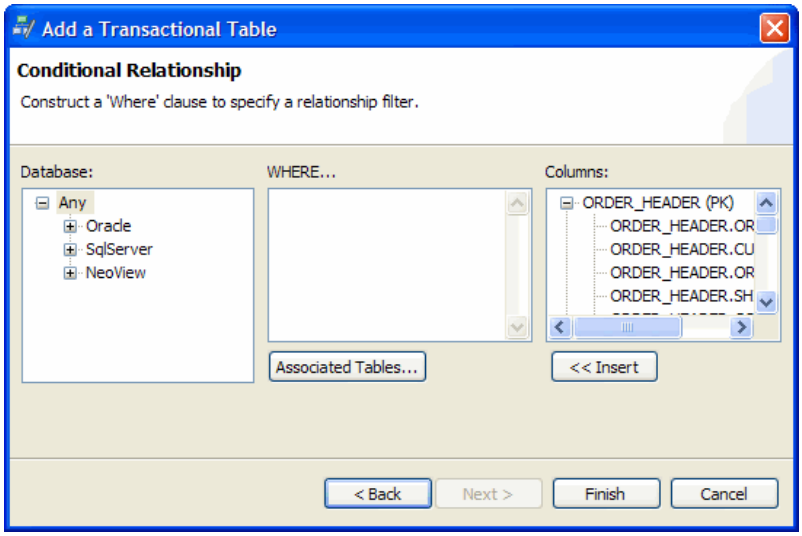
For SQL Server and DB2

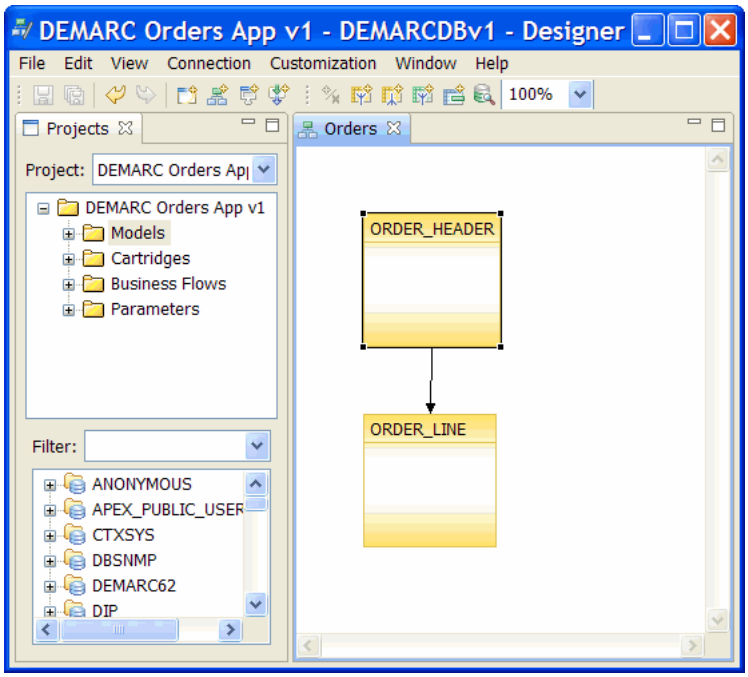
An example for SQL Server and DB2:



10 You can keep the default value on this page. Click **Next**.

11 The Conditional Relationship page enables you to specify a WHERE clause to define a relationship between the tables. In this case, you do not need a conditional relationship. Click **Finish** to return to the main Designer window.





12 Right-click the **ORDER_LINE** table and select **Add Transactional Table** from the pop-up menu.

TIP Another way to add tables to the model is to drag them from the Database Navigator pane and drop them on the object in the model to which they are related. For example, you can click and drag ORDER_LINE from the Database Navigator and drop it on the ORDER_HEADER table. If you add the table this way, you are prompted to define how to add the table (as transactional, lookup, or chaining table) and then you skip to step b.

---

a On the Choose Table page of the wizard, use the All Tables tab to find and select **ORDER_LINE_DIST**.

b Accept the default values by clicking **Next** until Finish is enabled. Click **Finish**.

13 Right-click **ORDER_LINE_DIST** and select **Add Lookup Table** from the pop-up menu.

c On the Choose Table page of the wizard, use the All Tables tab to find and select **STORE**.

d Accept the default values by clicking **Next** until Finish is enabled. Click **Finish**.

14 Right-click **ORDER_LINE** and select **Add Lookup Table** from the pop-up menu.

a On the Choose Table page of the wizard, use the All Tables tab to find and select **PRODUCT**.

b Accept the default values by clicking **Next** until Finish is enabled. Click **Finish**.

Adding multiple table uses with condition relationships

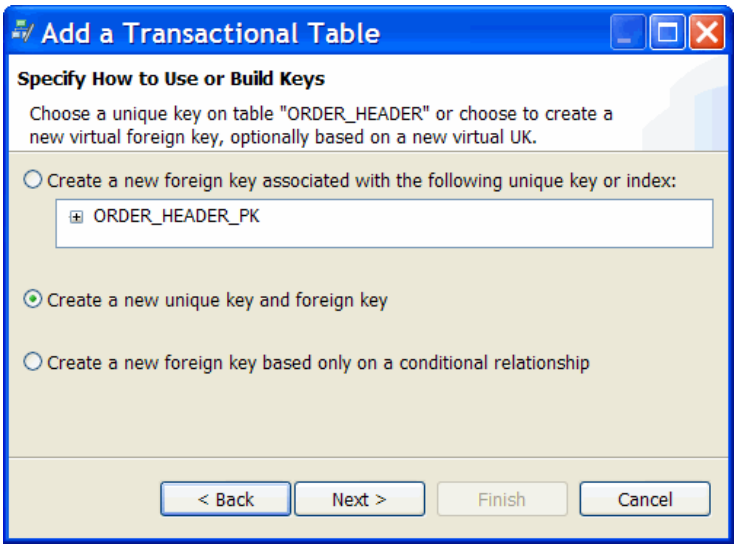
In some cases, you may need to use the same table more than once in your data model in order to meet the requirements of your application. This concept is known as multiple table use. When you have multiple table uses in your model, you need to ensure that the rows retrieved for each use are disjoint. If not, you may end up with duplicate data, which might break your application.

To guarantee the rows retrieved for each use of a table remain disjoint from one another, you can apply a conditional relationship when adding each table use.

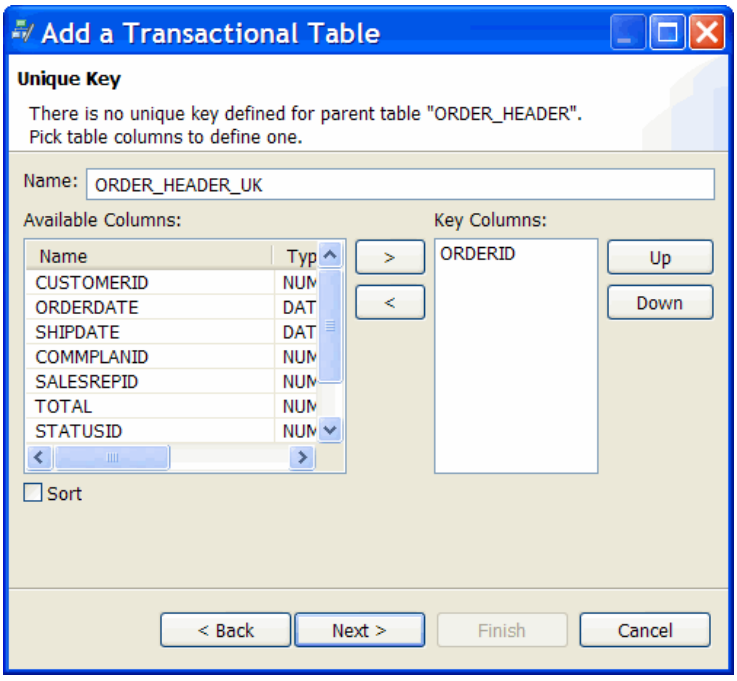
To add a table more than once with conditional relationships:

1. Right-click the **ORDER_HEADER** table and select **Add Transactional Table** from the pop-up menu.
   a. On the Choose Table page of the wizard, use the All Tables tab to find and select **ORDER_ATTACHMENT**.
   b. Click **Next**. ORDER_ATTACHMENT is required twice in the model, once here and then again as a table use related to ORDER_LINE. Hence, it requires both a foreign key and conditional relationship.

      A conditional relationship is one where a value in a parent table can be referenced by multiple child tables, but the relation is exclusive and determined by a condition.
   c. Choose the **Create a new unique key and foreign key** radio button.

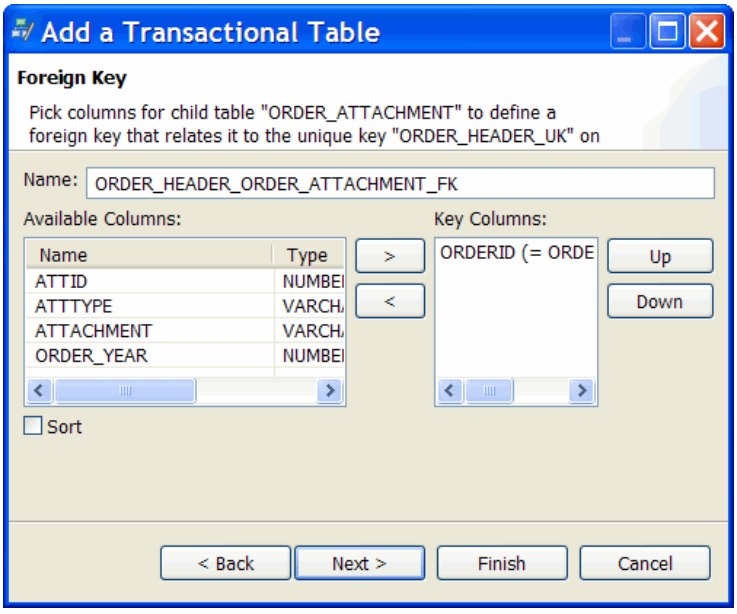


   d. Click **Next**. On the Unique Key page, you can see the available columns, which could be used for a unique key. Test Data Management may have detected some unique keys and placed them in the Key Columns list by default.
   e. If necessary, use the shuttle buttons (< and >) such that ORDERID is the only column in the Key Columns list.



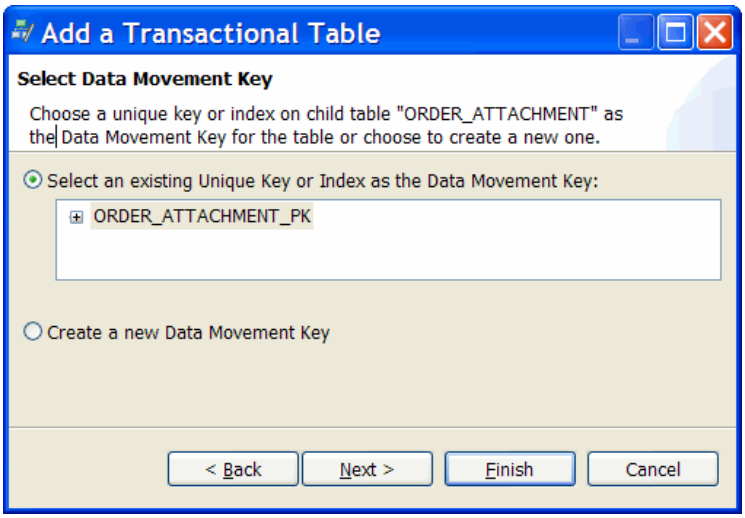
f Click **Next**.

On the Foreign Key page, ORDERID should be in the Key Columns list by default. If not, use the shuttle buttons (< and >) to move it to the Key Columns list.



g Click **Next**.

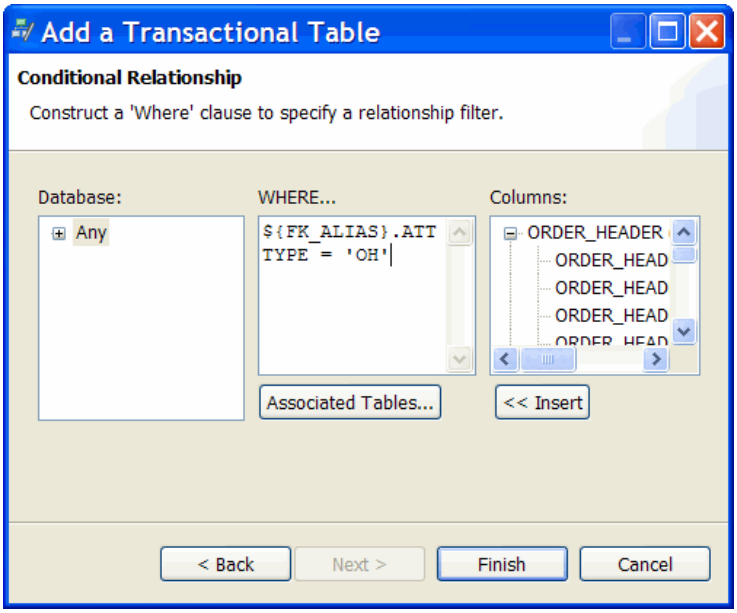
h The Data Movement Key page displays. Remember that the Oracle version of this page contains an option to use ROWID. In any case, you can just accept the default selection.



i Click **Next**.

j On the Conditional Relationship page, enter the following for the WHERE clause:

```
${FK_ALIAS}.ATTTYPE = 'OH'
```



k Click **Finish**.

Notice the icon on the link in the model to indicate that it is a conditional relationship. Now you will create the second table use of ORDER_ATTACHMENT with a different conditional relationship.

2 Right-click the **ORDER_LINE** table and select **Add Transactional Table** from the pop-up menu.

a On the Choose Table page of the wizard, use the All Tables tab to find and select **ORDER_ATTACHMENT**.

b Click **Next**.

c Choose the **Create a new unique key and foreign key** radio button.

d Click **Next**.

e On the Unique Key page, ORDERLINEID and ORDER_YEAR appear in the Key Columns list by default. Use the shuttle buttons (< and >) such that ORDERLINEID is the only column in the Key Columns list.

f Click **Next**.

On the Foreign Key page, you can see the available columns, which could be used for a foreign key. In this case, HP Test Data Management has placed no columns in the Key Columns list by default. Hence, you receive an error message at the top of the dialog indicating that you must select a foreign key.

g Use the shuttle buttons (< and >) such that ORDERID is the only column in the Key Columns list.

h Click **Next**.

i The Data Movement page displays. Click **Next** to accept the default selection.

j Enter the following for the WHERE clause:

```
${FK_ALIAS}.ATTTYPE = 'OL'
```

k Click **Finish**. Notice that this second use of the table is named ORDER_ATTACHMENT2 to distinguish it from the first use.

Completing the Orders data model

To flesh out the remainder of your model, you need to add the following tables:

1 Right-click **ORDER_HEADER** and select **Add Transactional Table** from the pop-up menu.

a On the Choose Table page of the wizard, use the All Tables tab to find and select **ORDER_TAX**.

b Accept the default values by clicking **Next** until Finish is enabled. Click **Finish**.

2 Right-click **ORDER_HEADER** and select **Add Lookup Table** from the pop-up menu.

a On the Choose Table page of the wizard, use the All Tables tab to find and select **CUSTOMER**.

b Accept the default values by clicking **Next** until Finish is enabled. Click **Finish**.

3 Right-click **ORDER_HEADER** and select **Add Lookup Table** from the pop-up menu.

a On the Choose Table page of the wizard, use the All Tables tab to find and select **STATUS**.

b Accept the default values by clicking **Next** until Finish is enabled. Click **Finish**.

4 Right-click **ORDER_HEADER** and select **Add Lookup Table** from the pop-up menu.

a On the Choose Table page of the wizard, use the All Tables tab to find and select **SALESREP**.

b Accept the default values by clicking **Next** until Finish is enabled. Click **Finish**.

5 Right-click **SALESREP** and select **Add Lookup Table** from the pop-up menu.

a On the Choose Table page of the wizard, use the All Tables tab to find and select **COMMPLAN**.

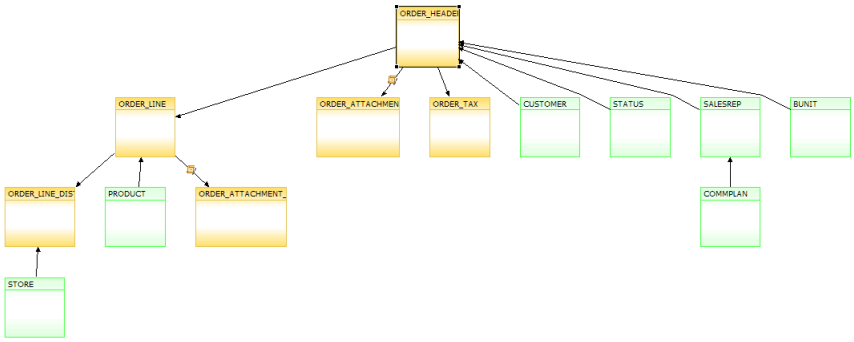
b Accept the default values by clicking **Next** until Finish is enabled. Click **Finish**.

6 Right-click **ORDER_HEADER** and select **Add Lookup Table** from the pop-up menu.

a On the Choose Table page of the wizard, use the All Tables tab to find and select **BUNIT**.

b Accept the default values by clicking **Next** until Finish is enabled. Click **Finish**.

Your model should now contain all these shapes and lines:



Advanced concept

The line colors are meaningful. Black means that the relationship has been validated. The relationship is checked whenever you open the project. The icons on the relationships with the ORDER_ATTACHMENT table uses indicate the presence of conditions.

If the lines and the boxes ever change to red, it means something has changed in the underlying database and the relationship is no longer valid. The reasons for this change can include:

- The database constraint has been dropped.
- You have deleted the virtual constraint.
- You have edited the database connection and are now connecting to a database where the objects no longer exist.
- The database is unavailable.

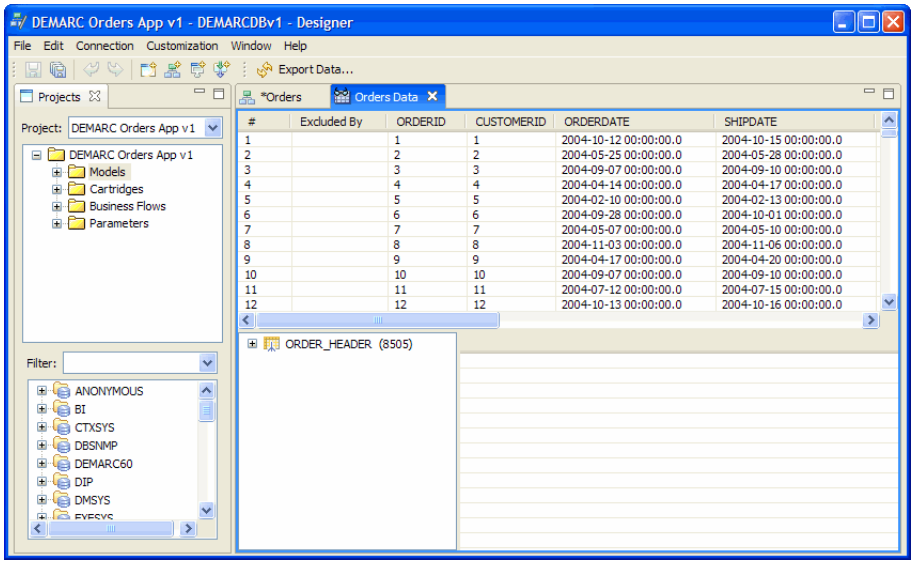
TIP Save your work. If you have not saved recently, click Save All in the toolbar to save what you have created to this point.

Previewing the data

You have taken care of the data structure. Take your first look at the data and consider what needs to be extracted.

1 Click **Preview** in the toolbar. If preview is disabled in the toolbar, click **ORDER_HEADER** in the model to enable it.

TIP Clicking the column headers sorts the data by that column's values. For example, click the ORDERDATE column header to sort the results by ORDERDATE. Click the ORDERID header to change to sort by ORDERID.



2 Scroll through the orders and notice that the ORDERIDs are consecutive numbers.

3 On the Orders Data tab, click the close icon (**X**) to close it and return to the model.

Defining a rule

With the model in place, you should next consider the scope of data to be extracted. Rules enable you to refine your model to reflect your requirements for extracting. You can define rules in any order, but, typically, your first rule defines the scope of your extraction operation. For example, in this case, the first rule specifies the age at which records become eligible for extraction.

Creating a rule

To begin defining your rule, you create it on the table to which it applies in your model:

1. In the Orders model, right-click the **ORDER_HEADER** shape and select **Add Rule** from the pop-up menu.
2. For Name, type **`Data from last X months`**.

Creating a parameter

In most cases, you should parameterize your rules such that you can adjust their behavior depending upon your latest business rules. For example, if you want to create a test data set that covers a longer period of time, you can change the time window of the extraction by modifying a parameter rather than rewriting your rules.

1. In the Rule dialog box, click **Parameters** to add a parameter to represent X in the rule.
2. Click **Add**.
3. For Name, type **`Months_to_Extract`**.
4. Click **OK**.
5. Enter or select the parameter properties in Table 3.

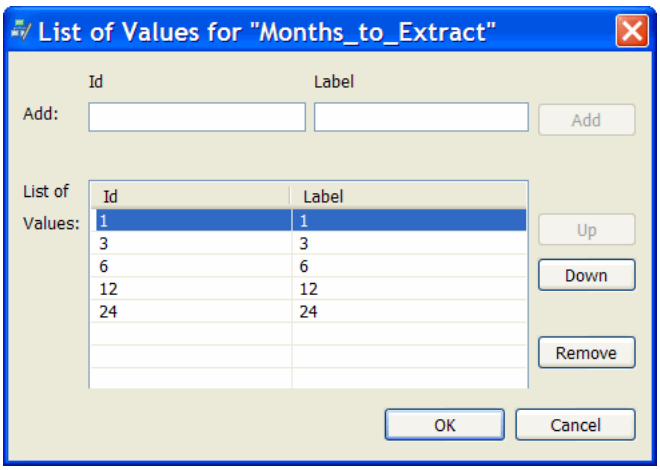
Table 3 Months_to_Extract parameter properties

Parameter name	Parameter settings
Months_to_Extract	Label: **`Months to extract`** Data Type: **Number** The parameter type must be a Number in order to compare it to other numeric values. Length: **`10`** Default: **`24`** (months) By default, orders placed in the last two years will be extracted for the subset. Validation: **Mandatory** List of Values: **Static**

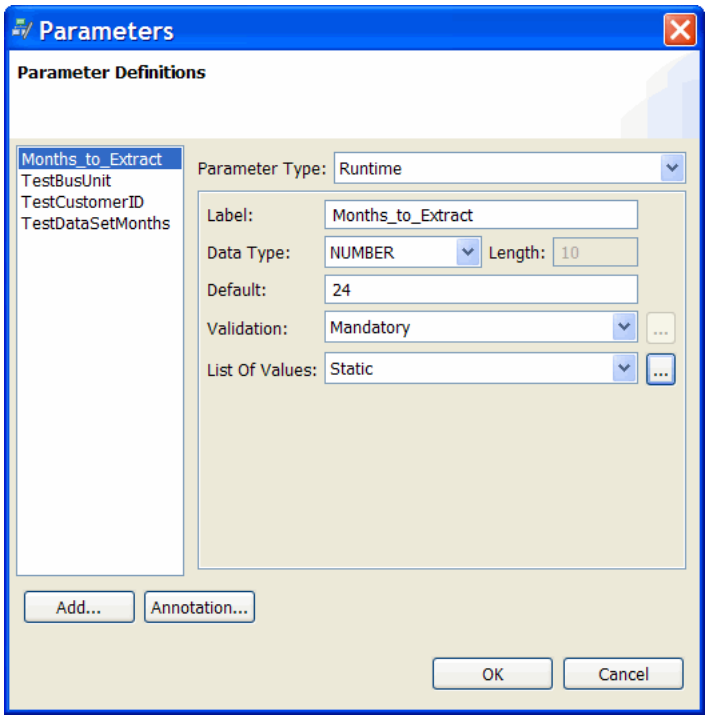
6. Click the **Browse** button to the right of List of Values.
7. Next to Add, type **`24`** for Id and **`24`** for Label.

8 Click the **Add** button.

9 Repeat step 7 and step 8 for **1**, **3**, **6**, and **12**. Use the Up and Down buttons to put the numbers in order.



10 Click **OK**.



11 Click **OK** to accept the parameter definition and return to the Rule dialog box.

Specifying rule properties

To complete your rule, specify its properties:

1 For Customization, choose **Mandatory**.

Advanced concept

The Customization property indicates whether and how the rule can be altered by a user to whom you distribute the project for customization. Mandatory rules cannot be changed or deleted by such a user. Refer to the *HP Test Data Management Designer User Guide* for more information about customization.

2 Under Database, expand the **Any** node, then select the node for your database.

3 Select the **Oracle** node and type the following WHERE clause:

```
"ORDER_HEADER"."ORDERDATE">add_months(sysdate,
-(:Months_to_Extract))
```

This clause will extract orders that were entered within the last number of months specified by the Months_to_Extract parameter.

TIP If you type in a parameter reference in your WHERE clause, for example, **`:Months_to_Extract`**, without first defining it, Designer automatically creates it for you. You would then need to open the parameter definition by clicking **Parameters** and change its properties as necessary, for example, setting its type, length, and default value.

4 Select the **SQL Server** node and enter this WHERE clause:

```
"ORDER_HEADER"."ORDERDATE">dateadd(M,
-cast((:Months_to_Extract) as int), getdate())
```

5 Select the **DB2** node and enter this WHERE clause:

```
"ORDER_HEADER"."ORDERDATE" > (CURRENT DATE -
:Months_to_Extract MONTHS)
```

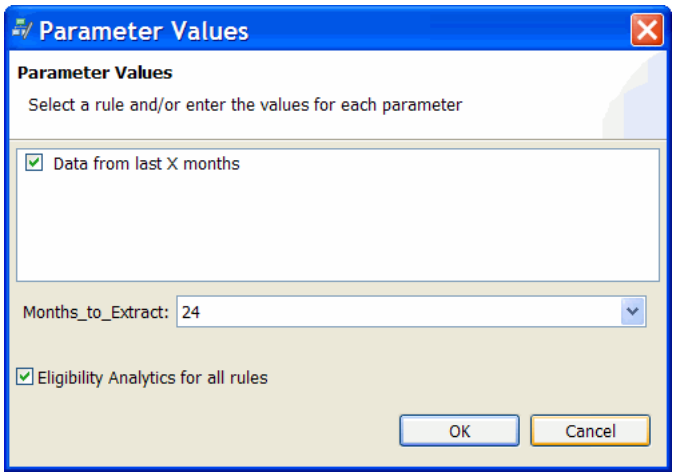
6 Click **OK** to close the Rule dialog and return to the model.

7 Click **Preview** to check how this change affects the data.

8 Designer prompts you with a Parameter Values dialog box, where you select which, if any, rules to apply. In this case, you have only one rule and it is already selected by default.

The dialog box also includes a field for you to provide a value for the Months_to_Extract parameter. For the purposes of this tutorial, leave the default value, **`24`**, and click **OK**.

NOTE For more information about eligibility analytics, refer to *HP Test Data Management Concepts Guide*.



You have now restricted the extract data to be orders placed during the last two years from today. Rows excluded by the rule appear in red and the name of the rule appears in the Excluded By column.

Do not close the preview tab yet.

TIP Save your work. If you have not saved recently, click Save All in the toolbar to save what you have created to this point.

Optional exercise

If you want to experiment further with preview, click the Refresh tool in the toolbar. This time, though, choose a different value for Months_to_Extract, for example, **6**. In the preview tab, some records that had been included when the value was 24 months are now excluded at 6 months. It may help to see this effect if you click the ORDERDATE column to order the rows by ORDERDATE.

Creating a subset by customer

Selecting the order data by customer is another method of subsetting that could complement subsetting by date. For example, if you know that some group of customers was very active in the last two years, you might want to select just those customers for your subset.

To add a rule that selects records based upon the customer:

1. If it is not already open, double-click the Orders model in the Project Navigator to open it for editing.
2. Right-click the CUSTOMER table and choose **Add Rule**.
3. For Name, type **`Extract_by_customer`**.
4. Click **Parameters** to add a parameter to represent the customer identifier in the rule.
5. To add a parameter, perform the following steps:
   a. Click **Add**.
   b. For Name, type **`Customer1`**, then click **OK**.

c Enter or select the parameter properties in Table 4.

Table 4 Cusomter1 parameter properties

Parameter name	Parameter settings
Customer1	Label: **`Customer 1 (required)`** Data Type: **Number** The parameter type must be a Number in order to compare it to other numeric values. Length: **`10`** Validation: **Mandatory** List of Values: **SQL**

d Click the **Browse** button to the right of List of Values. By using SQL to create a list of values, you can select only those customers who had orders during the time window specified by the Months_to_Extract rule.

e Select the **Drop-Down List** radio button.

f Under Database, expand the **Any** node.

g Select the **Oracle** node and enter the following SELECT statement.

```
SELECT CUSTOMER.CUSTOMERID, CUSTOMER.LASTNAME
   || ' ' || CUSTOMER.FIRSTNAME
FROM DEMARC.CUSTOMER
WHERE CUSTOMER.CUSTOMERID IN (
    SELECT ORDER_HEADER.CUSTOMERID
    FROM DEMARC.ORDER_HEADER
    WHERE ORDER_HEADER.ORDERDATE < add_months(sysdate,
    -(:Months_to_Extract))
    )
ORDER BY 2
```

Notice how the SELECT returns two values, the CUSTOMERID, which is the value actually used by the rule, and the customer's last and first names concatenated together, which is what you see in the list of values for ease of use. Furthermore, the SELECT contains a subquery that bounds the returned values by the date value entered for the Months_to_Extract parameter. This prevents you from choosing a customer who had no orders during the period of time for which you are extracting data.

h Select the **SQL Server** node and enter the following SELECT statement.

```
SELECT CUSTOMER.CUSTOMERID, CUSTOMER.LASTNAME
   + ' ' + CUSTOMER.FIRSTNAME a
FROM DEMARC.dbo.CUSTOMER
WHERE CUSTOMER.CUSTOMERID IN (
    SELECT ORDER_HEADER.CUSTOMERID
    FROM DEMARC.dbo.ORDER_HEADER
    WHERE ORDER_HEADER.ORDERDATE > dateadd(M,
-cast((:Months_to_Extract) as int), getdate())
    )
```

```
ORDER BY a
```

i. If you are connected to Oracle or SQL Server, you can select that node and click **Validate** to confirm your syntax. For DB2, you need to validate your syntax yourself.

j. Select the **DB2** node and enter the following SELECT statement.

```
SELECT CUSTOMER.CUSTOMERID, CUSTOMER.LASTNAME
   || ' ' || CUSTOMER.FIRSTNAME
FROM DEMARC.CUSTOMER CUSTOMER
WHERE CUSTOMER.CUSTOMERID IN (
 SELECT ORDER_HEADER.CUSTOMERID
 FROM DEMARC.ORDER_HEADER ORDER_HEADER
 WHERE ORDER_HEADER.ORDERDATE < add_months(sysdate,
  -(:Months_to_Extract))
 )
ORDER BY 2
```

k. Click **OK**.

6. Repeat step a through step k to create another parameter, **Customer2** with the following changes:

   — Choose **None** for Validation. Because you only require the first parameter, the others are optional. You do not need to validate any additional parameters.

   — Leave Default blank.

   — Because it is not required, append (optional) to the Label for this parameter. The user only needs to choose a value for the first parameter. The others can be left blank.

7. Optionally, repeat step step 6 as many times as you wish to enable you to retrieve the records for as many different customers as you like.

8. Click **OK**.

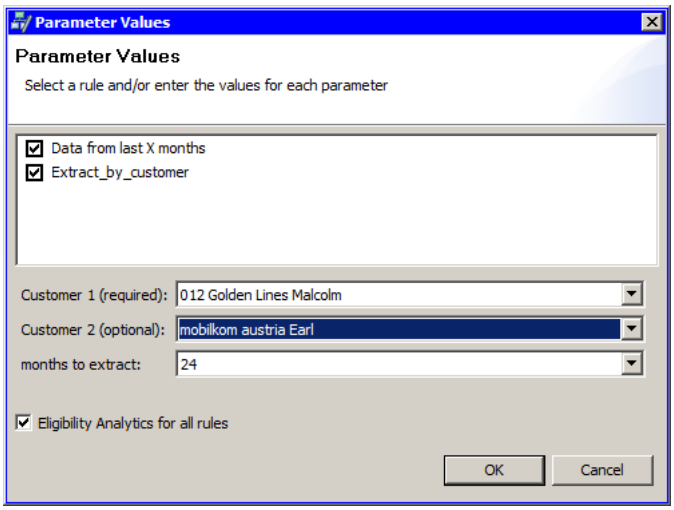
9. Under Database, select the **Any** node.

10. Type a WHERE clause similar to the following but including only customer parameters that you created in the parentheses:

```
"CUSTOMER"."CUSTOMERID" IN (:Customer1, :Customer2)
```

11. Click **OK**.

12. Click **Preview** to check how this rule affects the data selection. When the Parameter Values dialog displays, only the two new rules should be selected. Note that you must choose a value for Customer 1 (required) in order to proceed.



13 Select one of the customers from the list and click **OK**.

14 Repeat step 13 until all of the customer parameters you created have values.

15 When you have values for all of the customer parameters, click **OK**. Review the data and see what would be included and excluded from extraction by your rules.

For illustrative purposes, two rules are sufficient, but you could, of course, continue adding more rules depending upon your goal for the subset. Once you have the rules that you want and you have tested them with preview to ensure that they work as expected, you can create a cartridge, which will perform the actual extraction for you.

Defining a preview-only rule

You may have noticed in the earlier sections that previewing large amounts of data can take some time. When in the development phase for your extraction solution, a quick response time is helpful because you are frequently making model enhancements and previewing them. Limiting the data returned for testing purposes is often worthwhile. A preview rule enables you to define a rule that applies only in preview in Designer. The rule is not applied when you generate and deploy your cartridge for actual usage.

TIP Before you create a preview-only rule to limit the number of rows returned, you must consider the minimum amount of data that you need to adequately test your extraction solution. If you return too little data, you may not be able to completely evaluate the effect of your model and rules. You may need to iterate a few times to strike the proper balance between performance and testing needs.

1 Right-click the **ORDER_HEADER table** in the model and select **Add Rule** from the pop-up menu.

2 For Name, type **`Limit preview results`**.

3 For Usage, select **Preview**.

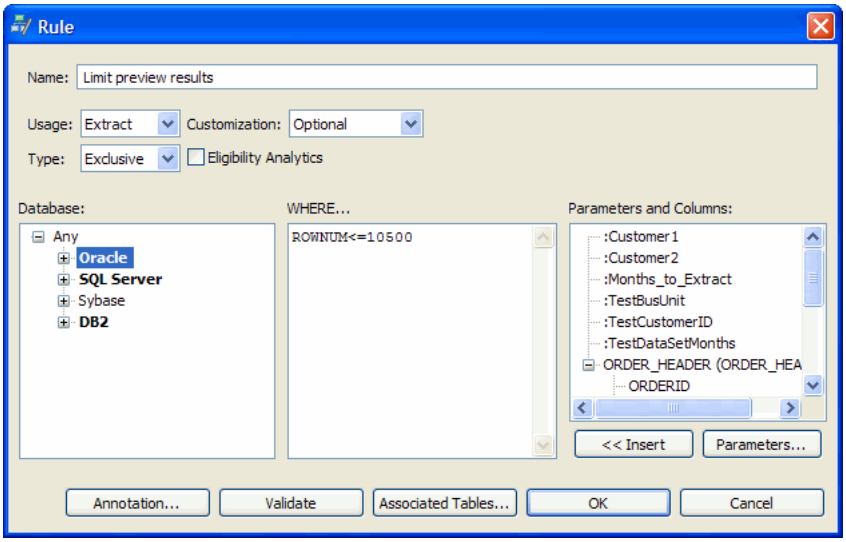
4 For Customization, choose **Optional**.

5 Under Database, expand the **Any** node.

6 Select the **Oracle** node.

7 Click in the **WHERE...** field.

8 Type **`ROWNUM<=10500`**.



9 If you are connected to an Oracle database, click **Validate** to confirm the WHERE clause is syntactically correct.

10 Click **OK**, to close the Validation dialog. If the statement failed validation, check for typing errors.

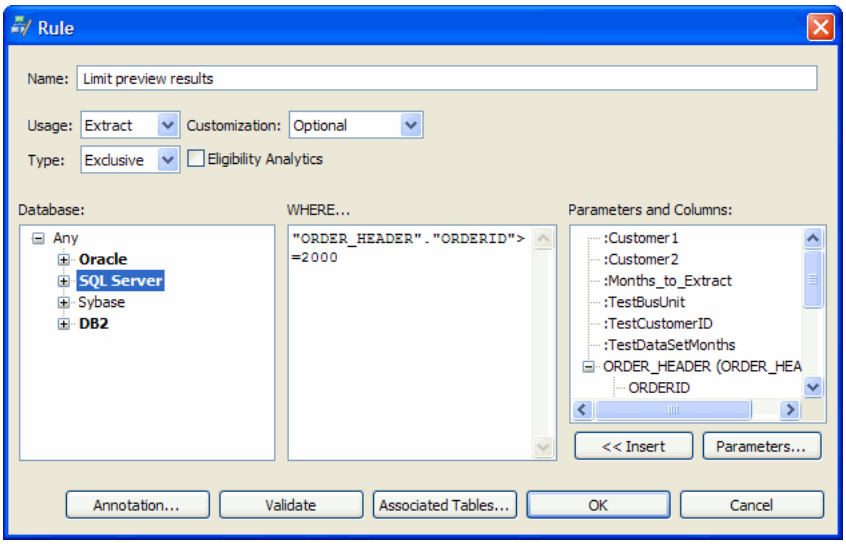
11 Select the **SQL Server** node.

12 Click in the **WHERE...** field.

13 Under the ORDER_HEADER node, double-click **ORDERID**.

14 Type **`>=2000`**. The entire WHERE clause is now:

```
"ORDER_HEADER"."ORDERID">=2000
```



15 If you are connected to a SQL Server database, click **Validate** to confirm the WHERE clause is syntactically correct.

16 Click **OK,** to close the Validation dialog. If the statement failed validation, check for typing errors.

17 Select the **DB2** node.

18 Click in the **WHERE...** field.

19 Under the ORDER_HEADER node, double-click **ORDERID**.

20 Type **>=2000**. The entire WHERE clause is now:

```
"ORDER_HEADER"."ORDERID">=2000
```

21 Click **Preview** in the toolbar. Fewer rows are returned in the Preview tab.

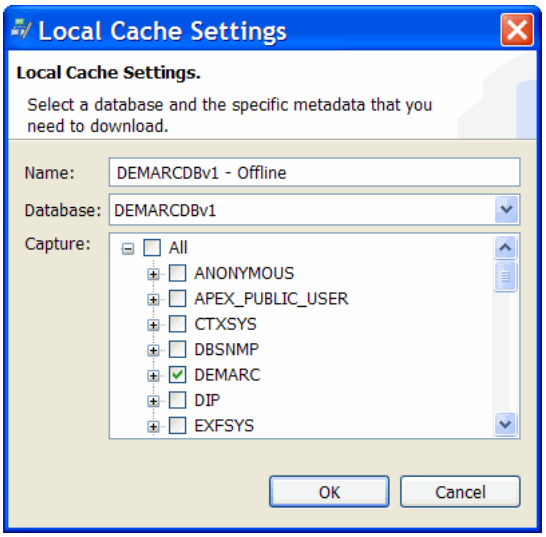
TIP Save your work. If you have not saved recently, click Save All in the toolbar to save what you have created to this point.

Working offline (optional)

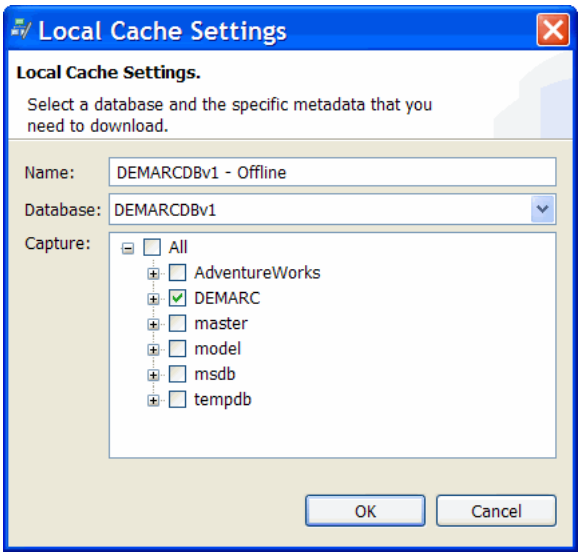
By creating a local copy of the metadata, you can work in a disconnected mode from the database. This feature is useful if, for example, you are working against a schema with many database objects. Working offline in this case can speed up the interaction. Furthermore, it also means you can continue to work when you are without access to the database over the network. You should note that you cannot preview data when working offline because only metadata is stored in the local cache.

1 Select **Connection > Work Offline** from the menu. The first time you make this choice, the following dialog appears. If you get an intermediate dialog listing existing local cache definitions, select **New** and click **Next**.

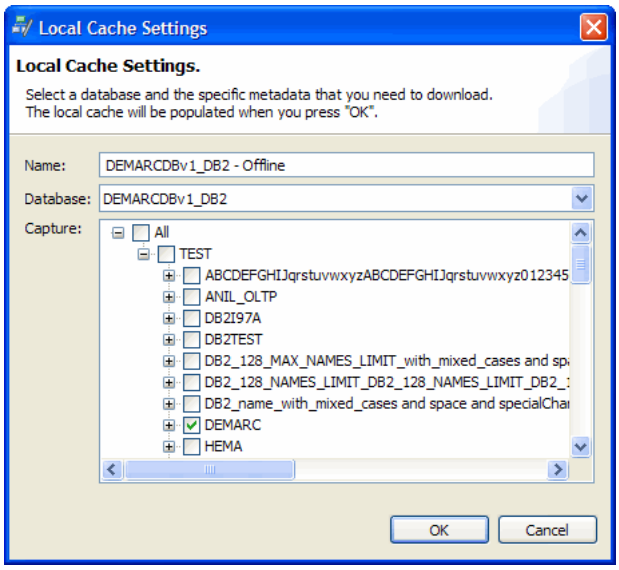
For Oracle



For SQL Server



For DB2 An example for DB2



2 From this list, you can select one or more schemas and database object definitions to be captured in the local cache. Ensure only the DEMARC box is checked and click **OK**. You are now working offline from a locally cached copy of the DEMARC schema.

TIP Be sure to scroll through the entire list to ensure that only DEMARC is selected for the local cache capture.

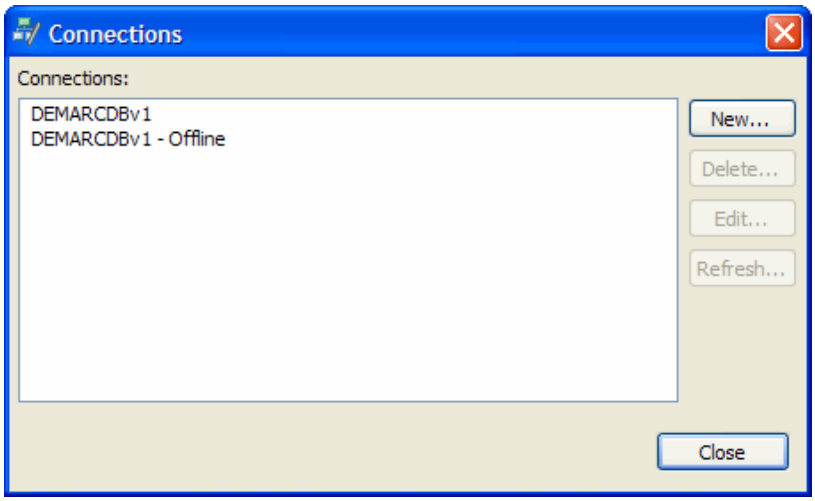
Notice how the Database Navigator (bottom left corner) has changed to only show the cached data objects.

3 To return to online mode, select **Connection > Work Offline** again. Designer reconnects to the database containing the currently open model. You can tell by the change in the Database Navigator in the lower left corner that you are now reconnected to the database. You should see a great many more objects there when connected to the database.

Managing connections (optional)

For the purposes of this tutorial, you only need one database connection. In the real world, you are likely to have many different database connections from which to choose. You can manage the cache and database connection definitions in Designer through the Connection menu.

1 Select **Connection > Edit Connections** from the menu. The Connections dialog appears:



2 Using the buttons in this dialog, you can:

- refresh a database or local cache
- edit the connection definitions
- edit the content of the caches
- delete both the connections and caches
- create new connections

In this case, just refresh the cache. Select **DEMARCDBv1 – Offline**, which you created in the previous section.

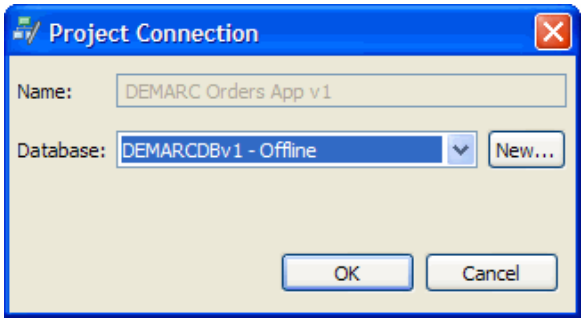
3 Click **Refresh**.

4 Click **Close**.

TIP Each project is associated with a particular connection by default. It can be offline or online. This association persists with the project after it is saved and is restored when the project is opened again.

Changing the connection for the project

1 Select **Connection > Project Connection**.



2 Select **DEMARCDBv1 - Offline** in the Database list of values.

3 Click **OK**.

4 For the next chapter, return to using the online database by choosing **Connection > Work Offline**.

Creating a model-based cartridge

A cartridge is the mechanism by which HP Test Data Management specifies a versioned instance of an extract model for one type of extraction. Among other things, you select the following in your cartridge definition:

- Which model to use
- Which rules to apply
- Data masking

This section includes:

Creating a cartridge

To create a model-based cartridge:

1 Go to **File > New Cartridge**.

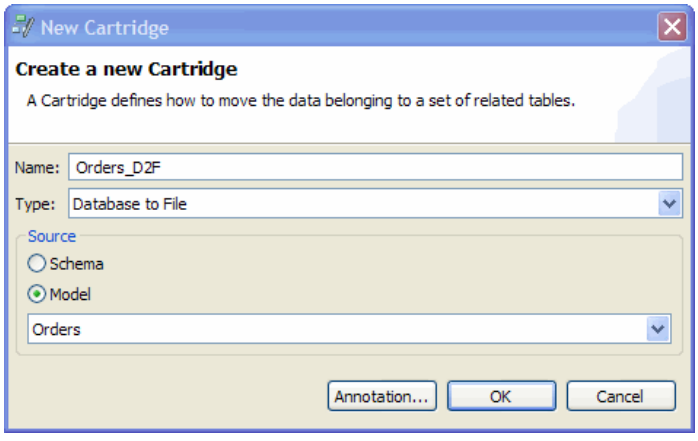
2 In the New Cartridge dialog box, type **`Orders_Model_D2F`** as the Name.

3 If it is not already selected, select the **Model** radio button.

Related information

HP Test Data Management Designer User Guide.

4 If it is not already selected, select **Orders** as the model.



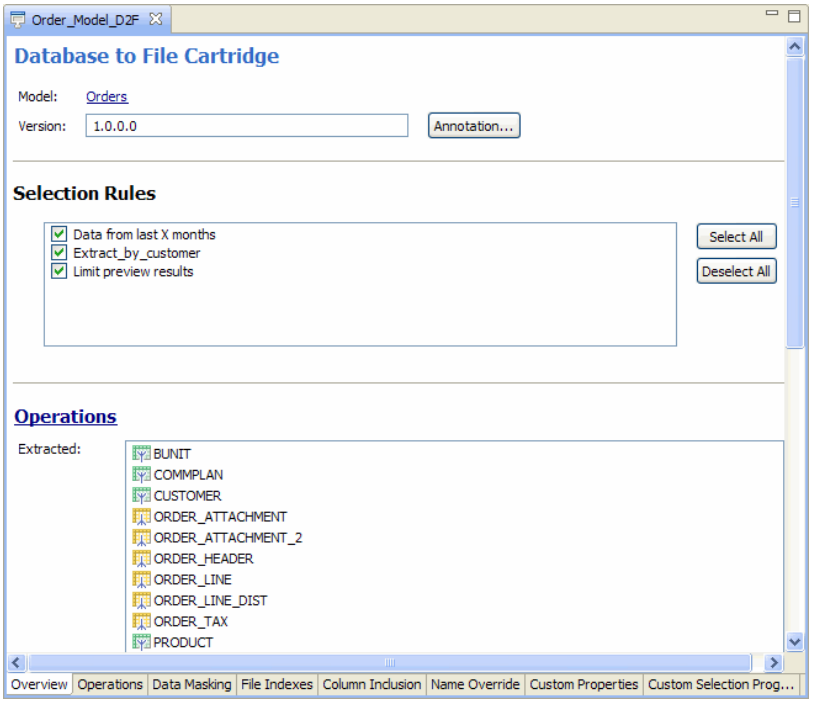
5 Click **OK**. The Database to File Cartridge editor appears.

TIP For information about navigating within the editor, refer to Navigating in the cartridge editor (page 55).

6 Apply data masks as described in Applying data masks (page 55).

Navigating in the cartridge editor

When you first open the cartridge editor, no tables are yet selected for extraction. If you look carefully at the bottom of the editor, a number of tabs are displayed, which correspond to the different parts of the cartridge you can edit. The first tab, Overview, is an overview of the cartridge. Each section on the page also has a title that acts as a hyperlink to the corresponding tab.



At the top of each page, you will find a link called **Back to Overview**, which returns you to the Overview page. Of course, you can also return to Overview by clicking the **Overview** tab.

Any change you make in any page is immediately reflected in the Overview page.

Applying data masks

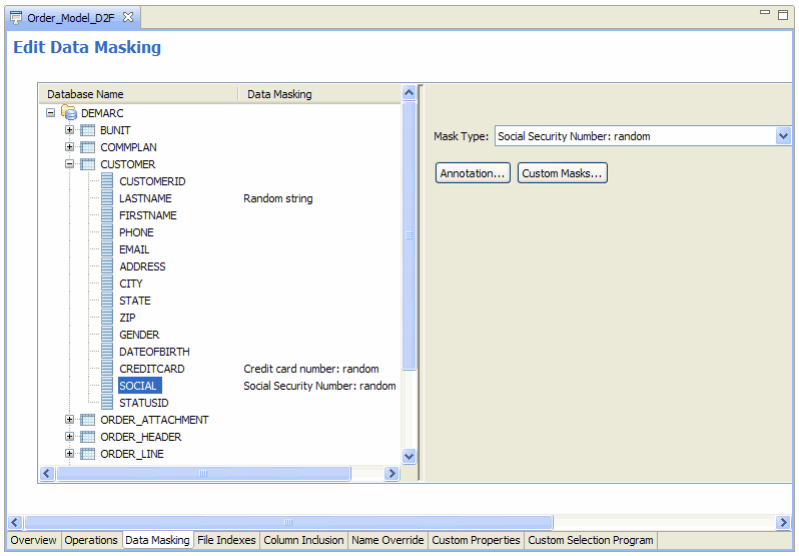
Data masking of a column means HP Test Data Management obfuscates the data values in the column. When creating test data from a production database, you should mask any sensitive data, such as credit card numbers, names, addresses, phone numbers, and so on. HP Test Data Management provides the ability to apply pre-defined masks or custom masks to your test data in order to preserve privacy and security. For purposes of this tutorial, use this feature for the CUSTOMER.CREDITCARD, CUSTOMER.SOCIAL, and CUSTOMER.LASTNAME columns.

To apply masks to columns:

1. Click the **Data Masking** title or tab.
2. Select the **CUSTOMER** table on the left and expand it.
3. Select the **LASTNAME** column.
4. In the Mask Type list of values, choose **Random string**.
5. Select the **CREDITCARD** column.
6. In the Mask Type list of values, choose **Credit card number: random**. This option will substitute random numbers in the CREDITCARD column.

> TIP The values in this column must be in a valid credit card format. If not, you either need to correct the column values or use a string mask instead. Otherwise, the data will not be masked in the test data. Refer to the *HP Test Data Management Designer User Guide* for more information.

7. Select the **SOCIAL** column.
8. In the Mask Type list of values, choose **Social Security Number: random**. This option will substitute random numbers for the numbers in the SOCIAL column.



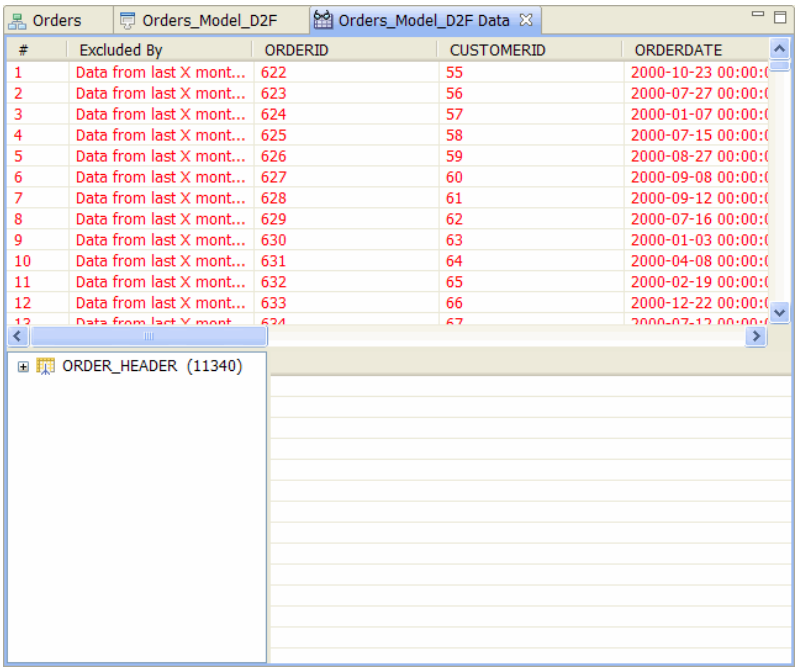
9. Click **Back to Overview** to return to the Overview page.

Previewing the cartridge

Unlike schema-based cartridges, you can preview a model-based cartridge.You can preview your cartridge in the same way you previewed your model to confirm it is behaving as you intended.

1. In the toolbar, click the **Preview** tool.
2. In the Parameter Values dialog box, enter parameter values and click **OK**. Notice the differences between this dialog and the one that you saw when you previewed from the model. Because the cartridge definition specifies which

rules to apply or not apply, this Parameter Values dialog box does not provide check boxes for all available parameters. Rather it provides just one check box to enable you to apply or ignore the preview rule.



3 Review the data in the Preview tab.

- The top part of the window shows the rows of the driving table. Select a row or range of rows in the top part of the window to filter the rows displayed in the bottom part. Use Ctl-click to select more than one row or clear the rows selected.
- The Excluded By column displays the rule that caused a row to be excluded. All rows that are excluded are displayed in red.
- Click on column headers to sort the rows by that value. For example, if you click the Excluded By column header, the rows are sorted according to the values of that column.
- Click and drag the column borders to resize the columns in the display.
- The model structure, including rules, is displayed in the lower left pane. Expand and collapse the node to view the tables and rule you want. Positive numbers in parentheses next to the driving table indicate the number of driving table rows included by that table. Negative numbers in parentheses next to a rule indicate the number of driving table rows excluded by the rule.

TIP Expand the ORDER_HEADER node in the lower left pane and select the CUSTOMER table. Notice that the SOCIAL and CREDITCARD column values are not masked in preview even though you specified masking in the cartridge. Masking is applied at runtime. It is not applied during Designer preview.

If you notice any problems in preview, you can return to the cartridge definition, modify it, and preview the results again by clicking the **Preview** tool.

4 When you are finished with the Preview tab, close it.

Summary and next steps

In this chapter you learned to:

- Create a model
- Create parameters
- Create rules
- Preview data
- Manage your connections (switch between connections, work offline)
- Create a model-based cartridge
- Edit cartridge properties in the editor
- Preview the data for your cartridge

The next step is to create a business flow for your cartridge.

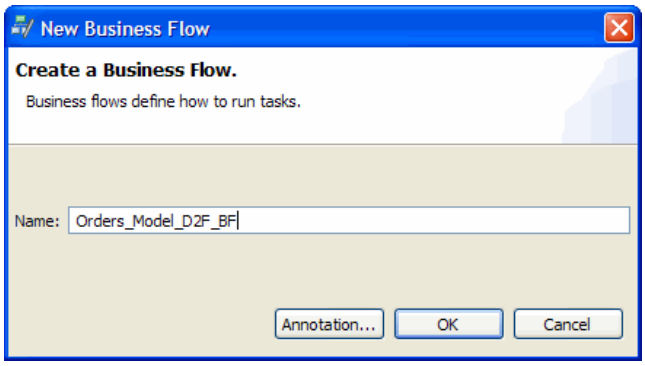
4 Creating a business flow

You can run your cartridge separately or as part of a larger workflow. For example, you can run a preprocessing script, run your cartridge, and then run a postprocessing script. You can go one step further and split the execution of your cartridge into steps and perform some scripted operations in the middle of the cartridge execution. With HP Test Data Management, you can create business flows to achieve this kind of workflow.

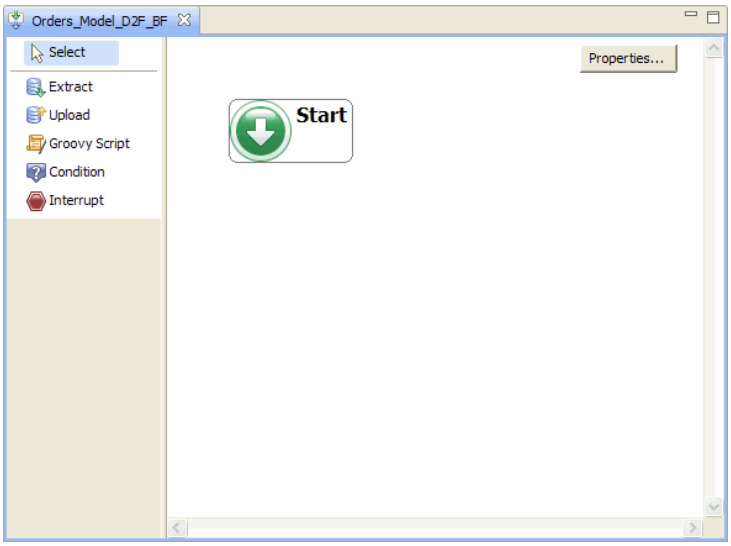
This chapter includes:

Creating a business flow

1. Select **File > New Business Flow**.
2. Type **`Orders_Model_D2F_BF`** for the Name.

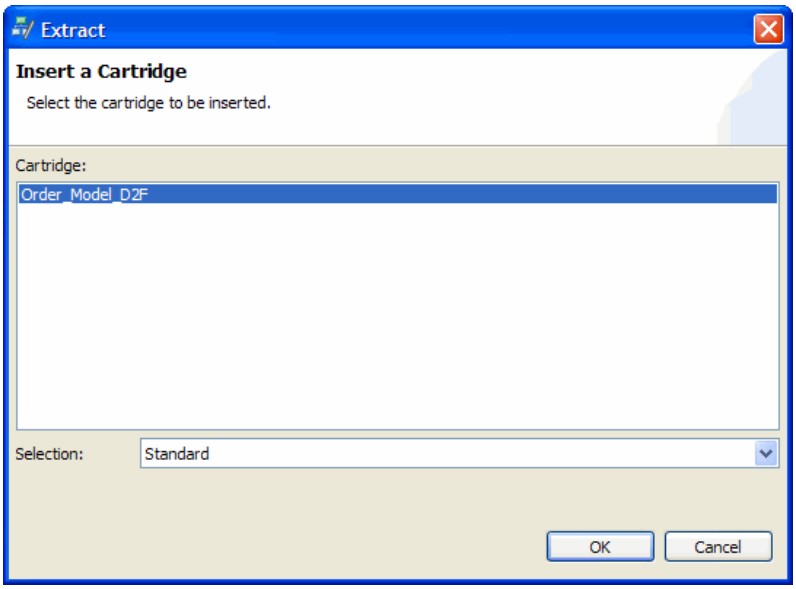


3. Click **OK** to create the business flow. The business flow always starts from the Start activity. The left side of the editor contains a toolbar with the various objects you can insert into the business flow.

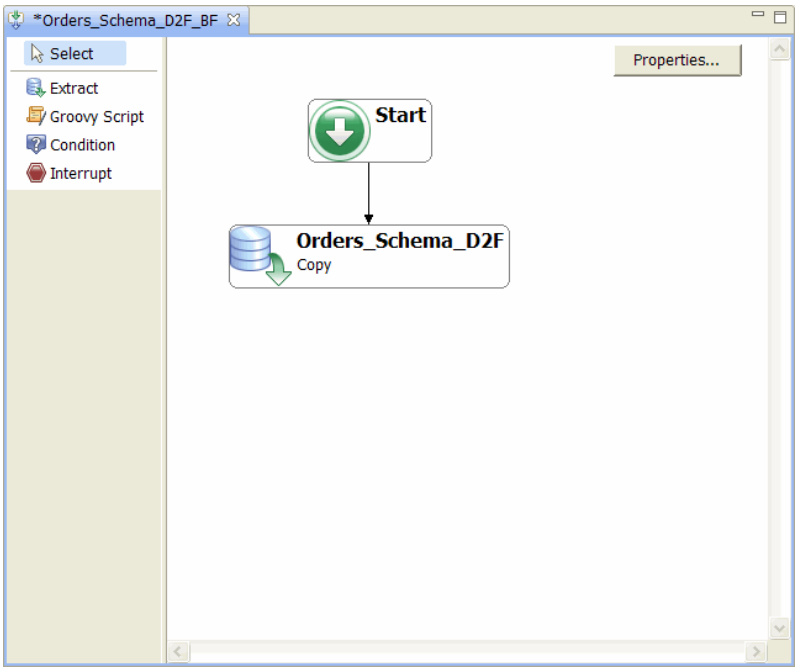


4 Click **Extract** and then click under the Start activity to place it.

5 In the Extract dialog, select **`Orders_Model_D2F`** in the Cartridge field, if not already selected.



6 Click **OK**. You now have a business flow with one cartridge.

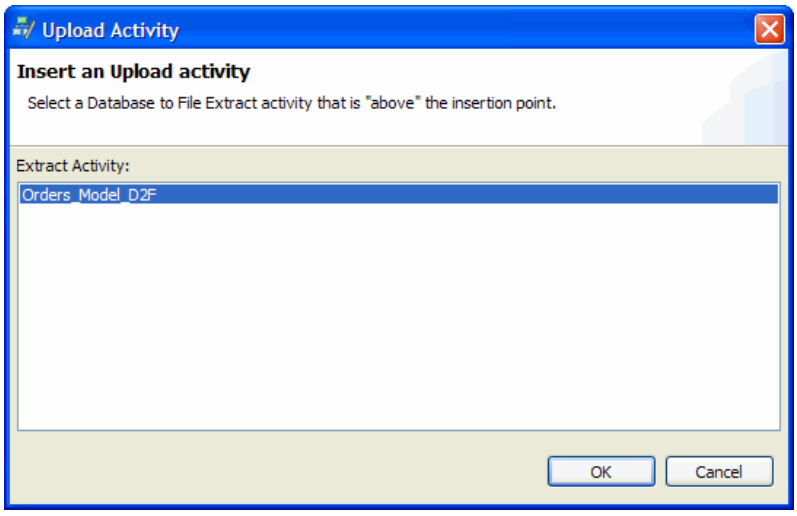


Adding an upload activity (optional)

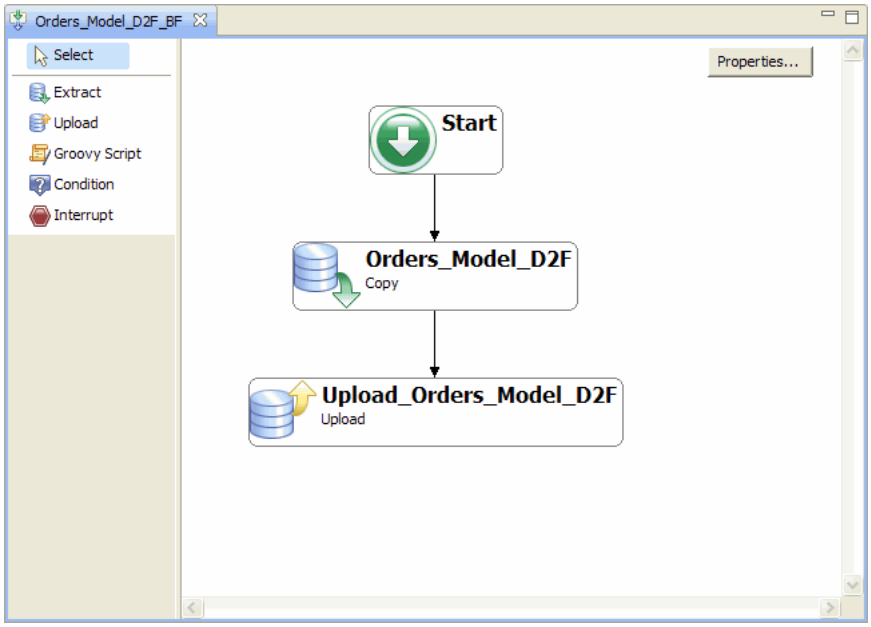
If you want to immediately upload the extracted data to your test database, you may want to include an upload activity in your business flow after extraction activity. When you run the business flow, it prompts you to specify a destination database and it will upload to that database as soon as the extraction successfully completes.

To add an upload activity:

1. Click **Upload** and then click under the extraction cartridge activity to place it.
2. In the Upload Activity dialog, select **`Orders_Model_D2F`** in the Extract Activity field, if not already selected.



3 Click **OK**. You now have an upload activity following your extraction cartridge.



Summary and next steps

In this chapter you learned to:

- Create a business flow
- Add a cartridge to the business flow

The next step is to deploy and run this business flow, which also deploys and runs your cartridge.

5 Deploying a business flow

When the business flow definition is complete, you are ready to deploy it to the local or remote system where you plan to execute it. Alternatively, you could also generate it on the file system for future deployment on another system by you or someone else.

This chapter describes how to set up the deployment environment, deploy and run a business flow in the environment, and monitor the business flow while it is running.

This chapter includes:

Deployment prerequisites

Before you deploy your business flow, you must perform the following tasks in the Web Console:

- Start the Web Console service.
- Invoke the Web Console URL from your browser.
- Install the repository.
- Create a deployment environment.
- (Optionally) Create users.

See also

For detailed information on how to perform these tasks in the Web Console, refer to *HP Test Data Management Web Console and Query Server User Guide*. In particular, look for the tutorial about configuring the Web Console.

Deploying the business flow

In order to run your business flow, you must first deploy it.

NOTE Before performing the steps in this section, you must meet all of the prerequisites in Deployment prerequisites (page 63).

To deploy your business flow:

1 Return to Designer or restart it if it is not currently open.

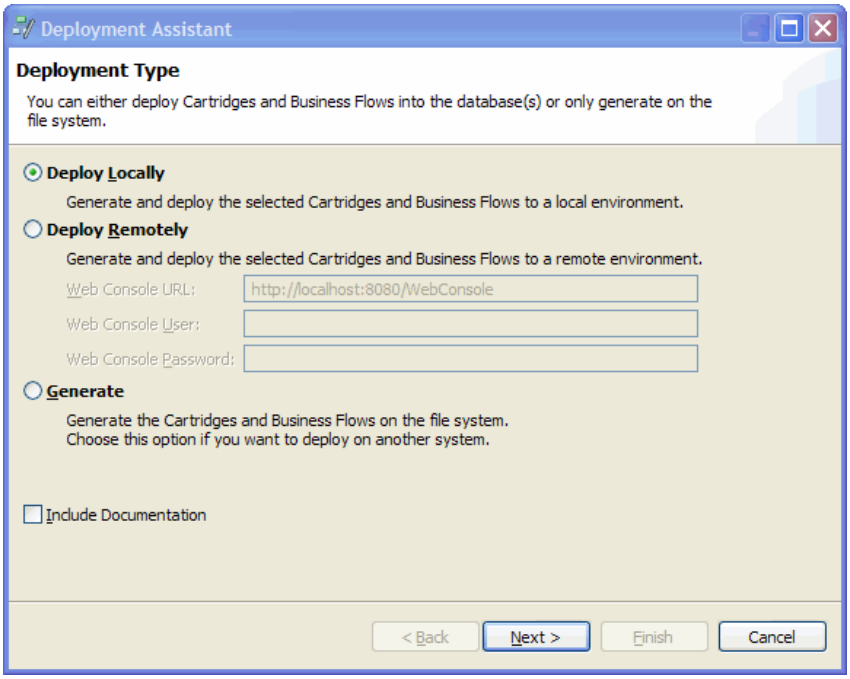
2 In the Project Navigator, right-click the **Orders_Model_D2F_BF** and select **Deploy** from the pop-up menu.

TIP In the Deployment Assistant on the Deployment Type page, you can select Deploy Locally, if you installed the repository on the same database server where you are currently running Designer. If you do not have the repository installed on the same database server where you are running Designer, you must choose Deploy Remotely or Generate. Refer to the *HP Test Data Management Designer User Guide* for more information.

---

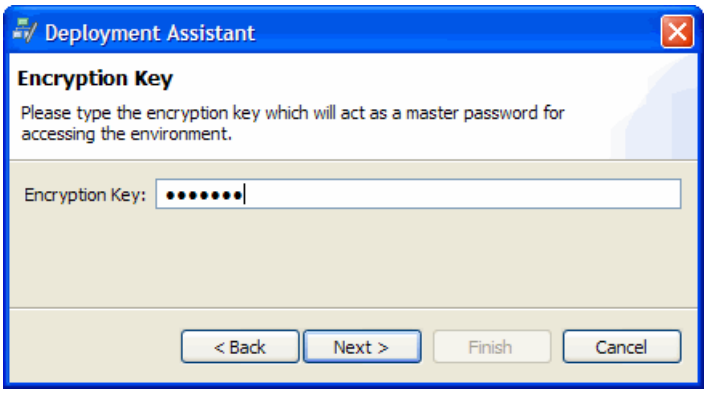
3 Select **Deploy Locally** for the purposes of this example.

4 Check **Include Documentation** to generate a PDF document that describes the business flow/cartridge structure, including your annotations.



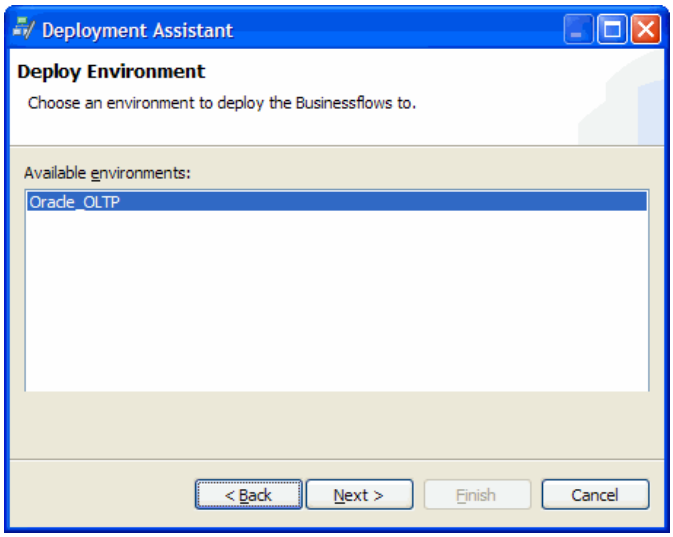
5 Click **Next**.

6 Type your encryption key in the Encryption Key field. The Encryption Key was set when installing the repository. The Encryption Key is only required once in each session.



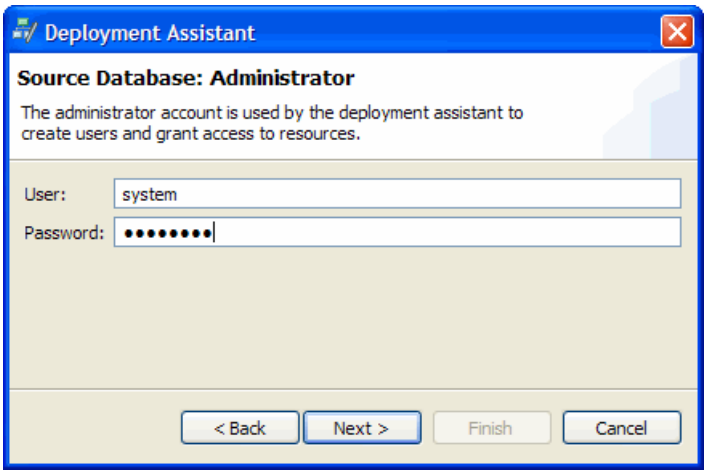
7 Click **Next**. The Deploy Environment page displays.

8 Choose the environment to which you want to deploy this business flow, for example, Oracle_OLTP.

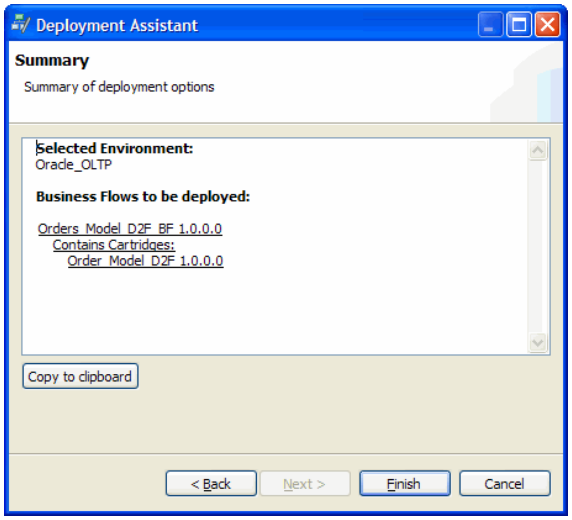


9 Click **Next**.

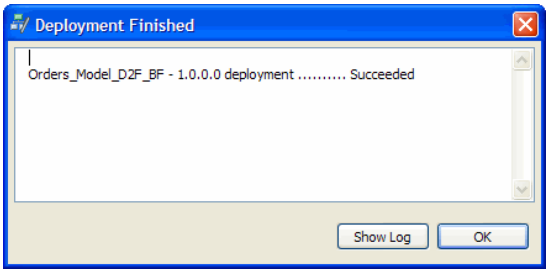
10 The Source Database Administrator page prompts you for the credentials of the active database. The Deployment Assistant needs this to grant access to the cartridge tables. For example, user *system* and password *oracle* or user *sa* and password *manager1*.



11 Click **Next**. The Summary page shows a summary of the options you have selected.



12 Click **Finish**. You may have to wait a few minutes before the Deployment Finished dialog appears.



13 When the Deployment Finished dialog appears, click **Show Log** to show the log file. Review the log and ensure there are no errors or problems.

14 If you discovered errors in the previous step, click **OK** and step back through the Deployment Assistant to correct the problems. If there were no errors, click **OK** to close the log file.

15 Click **OK** to close the Deployment Finished dialog.

16 In the Deployment Assistant, if you specified Include Documentation, you should find a PDF file with your business flow's documentation located in <*install_dir*>\obt\businessflow*environment_name*. For example:

```
C:\Program Files\HPTDM\obt\businessflow\Oracle_OLTP\
Orders_Model_D2F_BF.1_0_0_0.pdf
```

Examine the PDF file. You should find all of your annotations along with other useful information about the structure and design of your business flow.

In this same directory, you should also find the generated business flow file, `Orders_Model_D2F_BF.1_0_0_0.busflow`.

Running the business flow

After your business flow is successfully deployed, you are ready to run it. You can run jobs from the Web Console or the command line. For this tutorial, run the job through the Web Console.

1 Ensure that all of the requirements listed in Deployment prerequisites (page 63) have been met.

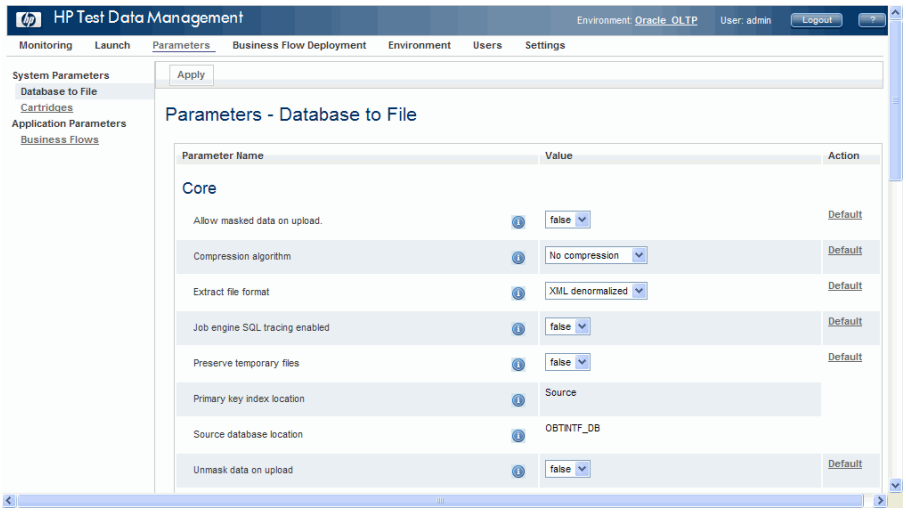
2 If the Web Console is not open in your browser, invoke it by accessing its URL. For example, `http://localhost:8090/WebConsole`.

3 If you created a user according to Deployment prerequisites (page 63), login as that user. Otherwise, login as the admin user, whose password you set when installing the repository.

TIP If you have more than one environment available in your Web Console, make sure that the currently active environment is the one where you plan to run your business flow. To check the active environment and change it if necessary, click **Environment** from the menu at the top of the page.

4 Before you launch the business flow, you need to change some parameter values. Click **Parameters** from the menu at the top of the page to review the parameters and their values.

The database to file parameters display first.



5 Because some of the data to be uploaded is masked and later on we plan to upload it, set the **Allow masked data on upload** parameter to **true**.

6 Scroll to the **Extract File Format** parameter. It is set to XML by default, which means that the extract file will be XML rather than comma separated values (CSV). For the purposes of this tutorial, change the value to **CSV normalized**.

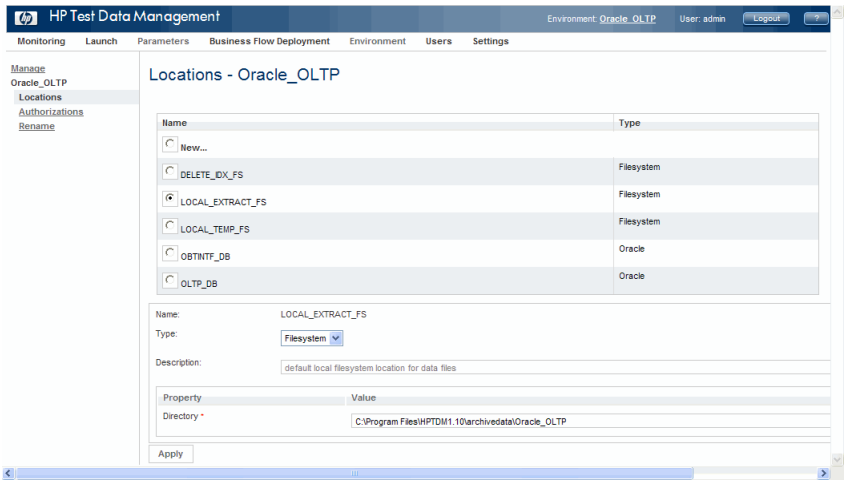
7 Click **Apply**.

8 (Optional) Because the business flow will create files on the file system, you might also want to confirm the exact location where HP Test Data Management will create files. To perform this procedure, you need to be the admin user or another user with Manage Environment privileges.

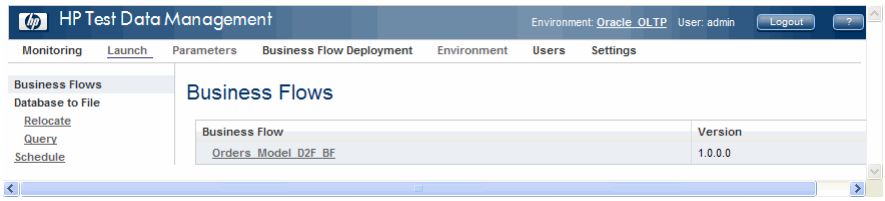
   a Click **Environment** from the menu at the top of the page.

   b Click **Locations** in the left navigation pane.

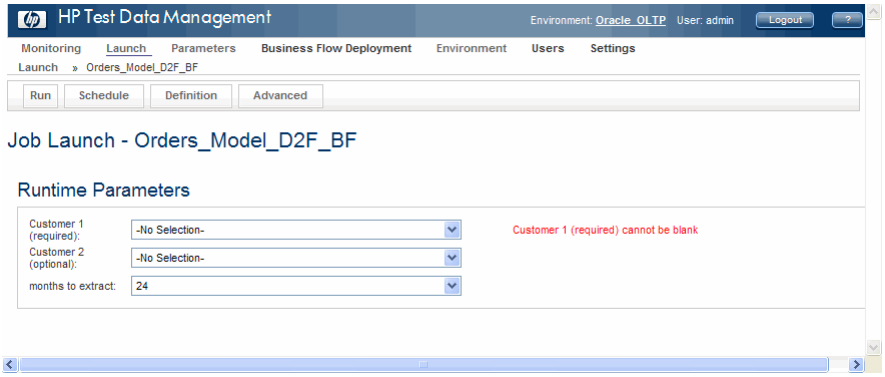
   c Click the **LOCAL_EXTRACT_FS** system parameter. A pane opens at the bottom of the page displaying the settings. Note down the **Directory** property value. This path is where files will be created. You could change this path, but, for the purposes of this tutorial, the default location is fine.



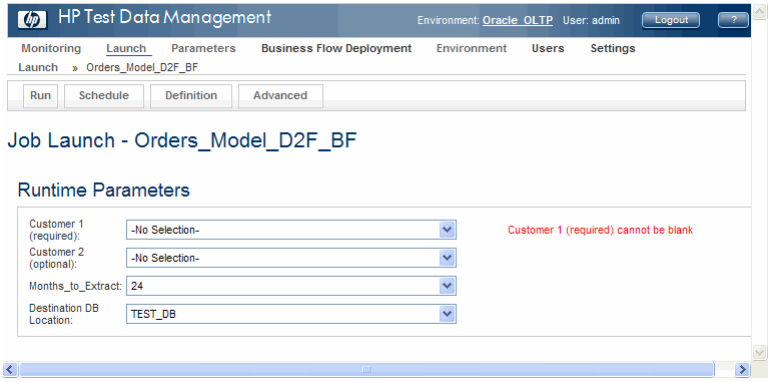
9 Click **Launch** from the menu at the top of the page.



10 Click **Orders_Model_D2F_BF**. The Launch page for that business flow appears. Notice that the first two parameters are the ones you created in Designer (business unit and customer). Below that, each of the selected tables is listed with the location in which to create the extract file for it.



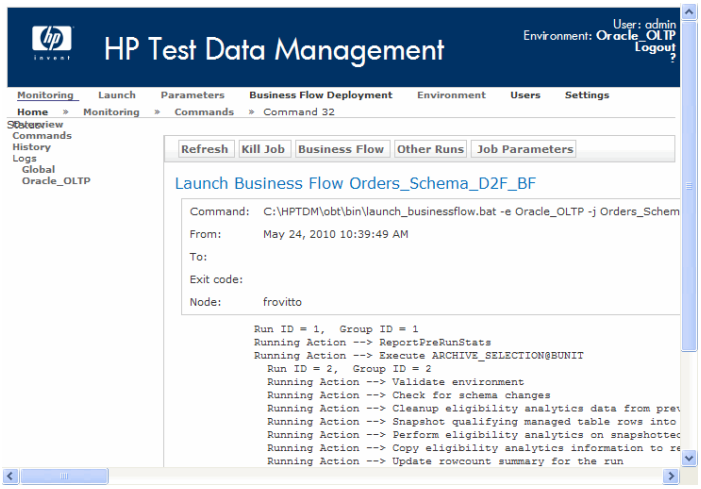
If you added the optional upload activity to your business flow in Adding an upload activity (optional) (page 61), the Launch page will also prompt you for a destination database.



11 For Customer 1 and 2, select customers from the drop-down list. For example, you could choose ABN AMRO June and ACER Malcolm. Leave the default value for months to extract.

12 Click **Run**.

13 Click **OK** to confirm the run when prompted. The business flow is launched and you are taken to a monitoring page that will periodically refresh with the latest status.



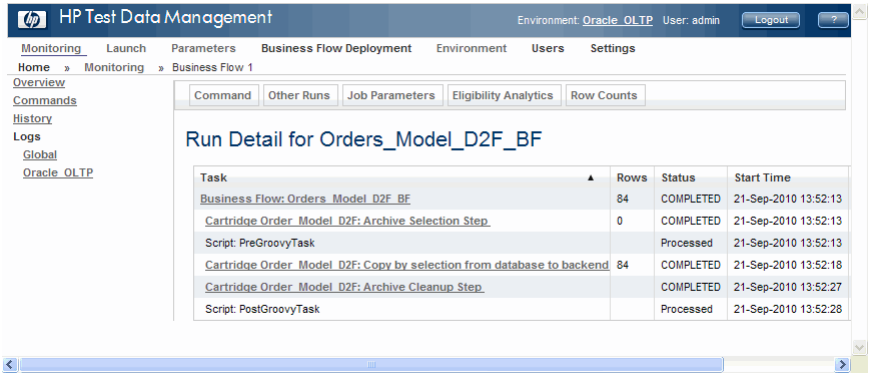
14 When the job completes, notice the success message at the end of the log:

```
Job completed successfully
Please see the Job Monitor for more information.
```

15 Click **Monitoring** in the menu at the top of the browser. Notice that the Orders_Model_D2F_BF business flow is displayed as having completed successfully.

16 Click **Orders_Model_D2F_BF** to drill down more deeply into its status.

TIP Had the job failed for some reason, you could also cancel the job from this page in the user interface. In this case, the job succeeded and you only see options to more closely review the results of the execution.



17 Click **Row Counts** to review the number of rows that were copied into the output.

Once you are done reviewing and validating the results of the job execution, you can begin to upload the test data to your test instance.

Uploading to your test database

NOTE If you included the optional upload activity in your business flow as described in Adding an upload activity (optional) (page 61), you do not need to perform the steps in this section.

Once the business flow has completed and you have verified its results, you can upload the data to your test instance.

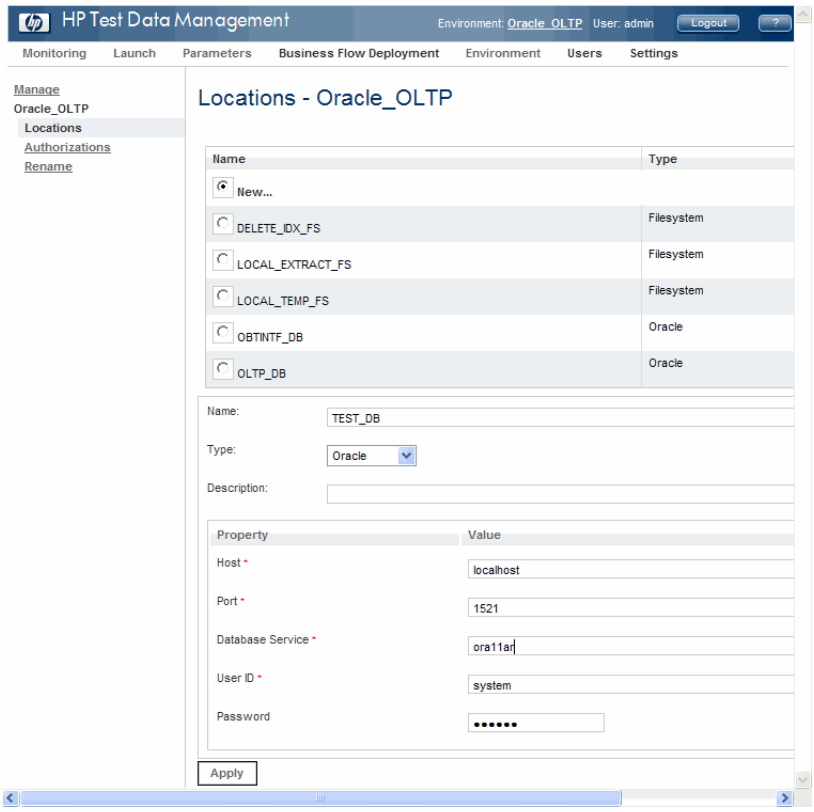
TIP You can also use Groovy to create a business flow that performs the upload and thus build this process into your business flow and schedule it. You need to work with HP Enterprise Services to build such a solution.

To upload the test data:

1. Click **Environment** from the top menu in the Web Console to add a database connection to your test database.
2. Click **Locations** in the left navigation bar. In order to upload your test data, you must first create a new location for the test database to which you plan to upload.
3. Select the **New** radio button.
4. Enter the properties listed in Table 5.

Table 5 TEST_DB connection properties

Connection name	Connection properties
TEST_DB	Type: **Oracle** or **SQL Server**, or **DB2**. Host: IP address or network name of the server where your test database instance resides. Port: The port on which your database listens. For example, 1521 or 5001. Database Service (Oracle only): The name of your Oracle instance. For example, **`ora11ar.`** Database instance (SQL Server only): The name of your SQL Server instance. For example, **`MSOLTP`**. Database (DB2 only): The name of your DB2 database. For example, **`TEST`**. User ID: The owner of the schema where you want to store the test data. For example, **`demarc`**. Password: The password for your test schema owner. For example, **`demarc`**.



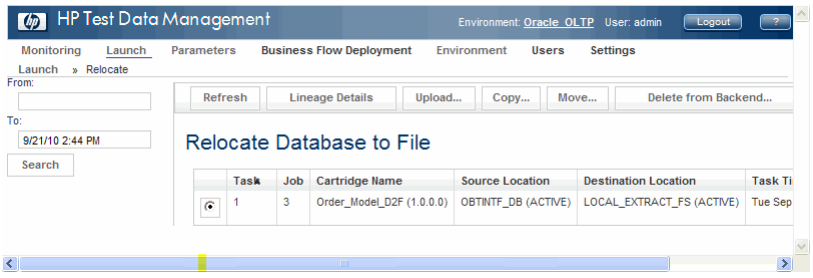
5 Click **Apply**.

6 Click **Launch** from the top menu in the Web Console.

7 In the left navigation bar, click **Relocate**. You are presented with a list of cartridges that have previously run.

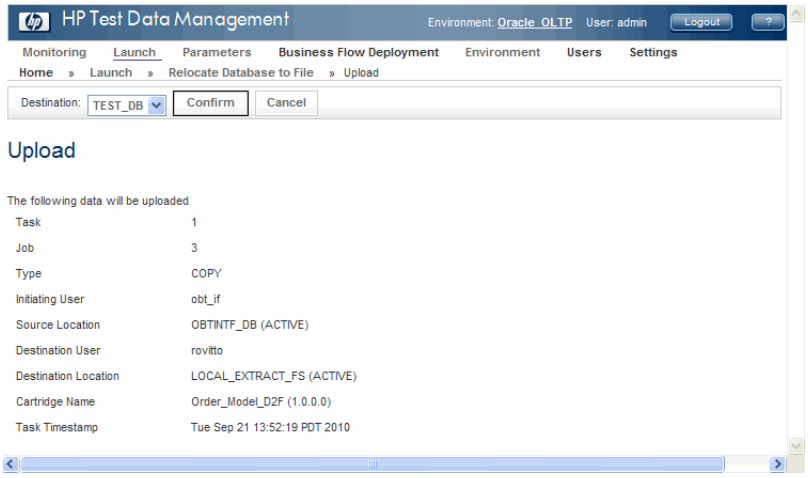
TIP In the case of a schema-based cartridge, you would see one cartridge for each table that you extracted. A schema-based business flow creates a cartridge per table. A mode-based business flow creates one cartridge for all tables in the business flow. Hence, you only see one cartridge for each run of the business flow.

8 Select the radio button for a cartridge.



9 Click **Upload**.

10 Confirm that TEST_DB is the Destination and click **Confirm**. The upload job launches and you receive log information just as you did when you ran the business flow. It may take several moments for the data to upload into your test database.



11 Use an ad hoc query tool to confirm that the table was in fact created in the correct location with the correct data. If not, ensure that your new location (TEST_DB) is correct.

TIP When you return to the Relocate Database to File page, notice that a new entry will be added to the end of the list for the upload job that you just ran. As a result, you will see Orders_Model_D2F listed twice, once at the beginning of the list for the copy to CSS job and once at the end of the list for the upload to database job.

Summary and next steps

In this chapter you learned to:

- Create an environment and user in the Web Console
- Deploy your business flow
- Find the generated business flow and its PDF documentation
- Run the business flow from the Web Console
- Monitor the execution of the business flow from the Web Console
- Review the results of the job execution
- Upload the extracted data to a test instance of your database

The next step is to create a test input spreadsheet.

6 Loading test data into spreadsheets

After your data has been extracted to CSV or XML files, you can use the query server to directly load your data into spreadsheet format for use with various testing tools.

This chapter describes how you can use the query server on MS Windows to load the extracted data into MS Excel.

This chapter includes:

See also *HP Test Data Management Web Console and Query Server User Guide* for more information about the query server.

Installing the query server database driver

In order to access your CSV or XML data through the query server, you need the query server, and at least one of the DataDirect drivers (JDBC, ODBC, or OLEDB) included with HP Test Data Management. The JDBC driver is installed by default. For the purposes of this tutorial, we will use the ODBC driver to access the XML through SQL and MS Excel. Hence, you must first install the ODBC driver.

NOTE On Unix, only the JDBC and ODBC drivers are provided. Refer to the *HP Test Data Management Web Console and Query Server User Guide* for information about configuring the drivers on Unix.

If you have already installed the ODBC driver, you can skip to Creating user collections (page 77). Otherwise, follow these instructions:

1. Open a command window by selecting **Start > Run**, typing **`cmd`**, and clicking **OK**.
2. Change directories to <*install_dir*>`\obt\bin`, for example:

   ```
   cd C:\Program Files\HPTDM\obt\bin
   ```

3. Once the server and data source are installed, invoke the driver batch file to install the desired drivers:
   - On MS Windows:

     ```
     oadriver.bat
     ```

— On Linux:

```
oadriver.sh
```

4 When prompted, indicate that you want to install a client driver:

```
1. Install driver
2. Uninstall driver
[Please enter your option] 1
```

5 When prompted, indicate that you want to install the ODBC driver:

```
1. ODBC Driver
2. ADO OLE DB Driver
[Please enter your driver option] 1
```

NOTE Client-side OLE DB is only available on MS Windows 32-bit platforms.

When the installation completes, you should receive a message similar to the following:

```
Installing OA ODBC Client...
OA ODBC Client is installed. Install log created at
C:\Program Files\HPTDM\obt\log\odbcClientInstall.log.
```

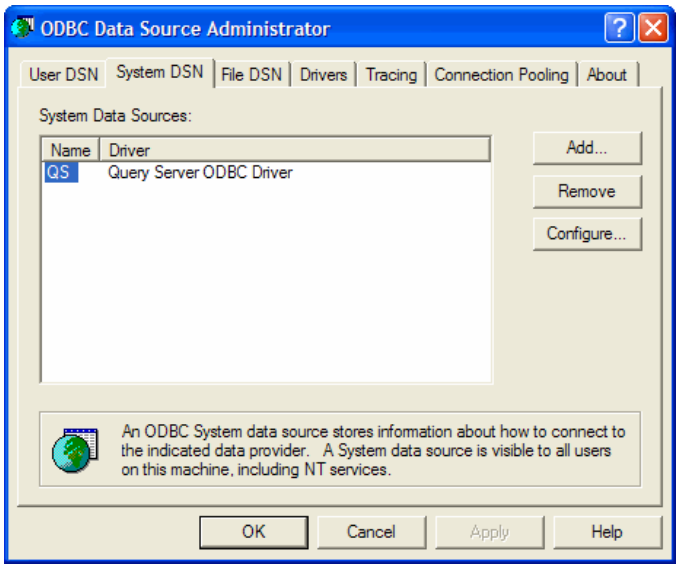
6 Optionally, review the log files in `<install_directory>\obt\log` for errors:

— `oaserver.log` (install log for the server)

— `odbcClientInstall.log` (install log for the client)

— `oaerror.log` (log of configuration errors during the setup)

7 Click the **System DSN** tab.



You should see the new ODBC system data source QS. This ODBC data source is created for your convenience when the ODBC driver is installed. It is automatically configured to access the default QS service data source

(database) xmlData. You can use the ODBC Data Source Administrator to change the name of the ODBC data source or to point it to a different QS service data source as desired.

By adding data collections to this data source, you can query them using standard SQL.

TIP If you do not see the QS data source here, check your log files for errors.

Creating user collections

An extract data collection is a file system location of extracted CSV, XML, and XSD files. Creating the collection enables the query server to view the files. A collection is defined by a specific file system directory and a wild-card filename pattern.

NOTE You can create collections through other methods as well. Refer to *HP Test Data Management Web Console and Query Server User Guide* for a complete description of collection methods.

You should create at least one collection to test file access and ensure that it behaves as expected. Subsequent to that, you or the database administrator may want to create additional collections for consumers of the extracted data. Different groups of consumers may have different data requirements, which may in turn necessitate specific collections.

To create a user collection using locally extracted files:

1. Start the query server by running the following command from the command prompt:

   ```
   <install_dir>\obt\bin\oaserver start
   ```

   TIP To stop the query server, run `oaserver stop`.

2. From the Start menu, choose **All Programs > HP Test Data Management > Interactive SQL (ODBC)**. A command window with an `ISQL>` prompt displays.
3. Connect to the server you started in the Management Console:

   ```
   connect install*OA@QS
   ```

4. Create a collection. For example:

   ```
   exec create collection orders_collection_user in schema
   orders using pattern 'C:\Program
   Files\HPTDM1.10\archivedata\Oracle_OLTP\Orders_Model_D2F*.xm*
   ';
   ```

If you receive an error, it typically indicates that your file pattern does not match any files. Confirm your path and file name pattern to ensure that it matches the location and files that you want to include.

TIP The `in schema <schema_name>` argument enables you to combine data from multiple cartridges into a single collection. In this case, we only have one cartridge, but, if there were more, they would be combined into the collection because of `in schema orders`.

NOTE The data directory is subdivided by environment names. Your CSV and XML files will reside in a subdirectory of data for your environment.

The collection is saved to the server. The success message includes the rows affected, which indicates the number of tables found in the extract files and configured for SQL access. If you receive any error messages, make the specified corrections and try creating the collection again.

5 You can now import the data into Microsoft Office Excel or another ODBC/JDBC client of your choice. Refer to Creating a spreadsheet from Microsoft Office Excel (page 78).

See also *HP Test Data Management Web Console and Query Server User Guide*

Creating a spreadsheet from Microsoft Office Excel

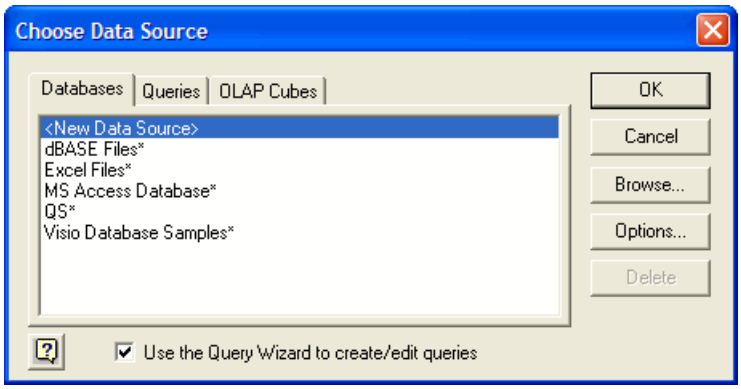
To access your collection in Microsoft Excel:

1 Open Microsoft Office Excel.

NOTE The steps in this section are based upon Microsoft Office 2003.

2 From the main menu, select **Data > Import External Data > New Database Query**.

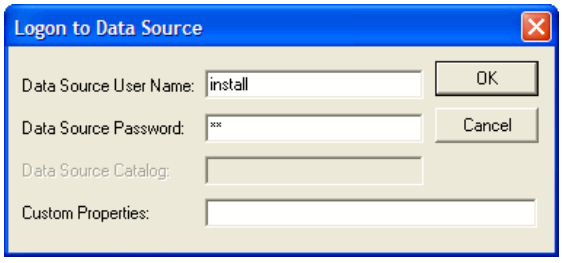
TIP In Microsoft Office Excel 2007, select Data > Get External Data > From Other Sources > From Microsoft Query.



3 Select **QS***.

4 Click **OK**.

5 In the OpenAccess Login dialog, type your User Name and Password. The default user name is **install** and the default password is **OA**.

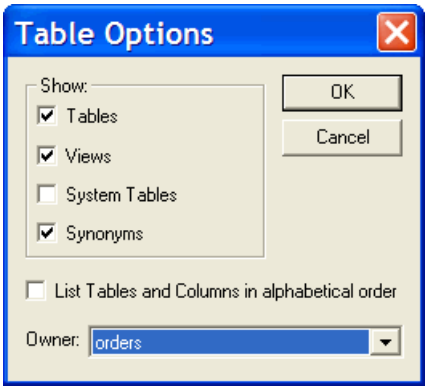


6 Click **OK**. The Query Wizard - Choose Columns page appears. Notice how the Available tables and columns list displays contains a number of tables prefixed with OA_. These tables are query server system tables. You can optionally filter these tables out of the list as follows:

a Click **Options**.

b Uncheck System Tables.

c Choose **orders** from the Owner list.



d Click **OK**. Notice how the tables prefixed by OA_ no longer appear.

Adding a single table

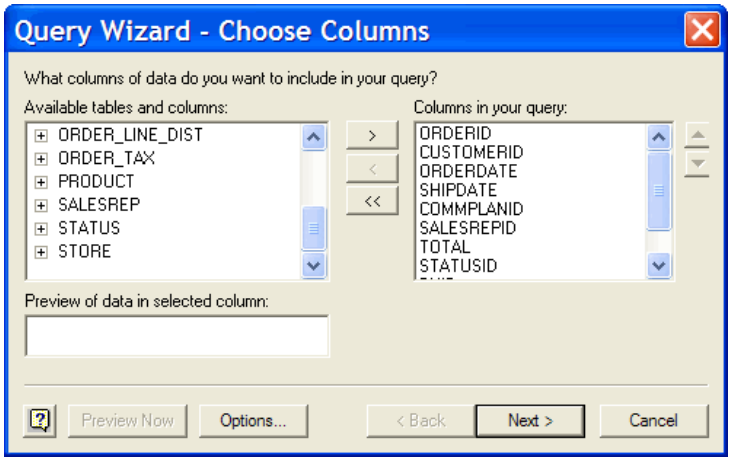
NOTE The data for a single table may be spread across several CSV files in your extract files. The query server collects all of the data from that table for you, regardless of the file in which it resides.

To add data from a single table to your spreadsheet:

1 In the Available tables and columns list, select the ORDER_HEADER table node.

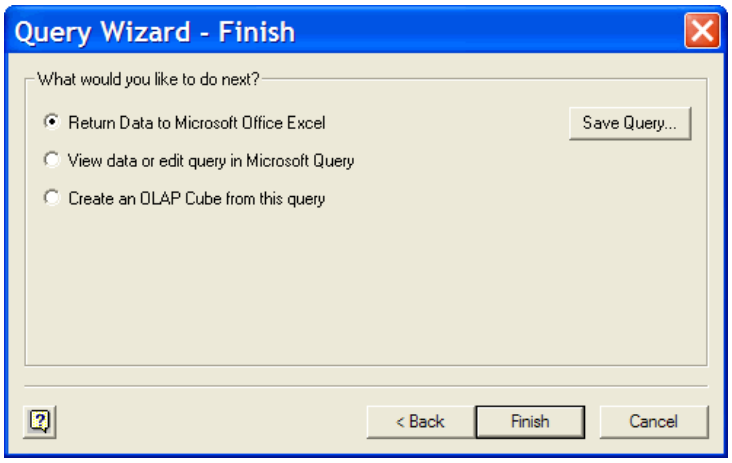
TIP If you wanted to select individual columns, you could simply expand the table nodes and select the desired columns.

2 Click the shuttle (>) to move the table and all of its columns to the Columns in your query list.



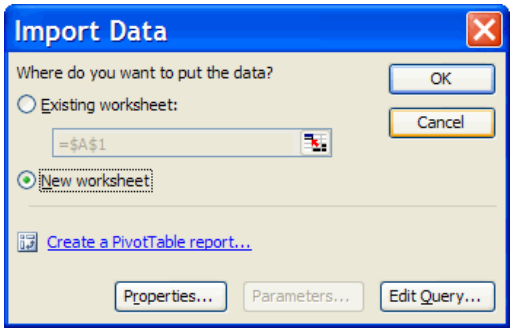
3 Click **Next**.

4 Continue to click **Next** until you reach the Finish page.

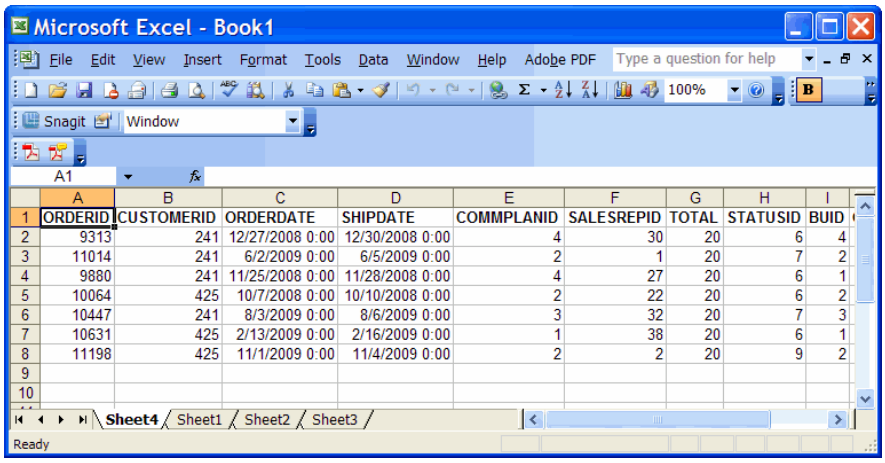


5 Click **Finish**. The Import Data dialog displays.

6 Select **New Worksheet**.



7 Click **OK**. The spreadsheet is populated with the data from the ORDER_HEADER table. Note how the query server has included a column header row for you.



TIP If you found that some of your data types could not be displayed properly in MS Excel, you could override those data types in the query server. Refer to *HP Test Data Management Web Console and Query Server User Guide* for more information.

Adding multiple, joined tables

Adding data from a single table is useful, but, in many cases, you need to include data from multiple tables and join it.

To add data from joined tables:

1 Repeat step 2 (page 78) through step 6 (page 79).

2 In the Available tables and columns list, select the ORDER_HEADER table node.

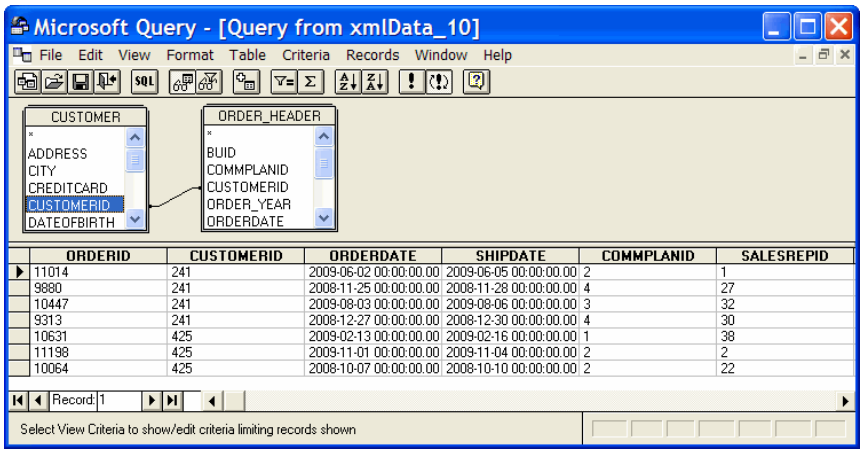
3 Click the shuttle (**>**) to move these tables and all of their columns to the Columns in your query list.

4 Repeat step 2 and step 3 for the CUSTOMER table.

5 Click **Next**. A message dialog appears indicating that, because you selected multiple tables, you need to use Microsoft Query to indicate how to join the tables.

6 Click **OK**. Microsoft Query opens with ORDER_HEADER and CUSTOMER tables displayed. You need to indicate which primary/foreign keys are necessary to join the data in these two tables.

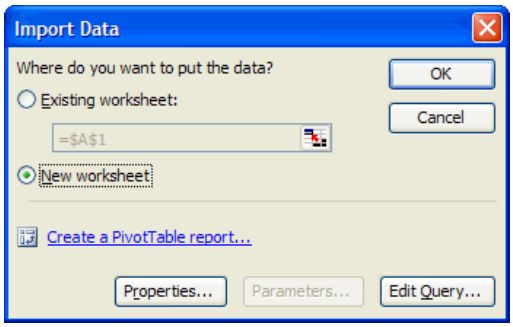
7 Click and drag the CUSTOMERID column in the CUSTOMER table and drop it on the CUSTOMERID column in the ORDER_HEADER table. This step indicates to Microsoft Query that is should use the CUSTOMERID column to join the data in the two tables.



TIP You could use Microsoft Query to further manipulate the data at this point, but it is not necessary for the purposes of this tutorial. For example, you could change the order of the columns or the sort order of the rows.

8 Click the **Return Data** tool. The Import Data dialog box appears.

9 Select **New Worksheet**.



10 Click **OK**. Your data is loaded into the spreadsheet and you can manipulate it as you would any other data in an Excel spreadsheet.

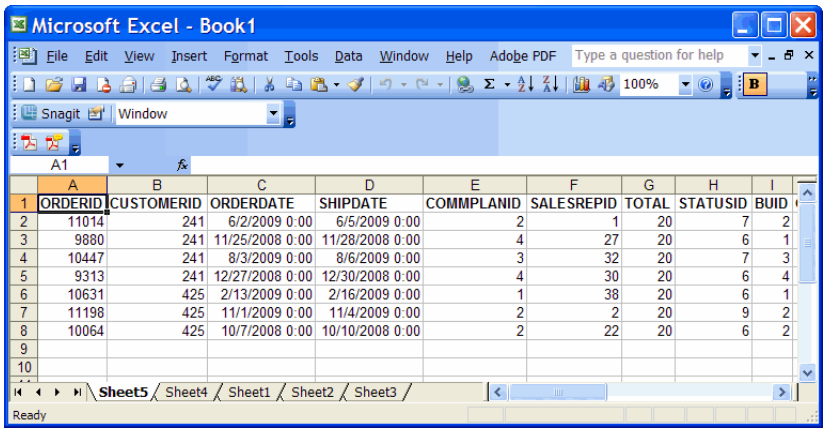
As you look through the spreadsheet, note the following:

— The query server has included a column header row for you.

— The spreadsheet includes joined data from the ORDER_HEADER and CUSTOMER tables.

— The SOCIAL, CREDITCARD, and LASTNAME column values are masked as you specified when building your cartridge.

TIP If you found that some of your data types could not be displayed properly in MS Excel, you could override those data types in the query server. Refer to *HP Test Data Management Web Console and Query Server User Guide* for more information.



	A	B	C	D	E	F	G	H	I
1	ORDERID	CUSTOMERID	ORDERDATE	SHIPDATE	COMMPLANID	SALESREPID	TOTAL	STATUSID	BUID
2	11014	241	6/2/2009 0:00	6/5/2009 0:00	2	1	20	7	2
3	9880	241	11/25/2008 0:00	11/28/2008 0:00	4	27	20	6	1
4	10447	241	8/3/2009 0:00	8/6/2009 0:00	3	32	20	7	3
5	9313	241	12/27/2008 0:00	12/30/2008 0:00	4	30	20	6	4
6	10631	425	2/13/2009 0:00	2/16/2009 0:00	1	38	20	6	1
7	11198	425	11/1/2009 0:00	11/4/2009 0:00	2	2	20	9	2
8	10064	425	10/7/2008 0:00	10/10/2008 0:00	2	22	20	6	2
9									
10									

Summary and next steps

In this chapter you learned to:

- install the SQL access database driver
- create collections
- import your data into MS Excel

You have now completed the basic HP Test Data Management tutorial. You may wish to review other portions of the HP Test Data Management documentation set to increase the depth of your knowledge of the product:

- Chapter 7, Creating a schema-based cartridge (optional) explains how to build a schema-based cartridge.
- *HP Test Data Management Designer User Guide* contains more information about how to use the Designer component to build test data management projects.
- *HP Test Data Management Web Console and Query Server User Guide* contains more information about how to use the Web Console component to deploy, run, monitor, and manage your business flows and cartridges. In particular, you may want to review the following tutorials:
  - Chapter 2, Configuring the Web Console
  - Chapter 3, Deploying and running business flows
  - Chapter 4, Viewing eligibility analytics

7 Creating a schema-based cartridge (optional)

One approach to creating test data is to build a schema-based cartridge. This chapter walks you through developing your project and cartridge definition for the Demarc data.

This chapter includes:

NOTE The steps and screen images in this optional exercise tutorial assume Oracle, SQL Server, or DB2. If you use another database, you can still follow the steps, but you may encounter some small differences in the steps and/or the appearance of the product.

Creating a schema-based cartridge

In most cases, the fastest way to design an extraction definition is to create a cartridge and add to it all of the tables to be extracted. This type of cartridge is known as a schema-based cartridge. In a schema-based cartridge, you simply choose tables without specifying their relationship to one another.

A cartridge is the mechanism by which HP Test Data Management specifies a versioned instance of an extraction definition. Among other things, you select the following in your cartridge definition:

- Which tables to include
- Which rules to apply
- Data masking
- Column inclusion

Alternatively, if your objective requires an understanding of the relationships among tables, you need to build a model-based solution instead. Building a data model-based solution is described in Creating a subset of data (page 19).

To create a schema-based cartridge:

1. Go to **File > New Cartridge**.
2. In the New Cartridge dialog box, type **`Orders_Schema_D2F`** as the Name.
3. Select the **Schema** radio button as the Source.

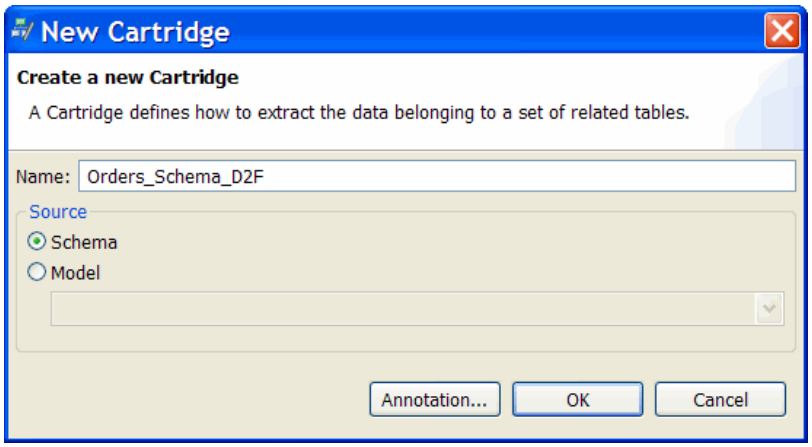
Advanced concept

Schema means that the cartridge is based upon the database schema rather than a defined data model with relationships. This type of cartridge is typically used when you want to quickly select a large number of tables and apply minimal rules for selection to them.

Model means that the cartridge is based upon a defined data model with relationships. This type of cartridge is typically used in cases where you need more numerous and complex rules, and simplified upload.

Related information

HP Test Data Management Designer User Guide.



4. Click **OK**.
5. Click **OK**. The Database to File Cartridge editor appears.

Navigating in the cartridge editor

Refer to Navigating in the cartridge editor (page 55) for general navigation tips about the cartridge editor.

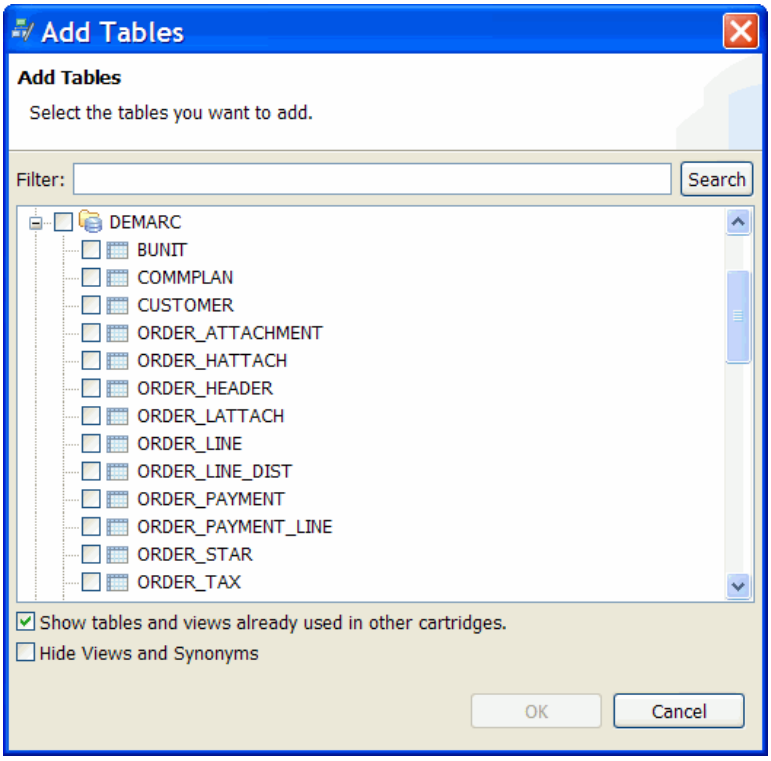
TIP The tabs you see in the editor are different for model-based and schema-based cartridges, but the basic navigation model remains the same.

Adding tables to the cartridge

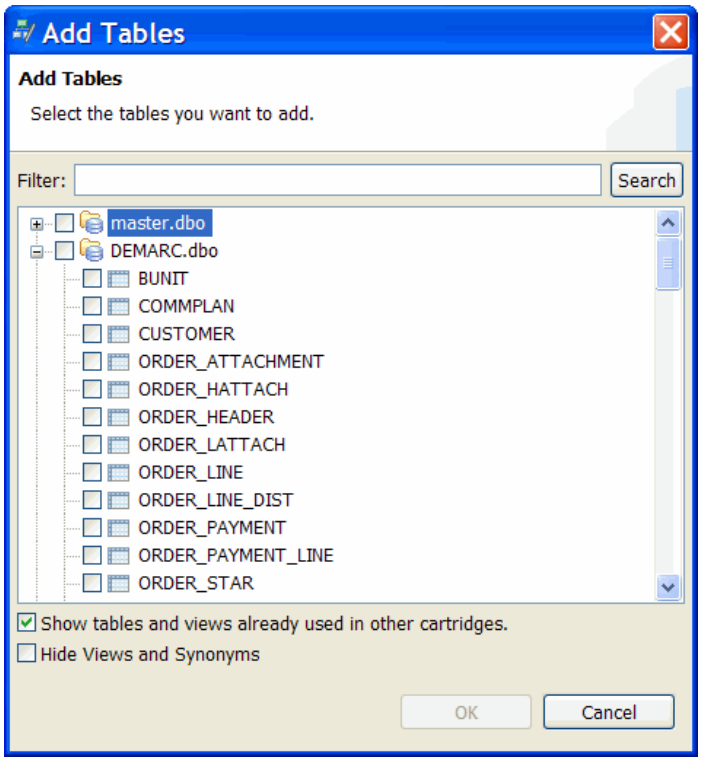
To choose tables for inclusion in the cartridge:

1 Click **Add** in the Tables to be Extracted section. The Add Tables dialog displays.

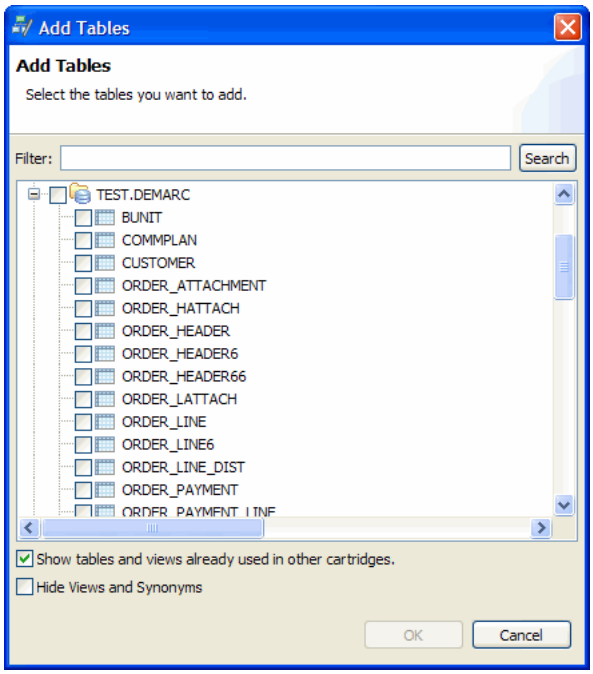
For Oracle An example for Oracle



For SQL Server An example for SQL Server 2005



For DB2 An example for DB2



2 Find the DEMARC schema and select all of its tables for inclusion in the cartridge.

3 Click **OK**.

Defining subsetting rules

For some tables, you probably only want to extract a subset of data for testing purposes. For others, you may want all of the data. For example, from a lookup table, you might want all of the data, but, from a transactional table, you might only want some of the data. To restrict the amount of data eligible for extraction, you create rules on the tables in the cartridge.

In most cases, it is a best practice to parameterize your rules such that you can change the basis of the condition at runtime. For example, if you want to select data from a table on the basis of its age, you should create a parameter that represents the age limit. In this way, you can dynamically choose the age of the records you want to copy at runtime.

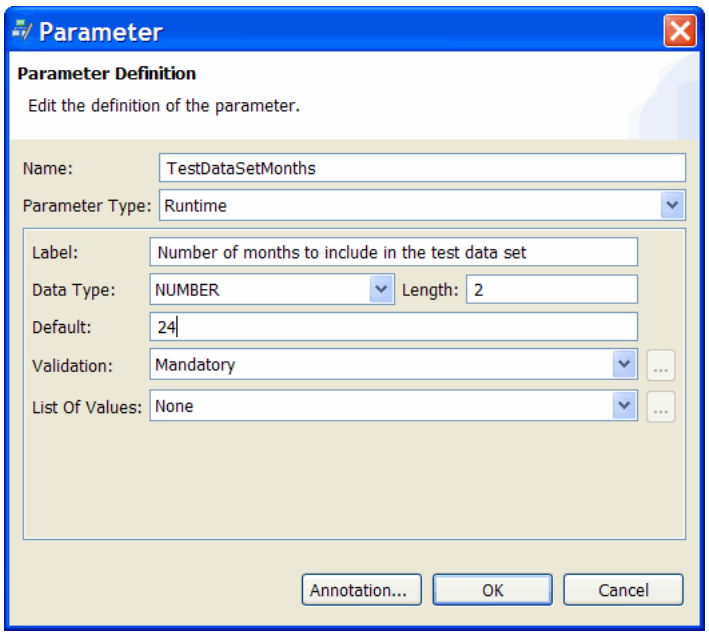
- Adding parameters (page 89)
- Adding rules (page 92)

Adding parameters

Before creating your rules for selecting data, you will create some parameters that you will need to use in those rules.

To add parameters to your project:

1. Right-click the **Parameters** folder in the Project Navigator and select **New Parameter** from the pop-up menu to create a parameter.



2. Enter or select the following for the parameter definition:
   - For Name, type **`TestDataSetMonths`**.
   - Ensure that Parameter Type is set to **Runtime**.

— For Label, type **`Number of months to include in the test data set`**.

— For Data Type, select **Number**.

— For Length, type **`2`**.

— For Default, type **`24`**.

— Ensure that Validation is **Mandatory**.

— Leave List of Values as **None**.

TIP Optionally, you could add a static list of values for different lengths of time. For example, you might create a list that includes 24 months, 27 months, 30 months, and so on. That way, users could pick from the list rather than entering values manually. For more information about adding lists of values, refer to Defining a rule (page 41).

3 Click **OK**.

4 Right-click the **Parameters** folder in the Project Navigator and select **New Parameter** from the pop-up menu to create a parameter.

5 Create the TestCustomerID parameter.

a Enter the properties listed in Table 6.

Table 6 TestCustomerID parameter properties

Parameter name	Parameter settings
TestCustomerID	Parameter Type: **Runtime**
	Label: **`Customer to include`**
	Data Type: **String**
	Length: 30
	Default: blank
	Validation: Mandatory
	List of Values: SQL

b Click the **Browse** button to the right of List of Values. Because the customer IDs are listed in a lookup table (CUSTOMER), you need to associate a SELECT statement with the list of values to populate it with valid customer IDs.

c Ensure that Database Connection is **Source**.

d Select the **Drop-Down List** radio button. Drop-down list indicates that the list will appear as a selectable list of items. Selecting **Look-Up List**, specifies that list will appear in a separate, searchable dialog, which is best for long lists.

e Under Database, expand the **Any** node.

f Select the **Oracle** node.

g Under SQL, enter the following SELECT statement:

```
SELECT to_char(A.CUSTOMERID), A.LASTNAME FROM
${SOURCE.DEMARC.CUSTOMER} A
WHERE UPPER(A.LASTNAME) LIKE
upper('%'||:FilterPattern||'%')
UNION
SELECT '%', ' (All Customers)' FROM DUAL
ORDER BY 2
```

Notice that the SELECT statement returns two values, one for the actual customer id column value and one for the customer's last name, which is what the user will see in the list. Furthermore, a UNION is used to run the extraction for all customers.

h Under Database, select the **SQL Server** node.

i Under SQL, enter the following SELECT statement:

```
SELECT CAST(A.CUSTOMERID AS VARCHAR), A.LASTNAME FROM
${SOURCE.DEMARC.dbo.CUSTOMER} A
WHERE UPPER(A.LASTNAME) LIKE upper('%'+:FilterPattern+'%')
UNION
SELECT '%', ' (All Customers)'
ORDER BY 2
```

TIP You can click **Validate** to confirm the syntax of your Oracle or SQL Server SELECT statements if you are connected to Oracle or SQL Server databases, respectively. If you are not connected to that database type, you may receive an error message.

j Under Database, select the **DB2** node.

k Under SQL, enter the following SELECT statement:

```
SELECT * FROM (
  SELECT CAST(A.CUSTOMERID as CHARACTER), A.LASTNAME FROM
   ${SOURCE.DEMARC.CUSTOMER} A
   WHERE UPPER(A.LASTNAME) LIKE
   UPPER('%'||:FilterPattern||'%')
   UNION ALL
   VALUES ('%',' (All Customers)'
)
ORDER BY 2
```

l Click **OK**.

6 Click **OK**.

7 Right-click the **Parameters** folder in the Project Navigator and select **New Parameter** from the pop-up menu to create a parameter.

8 Create the TestBusUnit parameter properties.

a Enter the properties listed in Table 7.

Table 7 TestBusUnit parameter properties

Parameter name	Parameter settings
TestBusUnit	Parameter Type: **Runtime**
	Label: **Business unit to include**
	Data Type: **String**
	Length: 30
	Default: blank
	Validation: Mandatory
	List of Values: SQL

b Repeat step b (page 90) through step l (page 91) under step 5 (page 90) but use the following SELECT statements:

Oracle

```
SELECT to_char(BUID), BUNAME from ${SOURCE.DEMARC.BUNIT}
UNION
SELECT '%', 'ALL' from DUAL
```

SQL Server

```
SELECT CAST(A.BUID AS VARCHAR), A.BUNAME FROM
${SOURCE.DEMARC.dbo.BUNIT} A
UNION
SELECT '%', 'ALL'
```

DB2

```
SELECT CAST(A.BUID as CHARACTER), A.BUNAME FROM
   ${SOURCE.DEMARC.BUNIT} A
   UNION ALL
   VALUES ('%',' (All Customers)'
```

9 Click **OK**.

Adding rules

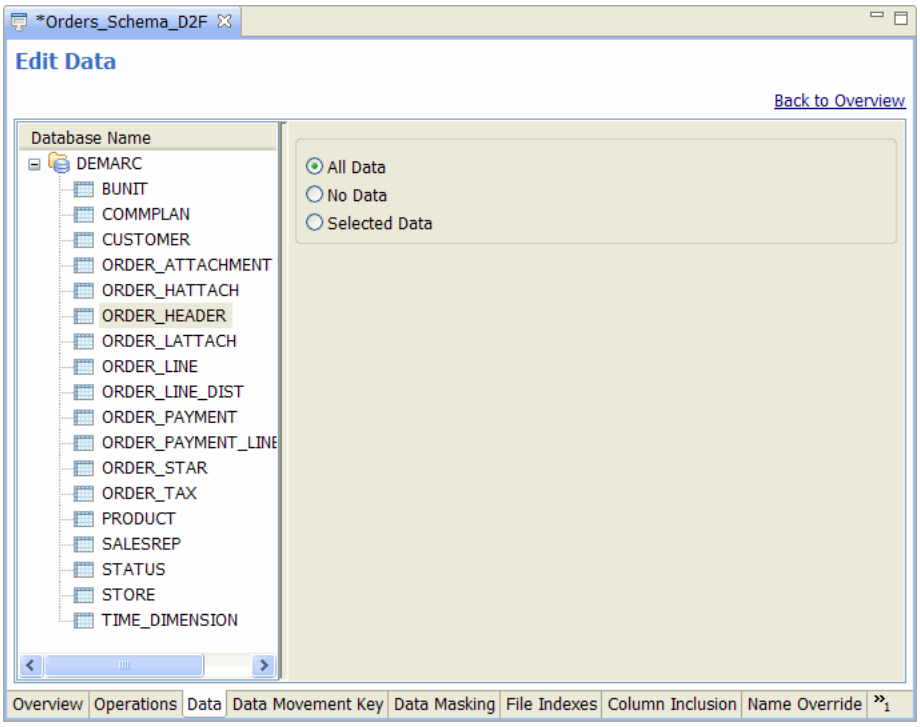
To add rules to the tables extracted by the cartridge:

1 If it is not already open, double-click the **Orders_Schema_D2F** cartridge in the Project Navigator to open the cartridge editor.

2 Select the **Data** tab.

3 Select the **ORDER_HEADER** table.

ORDER_HEADER is the driving table for the order entry application. All of the other tables are children of ORDER_HEADER. Notice that by default the cartridge will select all of the data from the ORDER_HEADER. In most cases, you will not want every row from transactional tables like ORDER_HEADER.



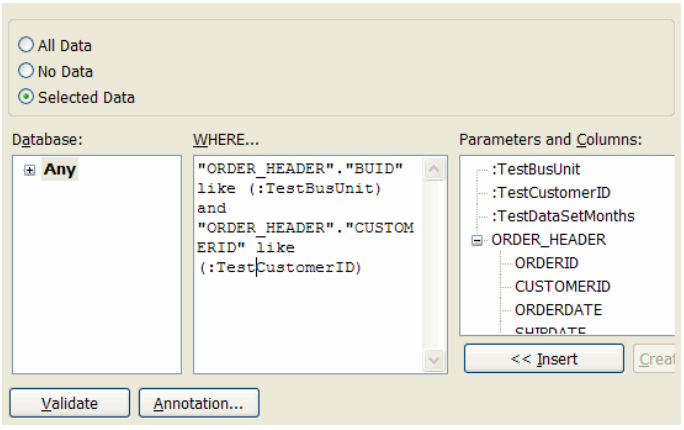
4 Select the **Selected Data** radio button. A pane opens where you can enter a WHERE clause to restrict the data returned from the ORDER_HEADER table.

5 Under Database, select the **Any** node.

6 Under WHERE, enter the following clause:

```
"ORDER_HEADER"."BUID" like (:TestBusUnit) and
"ORDER_HEADER"."CUSTOMERID" like (:TestCustomerID)
```

This WHERE clause restricts the data returned using two of the parameters you created in Adding parameters (page 89). It will only return data for the business unit and customer that you specify at runtime.



7 If you are connected to Oracle or SQL Server, click **Validate** to confirm the syntax of your WHERE clause.

TIP For WHERE clauses that have database specific syntax, you must select that database and then enter the SQL. Furthermore, the Validate button is only available for Oracle and SQL Server, and you must be connected to the appropriate database type to get an accurate validation. For example, if you enter a WHERE clause that is SQL Server specific and are connected to a SQL Server instance, you can click Validate to confirm your syntax. If you are not connected to a SQL Server instance, clicking Validate will typically throw an error on any SQL Server specific syntax. Validate is not available for DB2.

Because the driving table, ORDER_HEADER is restricted by customer id and business unit, its child transactional tables, like ORDER_LINE, must be restricted by the same criteria. Otherwise, you would end up with orphan rows in ORDER_LINE that have no corresponding row in ORDER_HEADER. Such data mismatches could spoil your test cases.

8 Repeat step 3 (page 92) through step 7 (page 93) for the tables listed in Table 8.

Table 8 WHERE clauses for tables

Table	WHERE clause
BUNIT	`"BUNIT"."BUID" LIKE (:TestBusUnit)`
CUSTOMER	`"CUSTOMER"."CUSTOMERID" LIKE :TestCustomerID`
ORDER_ATTACHMENT	**For Oracle and DB2:** `("ORDER_ATTACHMENT"."ATTTYPE" = 'OH' and "ORDER_ATTACHMENT"."ORDERID" IN (SELECT A.ORDERID FROM DEMARC.ORDER_HEADER A WHERE A.BUID LIKE :TestBusUnit AND A.CUSTOMERID LIKE :TestCustomerID)) OR ("ORDER_ATTACHMENT"."ATTTYPE" = 'OL' AND "ORDER_ATTACHMENT"."ORDERID" IN (SELECT B.ORDERLINEID FROM DEMARC.ORDER_LINE B WHERE B.ORDERID IN (SELECT A.ORDERID FROM DEMARC.ORDER_HEADER A WHERE A.BUID LIKE :TestBusUnit AND A.CUSTOMERID LIKE :TestCustomerID)))` **For SQL Server:** `("ORDER_ATTACHMENT"."ATTTYPE" = 'OH' and "ORDER_ATTACHMENT"."ORDERID" IN (SELECT A.ORDERID FROM DEMARC.dbo.ORDER_HEADER A WHERE A.BUID LIKE :TestBusUnit AND A.CUSTOMERID LIKE :TestCustomerID)) OR ("ORDER_ATTACHMENT"."ATTTYPE" = 'OL' AND "ORDER_ATTACHMENT"."ORDERID" IN (SELECT B.ORDERLINEID FROM DEMARC.dbo.ORDER_LINE B WHERE B.ORDERID IN (SELECT A.ORDERID FROM DEMARC.dbo.ORDER_HEADER A WHERE A.BUID LIKE :TestBusUnit AND A.CUSTOMERID LIKE :TestCustomerID)))`

Table 8 WHERE clauses for tables

Table	WHERE clause

ORDER_LINE	**For Oracle and DB2:**
	`"ORDER_LINE"."ORDERID" IN (SELECT ORDERID FROM DEMARC.ORDER_HEADER A WHERE A.BUID LIKE :TestBusUnit AND A.CUSTOMERID like :TestCustomerID)`
	For SQL Server:
	`"ORDER_LINE"."ORDERID" IN (SELECT ORDERID FROM DEMARC.dbo.ORDER_HEADER A WHERE A.BUID LIKE :TestBusUnit AND A.CUSTOMERID LIKE :TestCustomerID)`

ORDER_LINE_DIST	**For Oracle and DB2:**
	`"ORDER_LINE_DIST"."ORDERLINEID" IN (SELECT A.ORDERLINEID FROM DEMARC.ORDER_LINE A, DEMARC.ORDER_HEADER B WHERE A.ORDERID = B.ORDERID AND B.BUID LIKE (:TestBusUnit) and B.CUSTOMERID LIKE (:TestCustomerID))`
	For SQL Server:
	`"ORDER_LINE_DIST"."ORDERLINEID" IN (SELECT A.ORDERLINEID FROM DEMARC.dbo.ORDER_LINE A, DEMARC.dbo.ORDER_HEADER B WHERE A.ORDERID = B.ORDERID AND B.BUID LIKE (:TestBusUnit) and B.CUSTOMERID LIKE (:TestCustomerID))`

ORDER_PAYMENT	**For Oracle and DB2:**
	`"ORDER_PAYMENT"."CUSTOMERID" IN (SELECT A.CUSTOMERID FROM DEMARC.CUSTOMER A WHERE A.CUSTOMERID LIKE :TestCustomerID)`
	For SQL Server:
	`"ORDER_PAYMENT"."CUSTOMERID" IN (SELECT A.CUSTOMERID FROM DEMARC.dbo.CUSTOMER A WHERE A.CUSTOMERID LIKE :TestCustomerID)`

Table 8 WHERE clauses for tables

Table	WHERE clause

ORDER_PAYMENT_LINE	**For Oracle and DB2:**
	`"ORDER_PAYMENT_LINE"."ORDERLINEID" IN (SELECT A.ORDERLINEID FROM DEMARC.ORDER_LINE A, DEMARC.ORDER_HEADER B WHERE A.ORDERID = B.ORDERID AND B.BUID LIKE (:TestBusUnit) AND B.CUSTOMERID LIKE (:TestCustomerID))`
	For SQL Server:
	`"ORDER_PAYMENT_LINE"."ORDERLINEID" IN (SELECT A.ORDERLINEID FROM DEMARC.dbo.ORDER_LINE A, DEMARC.dbo.ORDER_HEADER B WHERE A.ORDERID = B.ORDERID AND B.BUID LIKE (:TestBusUnit) AND B.CUSTOMERID LIKE (:TestCustomerID))`

ORDER_STAR	**For Oracle and DB2:**
	`"ORDER_STAR"."ORDERID" IN (SELECT ORDERID FROM DEMARC.ORDER_HEADER A WHERE A.BUID LIKE :TestBusUnit AND A.CUSTOMERID LIKE :TestCustomerID)`
	For SQL Server:
	`"ORDER_STAR"."ORDERID" IN (SELECT ORDERID FROM DEMARC.dbo.ORDER_HEADER A WHERE A.BUID LIKE :TestBusUnit AND A.CUSTOMERID LIKE :TestCustomerID)`

ORDER_TAX	**For Oracle and DB2:**
	`"ORDER_TAX"."ORDERID" IN (SELECT ORDERID FROM DEMARC.ORDER_HEADER A WHERE A.BUID LIKE :TestBusUnit AND A.CUSTOMERID LIKE :TestCustomerID)`
	For SQL Server:
	`"ORDER_TAX"."ORDERID" IN (SELECT ORDERID FROM DEMARC.dbo.ORDER_HEADER A WHERE A.BUID LIKE :TestBusUnit AND A.CUSTOMERID LIKE :TestCustomerID)`

9 For each of the following tables, ensure that the **All Data** radio button is selected:

- COMMPLAN
- PRODUCT
- SALESREP
- STATUS
- STORE
- TIME_DIMENSION

10 For each of the following tables, select the **No Data** radio button:

- ORDER_HATTACH
- ORDER_LATTACH

11 Click **Back to Overview** to return to the Overview page.

Applying data masks to columns

Apply data masks as described in Applying data masks (page 55).

Adding data movement keys

All tables must have a data movement key associated with them in the Data Movement Key tab.

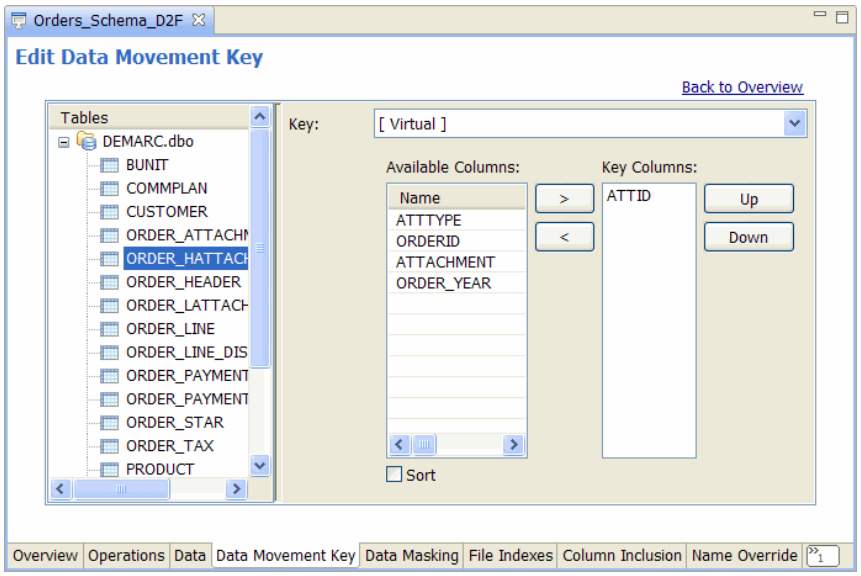
To confirm data movement keys:

1 Click the **Data Movement Key** title or tab from the Overview page.

2 Select each table to confirm that it has a data movement key assigned. ORDER_LATTACH and ORDER_HATTACH do not have data movement keys.

3 For ORDER_HATTACH, select **[Virtual]** from the list of values.

4 Use the shuttle (>) to move ATTID into the Key Columns list.



5 For ORDER_LATTACH, select **[Virtual]** from the list of values.

6 Use the shuttle (>) to move ATTID into the Key Columns list.

7 Click **Back to Overview** to return to the Overview page.

Creating a business flow

Create a business flow for your schema-based cartridge just as you did for your model-based cartridge in Chapter 4, Creating a business flow.

TIP Name the business flow **`Orders_Schema_D2F_BF`** and add the **`Orders_Schema_D2F`** extract cartridge to it.

Deploying a business flow

Deploy and run your business flow just as you did for your schema-based cartridge in Chapter 5, Deploying a business flow. Some pages and dialogs will appear differently for schema-based cartridges than they do for model-based cartridges, but the general flow is the same.

TIP When you run the business flow from Web Console, choose **`West`** for the business unit and **`ABN AMRO`** for the customer.

Summary

In this chapter you learned to:

- Create a schema-based cartridge
- Add tables to the cartridge
- Add rules to the tables in the cartridge
- Apply masks to specific columns
- Create parameters and lists of values.

Glossary

active database

The database from which you plan to extract data. Typically, this database is your online transaction processing (OLTP) or production database. In a two-tiered configuration, the active database resides on tier one and is the source for data movement operations.

active environment

The Web Console views and acts upon only one environment at a time, the active environment. To switch the active environment, you use the Change Active option in the Web Console.

activity

In Designer, a component of a business flow, which is added by using the toolbar. Note, activities in a business flow are different from what you see at runtime and therefore do not necessarily map directly to what you see in Console.

advanced selection

A method of data selection that discovers all of the interrelated rows from multiple tables and conceptually places them in the same application partition for extraction.

annotation

In Designer, a comment associated with the project, or one of its objects or components. These comments are collected and published in a PDF file when you right click a project or business flow and select Generate Documentation.

application partitioning

The concept of partitioning related rows together during data selection, regardless of whether they are in one or more tables. Application partitioning is unique to HP Test Data Management and contrasts with the more common table partitioning offered by the database management software, which only groups related rows from one table.

business flow

A series of activities, such as extraction operations and scripts, that run in sequence. You build business flows in Designer.

business flow status

The Web Console shows the last run of each business flow. The states are Complete/Error/Running.

cartridge

An instance of model- or schema-based eligibility criteria used to copy data from one location to another. Cartridges capture the application and business rules to ensure referential integrity of the data. For any one model in your project, you may have many cartridges that use it.

chaining table

The lower level table in a many-to-one or a many-to-many relationship between higher level and lower level tables in the model hierarchy.

collection

The configuration of a directory location and file pattern to match a set of extracted XML files, thus allowing SQL access to the extracted data.

comma separated values (CSV) A database to file output format that stores the data as values separated by commas and a metadata file. Each line in the CSV file corresponds to a row in a table. Within a line, fields are separated by commas, each field belonging to one table column. CSV files provide a simple format that many applications can import.

command Command files or JavaScript files launched by the Web Console on your behalf with status displays.

condition In Designer, the way you branch your business flow to run or skip an activity based on some criteria.

configuration parameter A type of parameter that has its values set by an administrator (someone who has repository privileges from Console) through the administrator interface. Typically, this type of parameter represents values that should be changed very infrequently, perhaps only at deployment time.

console user The Web Console identifies individual users, who are distinct from database users. The properties for a Console user are User Name, Full Name, Password, Enabled, Description, Email, Phone, and Privileges.

console user name The login name associated with a Console user.

constraint A column or a list of columns that enables you to identify rows in the database and relate them to one another.

customization A change that an administrator or DBA makes to a project provided by a third party, typically for a packaged application like Oracle PeopleSoft or Oracle E-Business Suite. As long as the customization is allowable by the project, the user can merge the customization into newer revisions of the third party project.

customization mode A Designer mode that provides visual cues to indicate customizations in the model. In a project with locked files, customization mode is on by default, but you can toggle it on and off from the toolbar in the model editor.

data masking The process of replacing private or confidential data during movement with a specified mask. You can choose from pre-defined masks that are part of HP Test Data Management or create your own mask.

data movement The method used by HP Test Data Management to actually copy data.

database constraint A constraint that exists in the database and can be discovered and referenced from Designer.

database to file A movement in which data goes from an active database to a file (XML or CSV format).

Deployment Assistant The user interface component used to deploy or generate business flows. You invoke Deployment Assistant from within Designer.

description A technical description created by the developer for her own reference. These descriptions do not appear in the generated PDF file for the cartridge or business flow.

Designer The user interface component used to develop, test, and deploy your extraction solution. Designer is a powerful graphical development environment for extraction solutions.

driving table A driving object is a root of a model hierarchy. Its relationship to the child tables drives the selection of transactions.

dynamic list of values A list of values for a parameter that obtains its members from a SELECT statement that returns identifiers and labels.

dynamic parameter A type of parameter that has its value set by a Groovy script that runs at deployment time to obtain a value. For example, this type of parameter can supply the type or version of a database or application, which can be obtained programmatically at deployment time.

embedded repository A Java database, installed with HP Test Data Management, that can act as your repository database, where you store your HP Test Data Management metadata. Alternatively, your source database or another database can act as the repository database.

environment The source and (optional) target credentials against which you plan to run commands. You can define multiple environments within your installation to identify various source databases.

error One of the ways in which you can interrupt a business flow. Error indicates that the business flow failed for some reason.

exclusive rules One of the ways in which HP Test Data Management determines whether to include or exclude rows from the extract operation. Exclusive rules require all rows in the constraint table to match for inclusion. Exclusive rules exclude the instance if the condition on any child is false, like STATUS='CLOSED'.

exit One of the ways in which you can interrupt a business flow. You can exit successfully or with a warning.

export The way that you save an HP Test Data Management project to an exchange format (.hdp) from the File menu. See also *import*.

export data The way that a user can send data to CSV format from Preview using the toolbar item.

extract data store The location where the data is to be copied. Can be an XML or CVS file.

generate documentation The process of collecting and grouping all annotations into a PDF file that also describes the business flow or cartridge structure.

import The way that you transfer projects from exchange format (.hdp) into the Project Navigator.

inclusive rules One of the ways in which HP Test Data Management determines whether to include or exclude rows from the extract operation. Inclusive rules require only one row in the constraint table to match the rule and be included. Inclusive rules include the instance if the condition on any child is true, like PRODUCT_RECALLED='Y'.

interrupt The way to stop or pause a business flow (pause, error, exit with warning, exit successfully).

local cache A capture of the metadata for your databases, schemas, and tables used when working offline in Designer.

local deployment The generation and deployment of your cartridge or business flow to an environment on your local, Designer client. Deployment files are generated locally and then deployed to the designated, local environment.

lookup table A table that contains helpful non-transactional information. For example, non-transactional information could be status definitions, or the name of the sales representative.

model A model identifies the tables and table relationships representing a business entity or related business entities. A project can have multiple models. Each model contains a driving table and all of its child and descendent tables.

model compatibility Each model in your project can have one or more dynamic parameters associated with it to verify the compatibility with the target environment. If the compatibility parameter returns false, then the cartridge referencing the model will not deploy or run and throw an error. For example, the script could return false for Oracle 10.2 and true for Oracle 11.1 to indicate that a cartridge referencing the model can only deploy and run against Oracle 11.1.

model-based cartridge A cartridge that moves data based upon a defined data model with relationships. This type of cartridge is typically used for ongoing extract operations.

OLTP database The online transaction processing database that typically is your active or source database.

pause One of the ways in which you can interrupt a business flow. Pausing suspends the business flow while awaiting operator intervention.

query server The component that provides SQL access to XML or CSV files.

remote deployment The generation and deployment of your cartridge or business flow to an environment on a system that is remote from your Designer client. Deployment files are generated locally and then deployed to the designated, remote environment.

repository	The location that holds business flow metadata, product configuration data, and data collected during runtime. The repository can be located on your active database, another logical database, or can be embedded database.
rule	Qualifications added to the model in order to include or exclude data based on certain criteria. For example, you might add a rule to exclude from extracting any orders that are not yet closed.
runtime parameter	A type of parameter that has its values set by the operator executing the job in Console or on the command line. Typically, this type of parameter represents operational values that tend to change frequently and therefore need to be set each time the job is run.
schema-based cartridge	A cartridge that moves data based upon the database schema rather than a defined data model with relationships. This type of cartridge is typically used for database retirement or the cleanup of orphan tables.
selection	The form of data selection to use (standard or advanced) for choosing data. When creating a cartridge or adding it to a business flow, you must specify the selection method.
source	The location (database) from which you are copying or moving data.
standard selection	A method of data selection that restricts itself to the rows identified by the model. Unlike advanced selection, it does not attempt to traverse related rows across multiple tables.
table use	A database table, view, or synonym that is referenced in Designer, for example, in the model. The same table can be used multiple times in a model. For example, a table could be appear as a transactional table and a lookup table in the same model.
target	The location (XML) to which you are copying data.
transactional data movement	Transactional movement uses set-based data movement and is the default method of movement.
transactional table	A table that contains information about the business transaction. For example, a transactional table might contain detailed tax or payment information related to each business transaction.
unique identifiers (UIDs)	A 16 hexadecimal identifier calculated based on the content of a Designer file. This value is used to determine if the user has customized key pieces of a project.
virtual constraint	A constraint that you define in Designer that only exists within HP Test Data Management as opposed to a database constraint, which exists within the database.
Web Console	A browser-based interface where you can create and manage your deployment environments, and deploy, run, administer, and monitor your business flows.

Index

U

W


Java™
COMPATIBLE