HP Test Data Management

Software version: 1.0

Runtime guide

Document release date: July 2010
Software release date: July 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

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java™ and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

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 build an extraction solution that copies or moves data out of your production database and into less expensive storage.

This guide provides information about:

- configuring the Web Console
- deploying business flows
- running business flows and jobs
- monitoring business flows and jobs

Intended audience

This guide is intended for:

- users configuring the Web Console
- users running business flows and jobs

Prerequisites

Prerequisites for installing 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.

- *HP Test Data Management Tutorial*

  Provides step-by-step instructions to build a sample test data module, deploy it, run it, and troubleshoot errors.

- *HP Test Data Management Developer's guide*

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

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

△ 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, go to:

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

Part I: Introduction

This part provides an introduction to the basics of data extraction from a database. It includes:

- Chapter 1, Introduction to test data management

1 Introduction to test data management

Test data management is a process where you typically copy a subset of data from your production database to a structured data file, such as an XML or comma separated values (CSV) file. Structured files provide open, standards-based formats that can be accessed using standard database mechanisms.

As part of the subsetting process, you perform the following:

- **Apply eligibility requirements to the data.** In most cases, you do not want all of the data from your production database. You need to define criteria that restrict the records copied. For example, you might only copy records that pertain to a certain period of time or customer.
- **Mask sensitive data.** Typically, your test database is not as secure as your production database. Hence, you need to mask any sensitive data, such as names, addresses, phone numbers, social security numbers, and so on. HP Test Data Management provides a variety of masking capabilities to help you protect confidential data.

Copying qualified records out of your active database and masking them is only the first leg in the lifecycle of test data management. Once you have the desired subset of data in a data extract file, you can upload from the file to a compatible test database. HP Test Data Management enables you to upload to heterogeneous databases, if necessary. For example, you may have extracted from a SQL Server database but need to upload to an Oracle database. You can also implement schema mapping in order to change the schema name when you load the data in the test database.

In addition to uploading the data from the extract file, you may also want to create a spreadsheet with some portion of the test data for use with other quality management tools. HP Test Data Management's query server enables you to quickly populate spreadsheets with test data from one or many database tables. Because you are using the extract file, the data you load into the spreadsheet is masked to your specifications and you can join tables, if necessary, without re-querying your source database.

This chapter includes:

See also For a complete conceptual introduction to Test Data Management, please refer to *HP Test Data Management Concepts guide*.

Before you begin

Before you begin performing the tasks in this guide, you should:

1. Review the *HP Test Data Management Concepts guide* to become familiar with the software and how you plan to use it.
2. Install HP Test Data Management according to the instructions in the *HP Test Data Management Installation guide*.
3. Go through *HP Test Data Management Tutorial*. The tutorial enables you to get hands on with the product quickly and exposes you to many of the most commonly used features.
4. Use the *HP Test Data Management Developer's guide* to design and develop or customize your test data solution.

Web Console overview

Most of the work of extracting data is performed in the Web Console. Web Console is a powerful browser-based interface used to:

- Configure the repository
- Create environments
- Create and modify users
- Deploy business flows
- Launch business flows to move data
- Monitor running business flows and their history

The Web Console enables you to remotely manage multiple environments for the extraction of data.

Part II: Tutorials

This part provides a set of tutorials that build on the basic tutorial described in the *HP Test Data Management Tutorial*. It includes:

- Chapter 2, Configuring the Web Console

2 Configuring the Web Console

Before you can deploy, run, and monitor your business flows to copy data, you must configure the Web Console. The Web Console is the primary runtime interface to Test Data Management.

This tutorial walks you through the process of configuring the Web Console, including:

- Starting and stopping the Web Console
- Installing the repository
- Creating environments
- Managing Web Console users.

NOTE Before you begin this tutorial, ensure that you have installed HP Test Data Management according to the *HP Test Data Management Installation guide*.

This chapter includes:

Starting the Web Console

The Web Console must be running before you can connect to it in your browser and perform operations such as installing the repository, creating environments, and managing users.

TIP When you install HP Test Data Management, the Installer gives you options to start the Web Console and launch its interface in your browser when the installation completes. If you chose those options and the Web Console is already running, you need not perform the steps in this section.

1 Start the Web Console in one of the following ways:

- On MS Windows, select **Start > HP Test Data Management > Start Web Console**.
- From the command line. For example, on MS Windows, select **Start > Run**, enter `cmd`, and click **OK**. Once you have accessed the command line, perform the following steps.

a. Change to the `bin` directory where you installed HP Test Data Management.

   On MS Windows:

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

   Unix:

   ```
   cd /home/HPDTDM/obt/bin
   ```

b. Enter the start command.

   On Windows:

   ```
   webConsole start
   ```

   On UNIX:

   ```
   ./webConsole.sh start
   ```

   A message should appear indicating that the Web Console has started and displaying its URL. For example:

   ```
   Starting the Web Console at http://localhost:8080/
   WebConsole
   ```

2. Launch your favorite browser, and connect to the Web Console using the following URL:

   ```
   http://localhost:8080/WebConsole
   ```

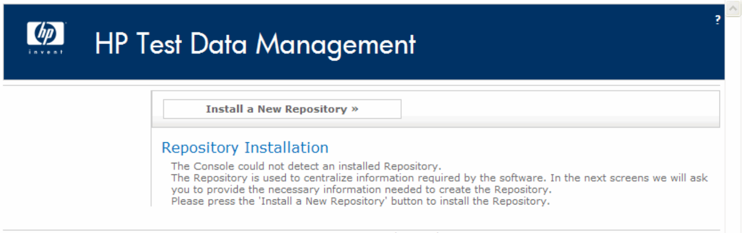
The first time that you start the Web Console after installing Test Data Management, you must install the repository. Refer to Installing the repository (page 18).

Installing the repository

Test Data Management stores metadata in a repository to facilitate the deployment and running of business flows, and administering the Web Console. You can create the repository in an Oracle or SQL Server database, or the embedded repository.

1. If the Web Console is not already started, start it according to the steps in Starting the Web Console (page 17).

   The Repository installation screen displays:



Repository Installation

The Console could not detect an installed Repository.
The Repository is used to centralize information required by the software. In the next screens we will ask you to provide the necessary information needed to create the Repository.
Please press the 'Install a New Repository' button to install the Repository.

2. Click **Install a New Repository**. The Repository Database: Administrator page displays.

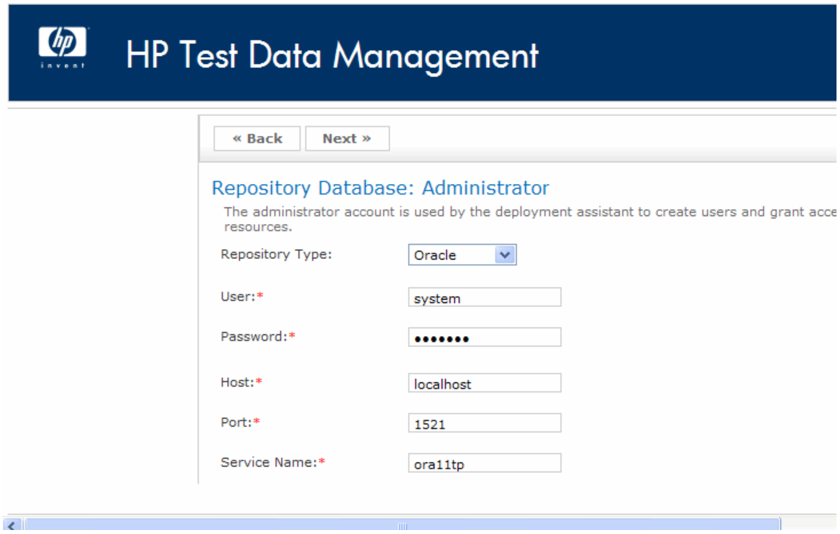
3 Select the repository type from the drop-down list:

- **Oracle** means you want to use an Oracle database and connect to it using the basic Oracle connection method.
- **SQL Server** means you want to use a SQL Server database.
- **Embedded** means that you want to embed the repository in Test Data Management and not use an external database for the repository.
- **JDBC URL** means you want to connect to a database using a JDBC URL string. One common usage of JDBC URL is to connect to an Oracle database with Real Application Clusters (RAC).

When you select a repository type, the properties of the Administrator page change accordingly. For example, when you choose SQL Server, a DB Server field is added, the default user changes to sa, and the default port changes to 1433.

For Oracle

An example for Oracle:



Repository Database: Administrator

The administrator account is used by the deployment assistant to create users and grant acce resources.

Repository Type: Oracle

User:* system

Password:* •••••••

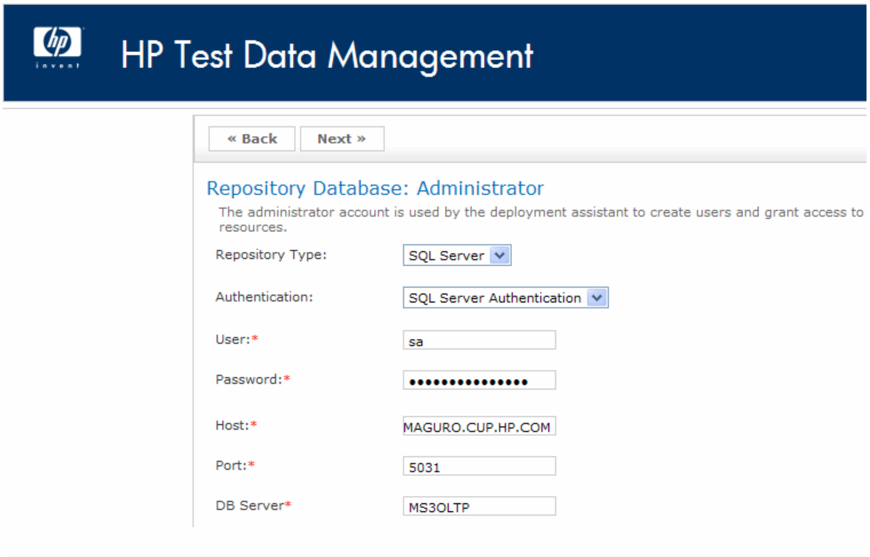
Host:* localhost

Port:* 1521

Service Name:* ora11tp

For SQL Server

An example for SQL Server 2008:



Repository Database: Administrator

The administrator account is used by the deployment assistant to create users and grant access to resources.

Repository Type:	SQL Server
Authentication:	SQL Server Authentication
User:*	sa
Password:*	•••••••••••••••
Host:*	MAGURO.CUP.HP.COM
Port:*	5031
DB Server*	MS3OLTP

4 Once you have selected the repository type, enter the appropriate credentials for that database type:

NOTE None of these properties applies to the Embedded option.

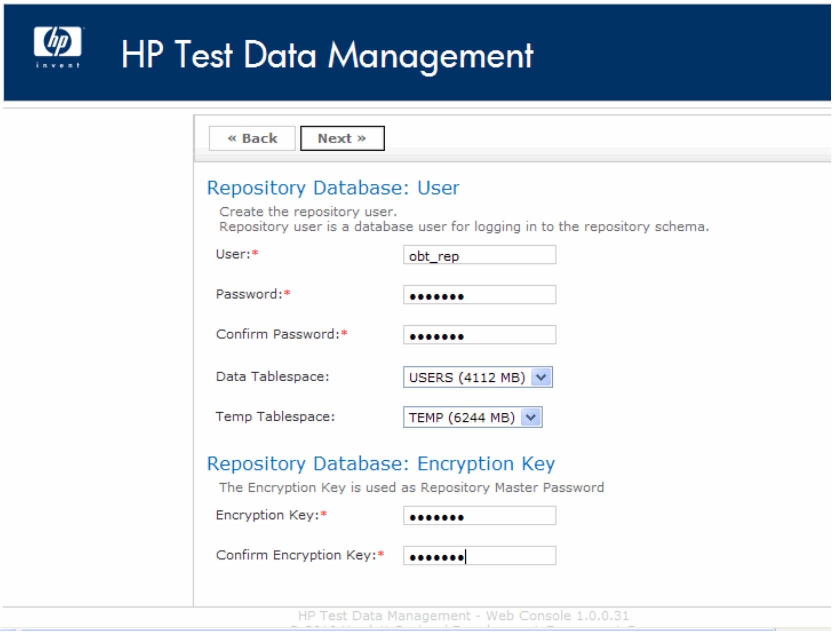
Field	Description
Authentication	(SQL Server only) Choose the type of authentication used by SQL Server: • Windows Authentication means that the SQL Server instance uses the same credentials as the Windows machine on which it is installed. • SQL Server Authentication
User	Enter the database administrator username. Note this option does not apply to embedded repositories.
Password	Enter the database administrator password. Note this option does not apply to embedded repositories.
Host	(Oracle and SQL Server) Enter the name of the machine where the database is installed.
Port	(Oracle and SQL Server) Enter the port number of your Oracle or SQL Server database.
Service Name	(Oracle only) Enter the name of your Oracle database, for example, ORCL.
DB Server	(SQL Server only) Enter the name of the SQL Server database, for example, MSOLTP.

5 Click **Next**.

The Repository Database: User page displays.

For Oracle

An example for Oracle:



HP Test Data Management

Repository Database: User

Create the repository user.
Repository user is a database user for logging in to the repository schema.

User:* obt_rep

Password:* •••••••

Confirm Password:* •••••••

Data Tablespace: USERS (4112 MB)

Temp Tablespace: TEMP (6244 MB)

Repository Database: Encryption Key

The Encryption Key is used as Repository Master Password

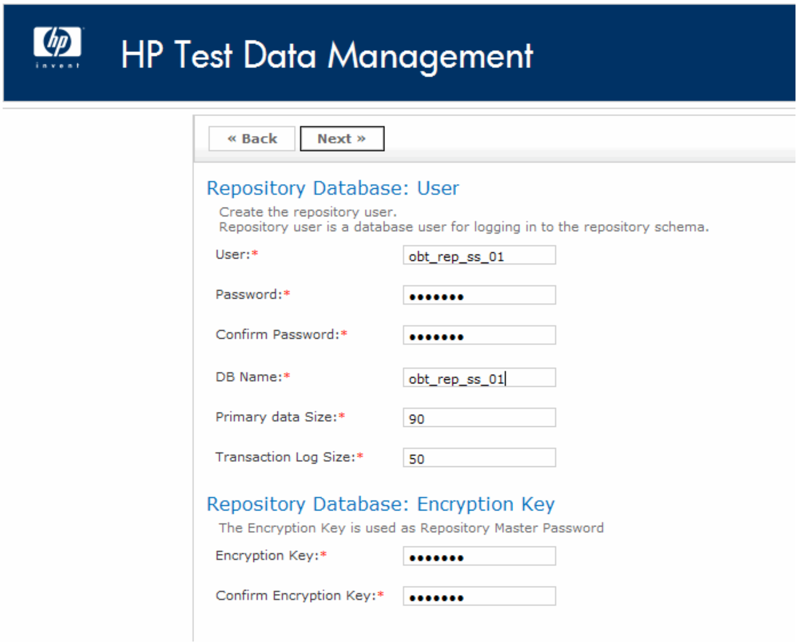
Encryption Key:* •••••••

Confirm Encryption Key:* •••••••

HP Test Data Management - Web Console 1.0.0.31

For SQL Server

An example for SQL Server:



HP Test Data Management

Repository Database: User

Create the repository user.
Repository user is a database user for logging in to the repository schema.

User:* obt_rep_ss_01

Password:* •••••••

Confirm Password:* •••••••

DB Name:* obt_rep_ss_01

Primary data Size:* 90

Transaction Log Size:* 50

Repository Database: Encryption Key

The Encryption Key is used as Repository Master Password

Encryption Key:* •••••••

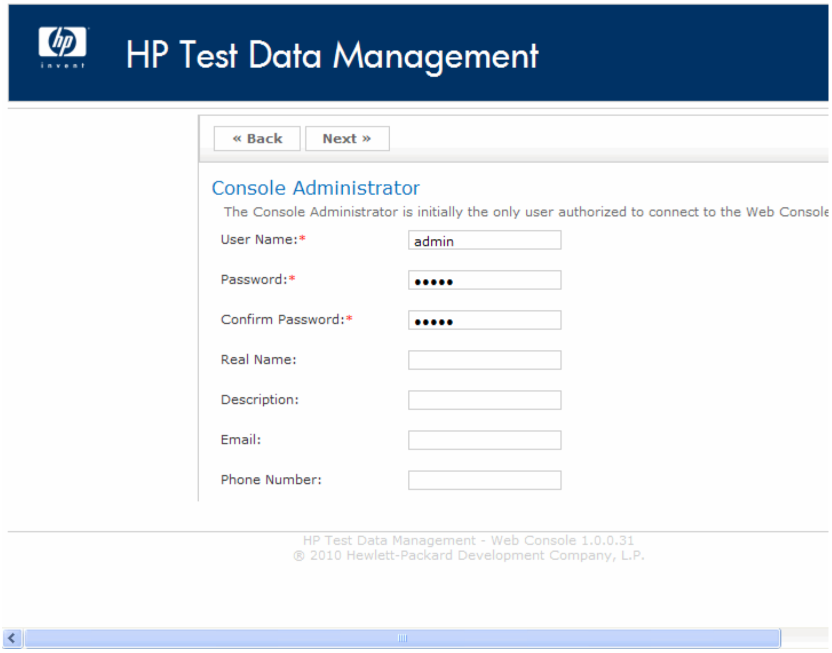
Confirm Encryption Key:* •••••••

6 Select or enter the following information for the repository user:

Field	Description
User	Enter the name you want to use for the repository user. The default value is obt_rep.
Password	Enter the password for the user you created.
Confirm Password	Enter the password again.
Data Tablespace	(Oracle only) Select the data tablespace you want to use.
Temp Tablespace	(Oracle only) Select the temporary tablespace you want to use.
Primary Data Size	(SQL Server only) Accept the default value or enter a maximum size in MB for the data file.
Transaction Log Size	(SQL Server only) Accept the default value or enter a maximum size in MB for the log file.
Encryption Key	Enter the encryption key you want to use.
Confirm Encryption Key	Enter the encryption key again.

7 Click **Next**.

The Console Administrator page displays.



Console Administrator

The Console Administrator is initially the only user authorized to connect to the Web Console

User Name:*	admin
Password:*	●●●●●
Confirm Password:*	●●●●●
Real Name:	
Description:	
Email:	
Phone Number:	

HP Test Data Management - Web Console 1.0.0.31
® 2010 Hewlett-Packard Development Company, L.P.

8 Select or enter the following information:

Field	Description
User Name	Enter the name you want to use for the Web Console administrator. The Web Console administrator is used to log on to the Web Console.
Password	Enter the password for the administrator.
Confirm Password	Enter the password again.
Real Name	Enter the name of the administrator user.
Description	Optionally, enter a description.
Email	Optionally, enter an email address.
Phone Number	Optionally, enter a phone number.

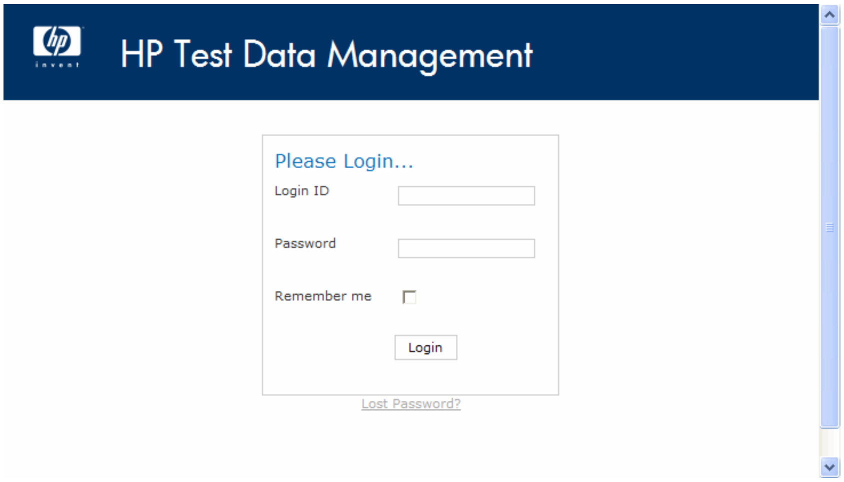
9 Click **Next**. The Summary page appears for your review.

10 Click **Finish**.

It may take several moments for the repository creation to complete. When the repository is created, the following message displays:

```
You have successfully installed the repository. Please do
NOT close your browser. Web Console will restart.
```

After the Web Console restarts, the login screen displays.



11 Enter the Web Console administrator user name and password that you just created.

12 Click **Login**.

You are now logged in to the Web Console and can create a new environment.

Creating an environment

Before you can deploy your business flows, you must create a deployment environment for them with the appropriate characteristics, such as the source database credentials and available data movement methods.

1 If the Web Console is not already started and open in your browser, you need to start it according to the instructions in Starting the Web Console (page 17).

2 If you have not previously installed the repository, you need to do so before you can continue. Refer to Installing the repository (page 18).

3 If you have not previously created an environment, the Web Console automatically prompts you to create one when you open it. Otherwise, click **Environment** at the top of the page and then **New** in the left navigation pane to display the New Environment wizard.

4 Specify a name for the new environment, for example, `Oracle_OLTP` or `SQLServer_OLTP`.

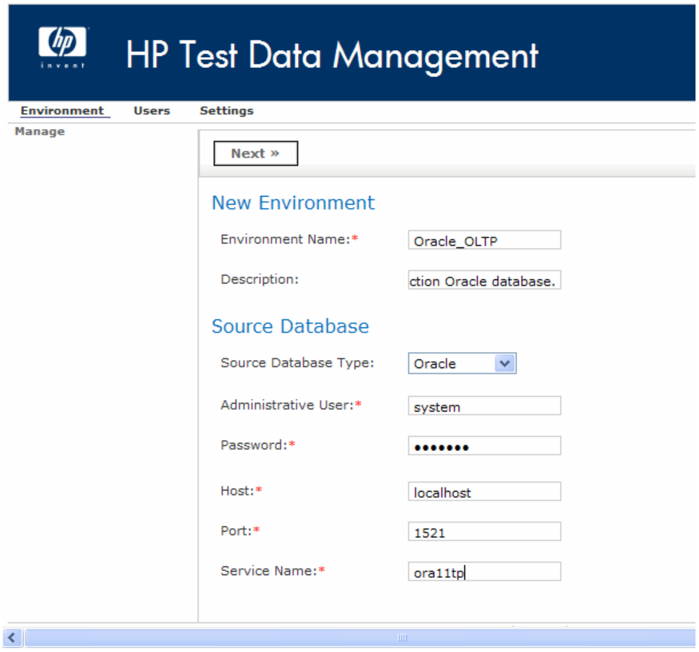
5 Select the source database type.

- **Oracle**
- **SQL Server**
- **Sybase**
- **JDBC URL**, which means you want to connect to a database using a JDBC URL string. One common usage of JDBC URL is to connect to an Oracle database with Real Application Clusters (RAC).

6 Enter the necessary credentials for the source database.

For Oracle

An example for Oracle:



New Environment

Environment Name:* Oracle_OLTP

Description: ction Oracle database.

Source Database

Source Database Type: Oracle

Administrative User:* system

Password:* •••••••

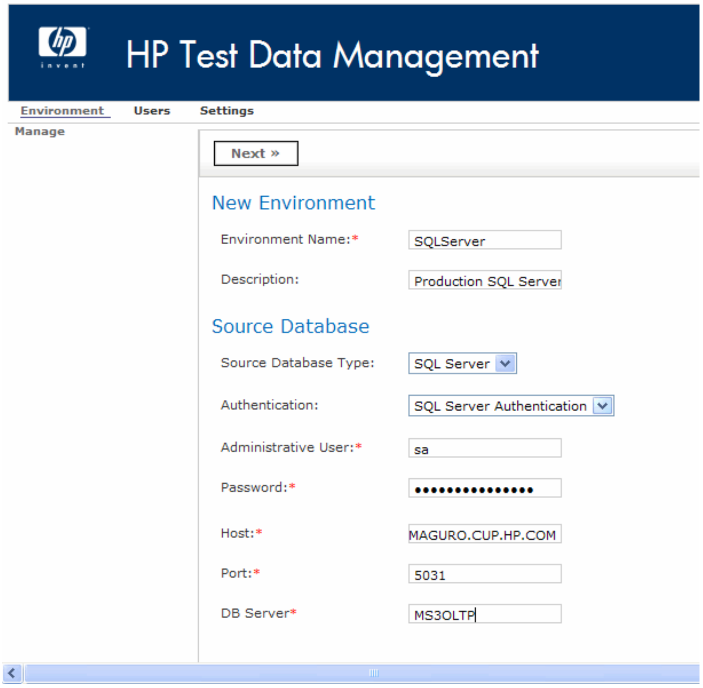
Host:* localhost

Port:* 1521

Service Name:* ora11tp

For SQL Server

An example for SQL Server:



New Environment

Environment Name:* SQLServer

Description: Production SQL Server

Source Database

Source Database Type: SQL Server

Authentication: SQL Server Authentication

Administrative User:* sa

Password:* •••••••••••••••

Host:* MAGURO.CUP.HP.COM

Port:* 5031

DB Server* MS3OLTP

7 Click **Next**.

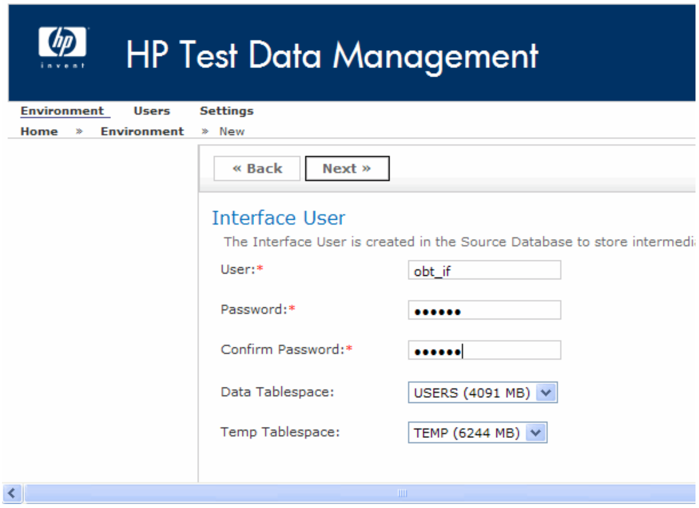
8 Select where to store your user indexes.

Any user indexes you create in Designer using the File Indexes tab are saved to the location you specify.

9 Click **Next**.

The Interface User page displays.

For Oracle An example for Oracle:



Interface User

The Interface User is created in the Source Database to store intermedi

User:* obt_if

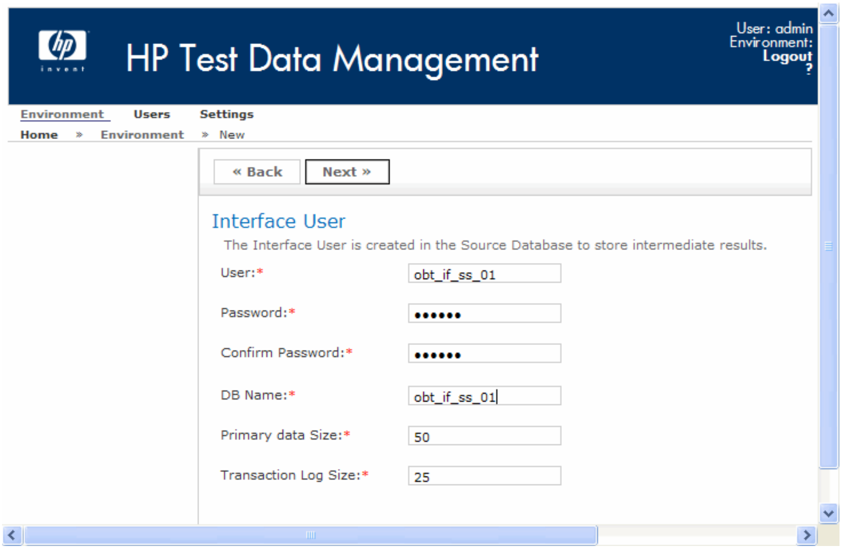
Password:* ••••••

Confirm Password:* ••••••

Data Tablespace: USERS (4091 MB)

Temp Tablespace: TEMP (6244 MB)

For SQL Server An example for SQL Server 2008:



Interface User

The Interface User is created in the Source Database to store intermediate results.

User:*	obt_if_ss_01
Password:*	••••••
Confirm Password:*	••••••
DB Name:*	obt_if_ss_01
Primary data Size:*	50
Transaction Log Size:*	25

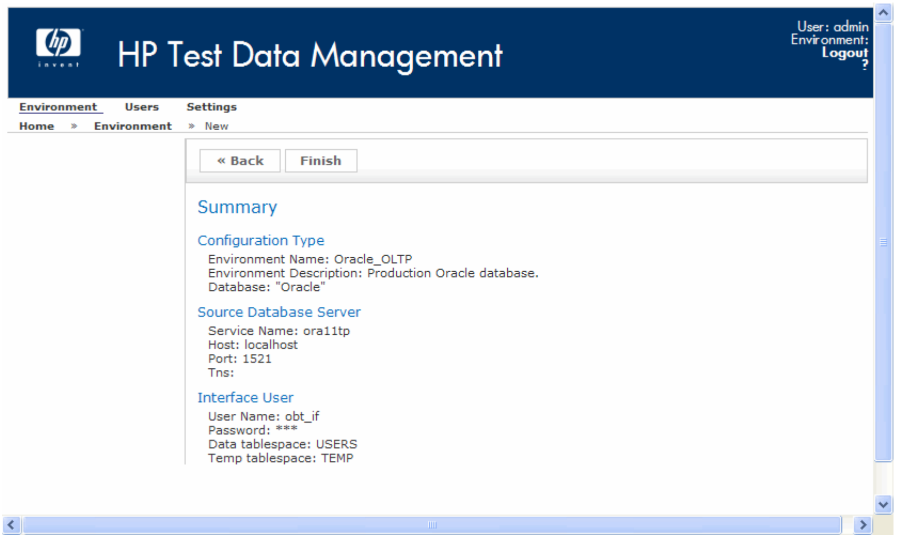
10 Enter the following information for the interface user:

Field	Description
User	Type the name you want to use for the Interface user. The default value is obt_if.
Password	Type the password for the user you created.
Confirm Password	Type the password again.
Data Tablespace	(Oracle only) Select the data tablespace you want to use.
Temp Tablespace	(Oracle only) Select the temporary tablespace you want to use.
Primary Data Size	(SQL Server only) Accept the default value or enter a maximum size in MB for the data file.
Transaction Log Size	(SQL Server only) Accept the default value or enter a maximum size in MB for the log file.

11 Click **Next**.

The summary page for the environment appears. Review the settings that you have chosen.

For Oracle An example for Oracle:



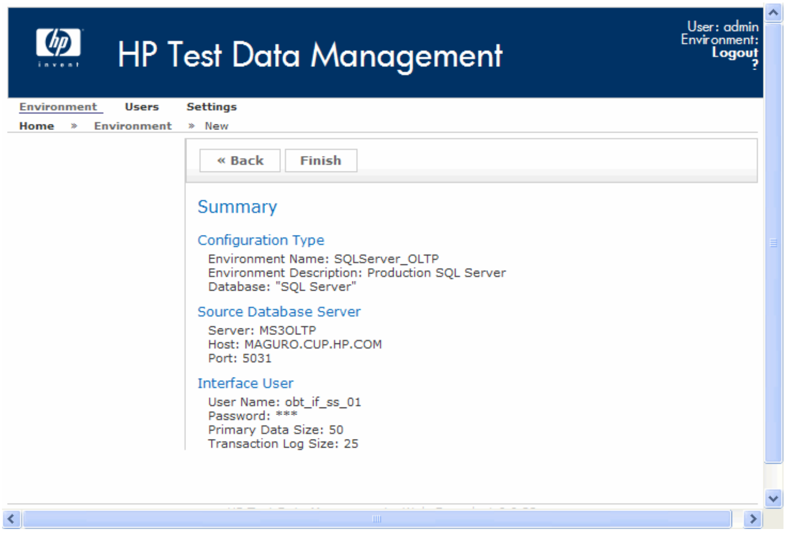
Summary

Configuration Type
Environment Name: Oracle_OLTP
Environment Description: Production Oracle database.
Database: "Oracle"

Source Database Server
Service Name: ora11tp
Host: localhost
Port: 1521
Tns:

Interface User
User Name: obt_if
Password: ***
Data tablespace: USERS
Temp tablespace: TEMP

For SQL Server An example for SQL Server 2008:



Summary

Configuration Type
Environment Name: SQLServer_OLTP
Environment Description: Production SQL Server
Database: "SQL Server"

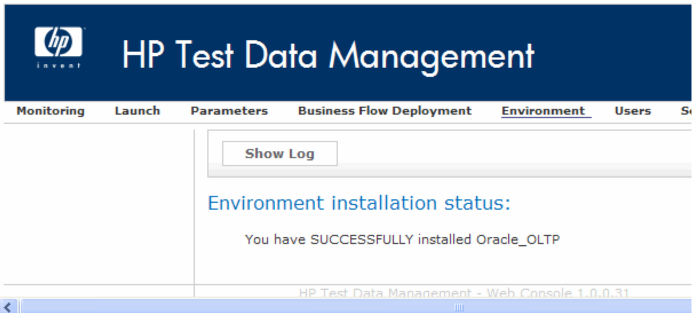
Source Database Server
Server: MS3OLTP
Host: MAGURO.CUP.HP.COM
Port: 5031

Interface User
User Name: obt_if_ss_01
Password: ***
Primary Data Size: 50
Transaction Log Size: 25

12 Click **Finish** to create the environment.

It may take several moments to create the environment. When the environment is ready, the success message should appear at the top of the page. If you receive any errors, use the back button on your browser to review your choices and ensure that they are correct.

For Oracle An example for Oracle:

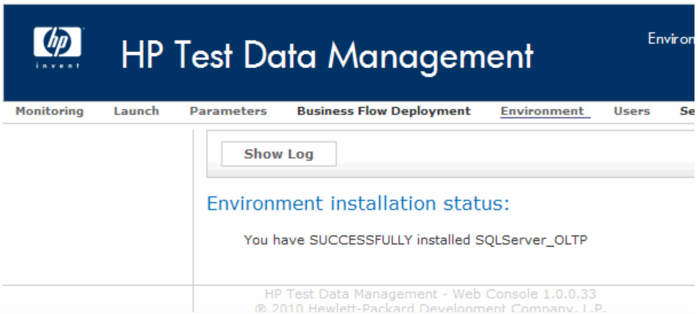


Environment installation status:

You have SUCCESSFULLY installed Oracle_OLTP

HP Test Data Management - Web Console 1.0.0.31

For SQL Server An example for SQL Server 2008:



Environment installation status:

You have SUCCESSFULLY installed SQLServer_OLTP

HP Test Data Management - Web Console 1.0.0.33
® 2010 Hewlett-Packard Development Company, L.P.

For the purposes of this tutorial, you can now optionally create a Web Console user Creating Web Console users (optional) (page 28).

Creating Web Console users (optional)

In most cases, more than one user needs to perform functions in the Web Console, but not every user needs the same privileges. For example, some users might only need to run jobs while others may need to deploy business flows, set parameter values, and run jobs. Web Console provides a user model that enables you to create as many users as you need, and grant or withhold privileges.

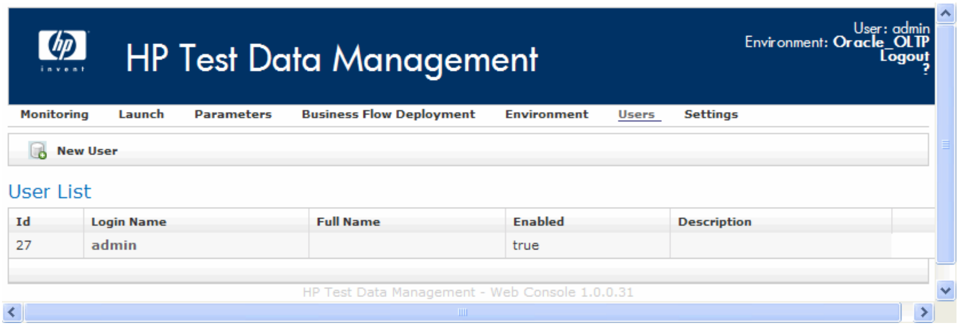
In this section:

- Creating a new Web Console user (page 28)
- Verifying the new Web Console user (page 31)

Creating a new Web Console user

1. If the Web Console is not started and open in your browser, refer to Starting the Web Console (page 17).
2. If you have not previously installed the repository and created an environment, you need to do so before you can continue. Refer to Installing the repository (page 18) and Creating an environment (page 24).
3. In the menu bar at the top of the page, select **Users**.

   The User List page displays. If you have not previously created any other users, you should only see the user named admin.

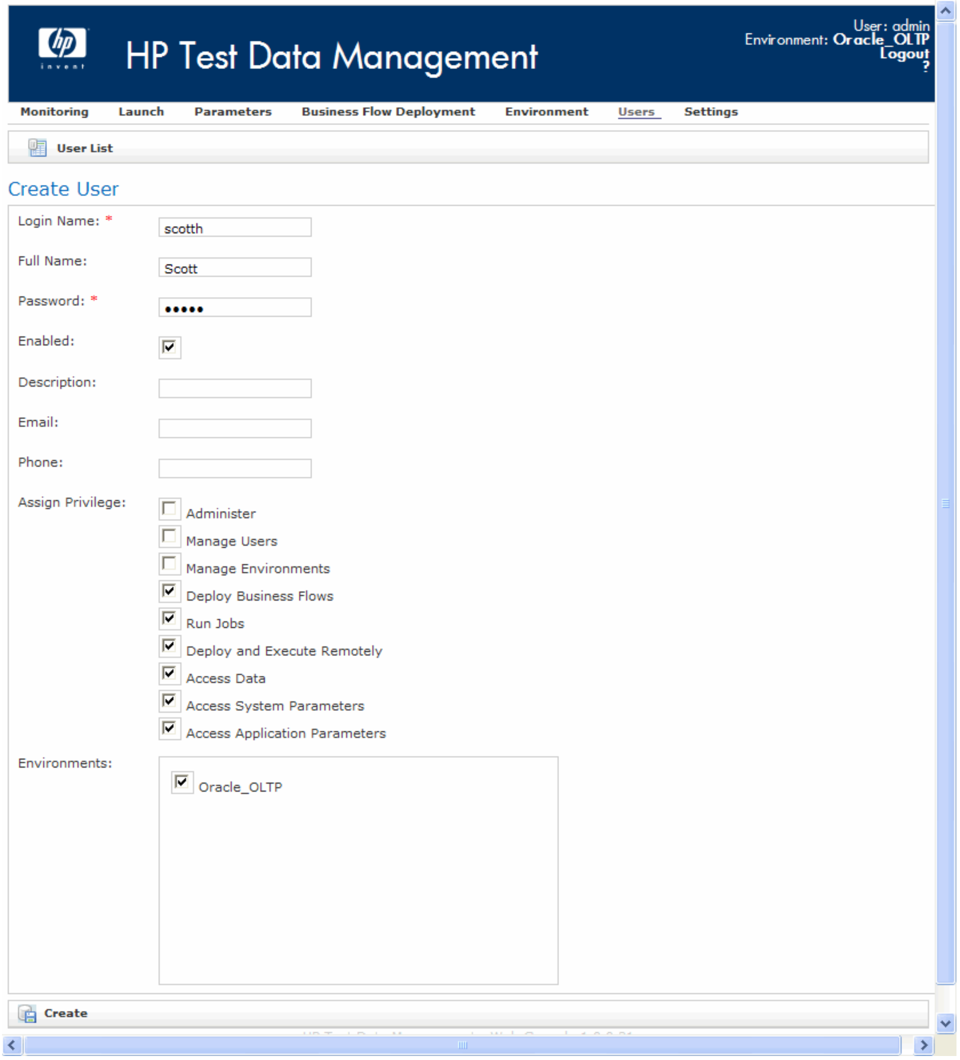


User List

Id	Login Name	Full Name	Enabled	Description
27	admin		true	

4 Select **New User**.

The Create User page displays.



Create User

Login Name: * scotth

Full Name: Scott

Password: * •••••

Enabled: ☑

Description:

Email:

Phone:

Assign Privilege:
- □ Administer
- □ Manage Users
- □ Manage Environments
- ☑ Deploy Business Flows
- ☑ Run Jobs
- ☑ Deploy and Execute Remotely
- ☑ Access Data
- ☑ Access System Parameters
- ☑ Access Application Parameters

Environments:
- ☑ Oracle_OLTP

5 Enter at least the following values:

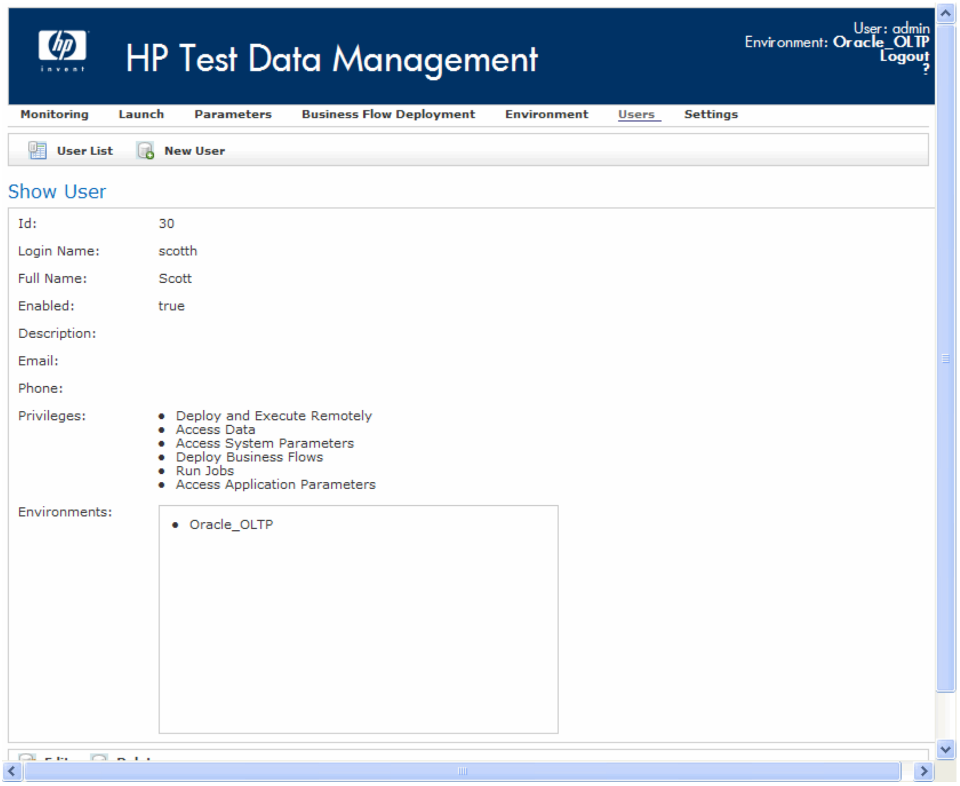
- Login Name: tom_baker
- Full Name: Tom Baker
- Password: welcome
- Enabled: checked

- Assign Privilege: Check the following:
    - — Deploy Business Flows
    - — Run Jobs
    - — Deploy and Execute Remotely
    - — Access Data
    - — Access System Parameters
    - — Access Application Parameters

> NOTE This new user is only expected to deploy, run, and monitor business flows in the specified environment. Hence, we do not grant this user the privileges to create users, manage environments, or otherwise administer the Web Console.

6. Select the environment you want the user to access.
7. Click **Create** at the bottom of the page.

   The Show User page displays.



Show User

Id: 30

Login Name: scotth

Full Name: Scott

Enabled: true

Description:

Email:

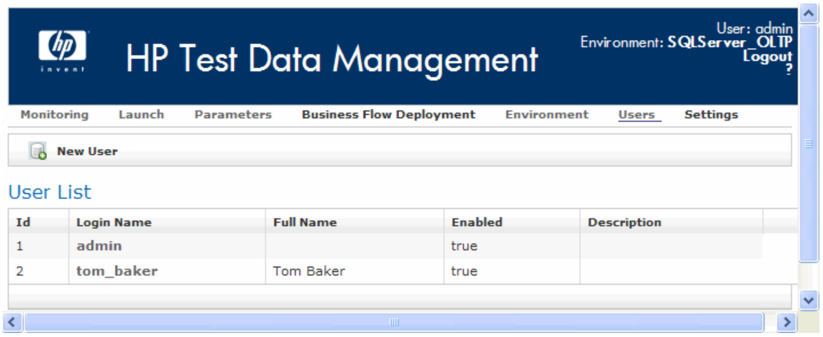
Phone:

Privileges:
- Deploy and Execute Remotely
- Access Data
- Access System Parameters
- Deploy Business Flows
- Run Jobs
- Access Application Parameters

Environments:
- Oracle_OLTP

The new user can deploy, run, and monitor business flows, and set parameters.

8. Click **User List** to confirm that the new user has been added.



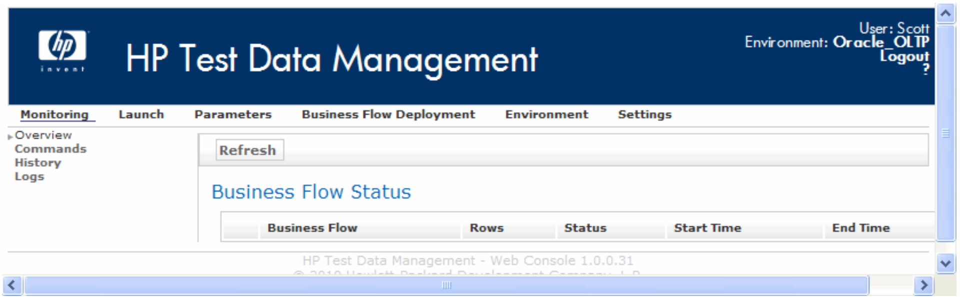
User List

Id	Login Name	Full Name	Enabled	Description
1	admin		true	
2	tom_baker	Tom Baker	true	

Verifying the new Web Console user

1 Click **Logout**.

2 Log back in as the user you just created.



Business Flow Status

Business Flow	Rows	Status	Start Time	End Time

HP Test Data Management - Web Console 1.0.0.31

Notice how the User link does not appear to Tom Baker. He does not have permissions to create users.

3 Click **Logout**.

4 Log back in as the Web Console administrator.

Part III: Task reference

This part provides an advanced task reference to assist you in performing specific tasks within the Web Console and from the command line. It includes:

3 Starting and configuring the Web Console

The Web Console is the primary runtime interface to Test Data Management. You use it to deploy, run, and monitor jobs, and administer environments and users. The Web Console requires a repository in which to store its metadata. Hence, the first time you start the Web Console, you are prompted to install or upgrade the repository.

In this chapter

- Starting and stopping the Web Console (page 35)
- Installing a new repository (page 38)
- Adding Web Console nodes (page 42)

Starting and stopping the Web Console

Before you can connect to its interface in your Web browser and perform tasks, you must start Web Console. You can also stop Web Console when it is not in use or when performing maintenance activities that require it be taken down.

In this section

- Starting Web Console (page 35)
- Stopping Web Console (page 37)

Starting Web Console

You can start Web Console from the Start menu on MS Windows or from the command line on any platform.

In this section

- Starting Web Console from the Start menu (page 35)
- Starting Web Console from the command line (page 36)
- Starting Web Console on a different port (page 36)
- Starting Web Console with SSL enabled (page 37)

Starting Web Console from the Start menu

1. After installing Test Data Management, start the Web Console from the Start menu:

   Start > All Programs > HP Test Data Management > Start Web Console

   where HP Test Data Management is the program group where you chose to install Test Data Management during installation.

A command window opens and closes automatically after the Web Console is started.

2 Launch your favorite browser, and connect to the Web Console using the following default URL:

```
http://<hostname>:8080/WebConsole
```

where <hostname> is the name of the machine on which you installed Test Data Management.

3 If you are starting Web Console for the first time after installation of Test Data Management, you are prompted to either install a new repository or upgrade an existing one. Refer to Installing a new repository (page 38).

Starting Web Console from the command line

1 Open a command window and navigate to the directory that contains the Web Console script.

```
cd <install_directory>/obt/bin
```

where <install_directory> is the location where you installed Test Data Management.

2 To start the Web Console server, enter the following command:

On Windows:

```
webConsole start
```

On UNIX:

```
./webConsole.sh start
```

3 Launch your favorite browser, and connect to the Web Console using the following default URL:

```
http://<hostname>:8080/WebConsole
```

where <hostname> is the name of the machine on which you installed Test Data Management.

4 If you are starting Web Console for the first time after installation of Test Data Management, you are prompted to either install a new repository or upgrade an existing one. Refer to Installing a new repository (page 38).

Starting Web Console on a different port

By default, the Web Console runs on port 8080. If you want to use a different port number, you must start the Web Console server from the command line and specify a different port number.

1 To start Web Console using a different port number, enter the following command:

On Windows:

```
webConsole -p <port> start
```

On UNIX:

```
./webConsole.sh -p <port> start
```

where <port> is the port number you plan to use for Web Console.

2 Launch your favorite browser, and connect to Web Console using the following URL:

```
http://<hostname>:<port>/WebConsole
```

where:

<hostname> is the name of the machine on which you installed Test Data Management.

<port> is the port number you specified in the previous step.

3 If you are starting Web Console for the first time after installation of Test Data Management, you are prompted to either install a new repository or upgrade an existing one. Refer to Installing a new repository (page 38).

Starting Web Console with SSL enabled

You can configure Web Console to run with SSL using the XML configuration file (jetty.xml) for its application server, called Jetty. Refer to the following URL for more information about Jetty and how to configure it for SSL:

```
http://docs.codehaus.org/display/JETTY/Jetty+Wiki
```

If you choose to modify jetty.xml, you can no longer start the Web Console using the Start menu or command line scripts supplied with the product. You instead must use the following:

```
java -jar webConsole/start.jar
```

Stopping Web Console

You can stop Web Console from the Start menu on MS Windows or from the command line on any platform.

In this section

- Stopping Web Console from the Start menu (page 37)
- Stopping Web Console from the command line (page 38)
- Stopping Web Console on a different port (page 38)

Stopping Web Console from the Start menu

1 Stop Web Console from the Start menu:

Start > All Programs > <HPDB_Pgm_Group> > Stop Web Console

where **<HPDB_Pgm_Group>** is the program group where you chose to install Test Data Management during installation.

A command window opens and closes automatically after the Web Console is started.

Stopping Web Console from the command line

1. Stop Web Console using the following command:

   ```
   webConsole stop
   ```

Stopping Web Console on a different port

If you started the Web Console server using a different port than 8080, then you must stop the Web Console from the command line and specify that port number.

1. To stop the Web Console using a different port number, enter the following command:

   ```
   webConsole -p <port> stop
   ```

   where <port> is the port number you used for Web Console.

Installing a new repository

You can create a repository on an Oracle or SQL Server database, or the embedded repository. If you are creating the repository on Oracle or SQL Server, you have the option to connect with the standard Oracle or SQL Server connection formats, or to use a JDBC URL.

1. If it is not already running, start the Web Console.

   See also Starting Web Console (page 35)

2. Click **Install a New Repository**.

   The Repository Database: Administrator page displays.

3. Select the RDBMS type for your repository and enter the appropriate information:

   — Oracle (page 38)

   — SQL Server (page 40)

   — Embedded (page 41)

   — JDBC URL (page 41)

Oracle

a. Enter the following administrator information:

Field	Description
User	Enter the administrator username.
Password	Enter the password for the specified user.

Field	Description
Host	Enter the name of the machine where your Oracle database is installed.
Port	Enter the port number of your Oracle database, for example, 1521.
Service Name	Enter the name of your Oracle database, for example, ORCL.

b Click **Next**.

The Repository Database: User page displays.

c Select or enter the following information for the repository user:

Field	Description
User	Enter the name you want to use for the repository user. The default value is obt_rep.
Password	Enter the password for the user you created.
Confirm Password	Enter the password again.
Data Tablespace	Select the data tablespace you want to use.
Temp Tablespace	Select the temporary tablespace you want to use.
Encryption Key	Enter the encryption key you want to use.
Confirm Encryption Key	Enter the encryption key again.

SQL Server

a Enter or select the following administrator information:

Field	Description
Authentication	Select SQL Server Authentication or Windows Authentication from the list of values. • **SQL Server Authentication** indicates that 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. • **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.
User	Enter the administrator username. Required for SQL Server authentication only.
Password	Enter the administrator password for the specified user. Required for SQL Server authentication only.
Host	Enter the host name of the machine where your SQL Server database is installed.
Port	Enter the port number of your SQL Server database.
DB Server	Enter the name of your SQL Server database server.

b Click **Next**.

The Repository Database: User page displays.

c Select or enter the following information for the repository user:

Field	Description
User	Enter the name you want to use for the repository user. The default value is obt_rep.
Password	Enter the password for the user you created.
Confirm Password	Enter the password again.
Primary Data Size	Accept the default value or enter a maximum size in MB for the data file.
Transaction Log Size	Accept the default value or enter a maximum size in MB for the log file.
Encryption Key	Enter the encryption key you want to use.
Confirm Encryption Key	Enter the encryption key again.

Embedded

a. Click **Next** to start the embedded repository.

   The Repository Database: User page displays.

b. Select or enter the following information for the repository user:

Field	Description
User	Enter the name you want to use for the repository user. The default value is obt_rep.
Password	Enter the password for the user you created.
Confirm Password	Enter the password again.
Encryption Key	Enter the encryption key you want to use.
Confirm Encryption Key	Enter the encryption key again.

JDBC URL

a. Enter the following administrator information:

Field	Description
User	Enter the administrator username.
Password	Enter the administrator password for the specified user.
URL	Enter the URL for the database.

b. Click **Next**.

   The Repository Database: User page displays.

c. Select or enter the following information for the repository user:

Field	Description
User	Enter the name you want to use for the repository user. The default value is obt_rep.
Password	Enter the password for the user you created.
Confirm Password	Enter the password again.
Data Tablespace	Select the data tablespace you want to use.
Temp Tablespace	Select the temporary tablespace you want to use.
Encryption Key	Enter the encryption key you want to use.
Confirm Encryption Key	Enter the encryption key again.

4. Click **Next**.

The Console Administrator page displays.

5 Enter the following information:

Field	Description
User Name	Enter the name you want to use for the Web Console administrator. The Web Console administrator is used to log on to the Web Console.
Password	Enter the password for the administrator.
Confirm Password	Enter the password again.
Real Name	Optionally, enter the name of the administrator user.
Description	Optionally, enter a description.
Email	Optionally, enter an email address.
Phone Number	Optionally, enter a phone number.

6 Click **Next**.

The Summary page displays.

7 Click **Finish** to create the repository, or click **Back** to make changes.

The following message displays:

```
You have successfully installed the repository. Please do
not close your browser. Web Console will restart.
```

After the Web Console restarts, the login screen displays.

Adding Web Console nodes

In some cases, you may want to have additional Web Console nodes on other machines that operate upon the same repository. Having multiple Web Console nodes enables you to perform operations against the same repository from any one of several different machines, thus avoiding any performance bottlenecks on the primary Web Console node. By copying certain files from your first or primary node of the Web Console, you can effectively clone the primary node on other machines.

To configure an additional Web Console node:

NOTE This procedure assumes that you have already installed and fully configured one instance of Test Data Management as per the instructions in the *HP Test Data Management Installation guide* and Installing a new repository (page 38).

1. From the primary node, where you first installed Test Data Management and configured the Web Console, make copies of the following files and move them to a location accessible to the machine where you want to create an additional node:
   - `<prime_install_dir>\obt\config\connection-sources.xml`
   - `<prime_install_dir>\obt\config\hpdbarchiving.auth`
   - `<prime_install_dir>\obt\sql_tuning.properties`
2. On the machine where you want to create the additional Web Console node, use the Installer to create and populate an installation directory tree just as you would for a new installation. Follow the instructions in the *HP Test Data Management Installation guide*, but do not start the Web Console.
3. Overwrite the following files on the clone machine with the copies you made from the primary node:
   - `<clone_install_dir>\obt\config\connection-sources.xml`
   - `<clone_install_dir>\obt\config\hpdbarchiving.auth`
   - `<clone_install_dir>\obt\sql_tuning.properties`
4. Start the cloned Web Console node as described in Starting and stopping the Web Console (page 35).
5. Confirm that you are operating upon the same repository as the primary node. You should not be prompted to install or upgrade a repository as you normally would for a new installation of the Web Console. Furthermore, if you are logged in as the administrator user, you should be able to see any environments that were created from the primary node.

4 Running business flows and jobs from the Web Console

HP Test Data Management uses business flows to execute your logic for copying test data from the database. After copying, the post-copy jobs allow you to view and manipulate your data. After copying data to structured files, you can enable transparent access to your test data. After copying to file, you can query against, view, and relocate the XML data.

In this chapter

Before you begin

Before you run business flows, you should perform some pre-flight checks, for example, confirming parameter settings. Beyond that, before you run a business flow, you must deploy it.

Running business flows

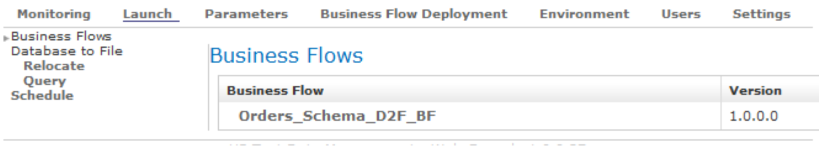
To run a business flow from the Web Console:

1 Start the Web Console, if it is not already started.

See also Starting Web Console (page 35)

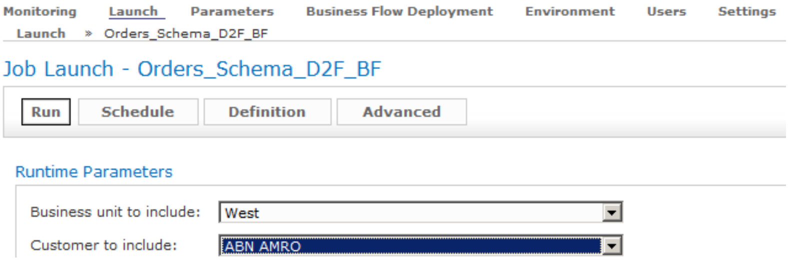
2 Perform any needed pre-flight checks and preparatory steps. Refer to Before you begin (page 45).

3 Click **Launch** from the menu at the top of the page. The Business Flows page displays the list of business flows you can run.



Business Flow	Version
Orders_Schema_D2F_BF	1.0.0.0

4 Select the business flow you want to run. The Job Launch page displays for the selected business flow.



Job Launch - Orders_Schema_D2F_BF

Runtime Parameters

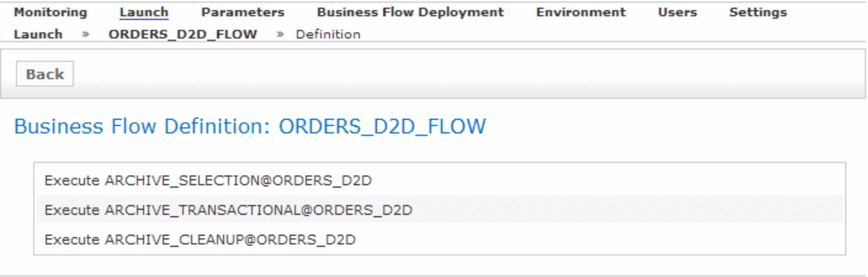
Business unit to include: West

Customer to include: ABN AMRO

5 Enter or select the values for the runtime parameters.

TIP If the parameter value is left blank, and a default value was defined in Designer, then the default value is used. If there is no default value, then NULL is used.

6 Optionally click **Definition** to view the business flow definition page.



Business Flow Definition: ORDERS_D2D_FLOW

Execute ARCHIVE_SELECTION@ORDERS_D2D
Execute ARCHIVE_TRANSACTIONAL@ORDERS_D2D
Execute ARCHIVE_CLEANUP@ORDERS_D2D

7 Click **Back** to return to the Job Launch page.

8 Optionally click **Advanced** to view the following sections:

- Command Line—displays the correct job syntax for the parameter values entered. You can copy the text directly if you plan to run the job from the command line.
- Business Flow Parameters—displays any business flow parameters that exist for the business flow. The user-defined business flow parameters are created in Designer as configuration parameters, and can be modified using the Parameters - Business Flows page.
- Dynamic Parameters—displays any dynamic parameters that exist for the business flow. Dynamic parameters are created in Designer and the values are set during deployment.

9 Click **Run** to run the job immediately.

TIP If you prefer to schedule the job for a later time rather than run it immediately, refer to Scheduling business flows (page 47).

10 Click **Confirm** to confirm the job.

11 Optionally, you can monitor the job status. Refer to Chapter 6, Monitoring jobs.

Scheduling jobs

In addition to running jobs immediately, you can use the Web Console to schedule business flows and jobs, view the existing scheduled jobs, and alter the schedule.

This section includes

- Scheduling business flows (page 47)
- Editing scheduled business flows (page 47)

Scheduling business flows

To schedule a business flow:

1 Start the Web Console, if it is not already started.

See also Starting Web Console (page 35)

2 Perform any needed pre-flight checks and preparatory steps. Refer to Before you begin (page 45).

3 Click **Launch** from the menu at the top of the page. The Business Flows page displays the list of business flows you can run.

4 Select the business flow you want to run.

5 Enter or select the values for the runtime parameters.

6 Click **Schedule**. The Schedule Job Launch page displays.

7 Enter a title and optional description for the business flow.

8 Ensure the **Enabled** is checked to enable the job for running.

9 Choose how often you want the business flow to run:

- **Run Once** means you want to run the business flow once at a specified date and time. Enter the date and time you want the job to run.
- **Monthly** means that you want to run the business flow once per month. Choose a day of the month and enter the time you want the job to run.
- **Weekly** means that you want to run the business flow once per week. Choose a day of the week and enter the time you want the job to run.
- **Daily** means that you want the business flow to run once per day. Enter the time you want the job to run.

10 Click **Parameters** to ensure the parameters you choose are correct.

11 Click **Save** to save the scheduled business flow.

Editing scheduled business flows

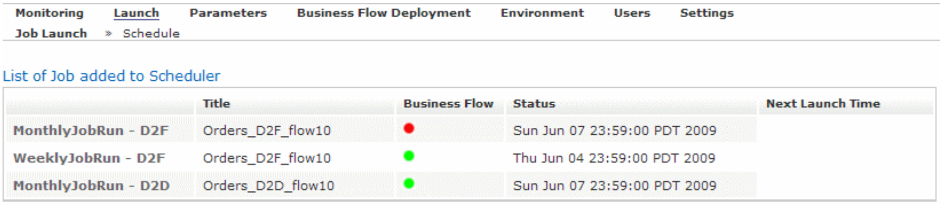
You can view a list of all scheduled jobs, and edit or delete any of them.

1 Start the Web Console, if it is not already started.

See also Starting Web Console (page 35)

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

3 Click **Schedule** from the left navigation pane. The Scheduler displays the list of scheduled jobs.



List of Job added to Scheduler

	Title	Business Flow	Status	Next Launch Time
MonthlyJobRun - D2F	Orders_D2F_flow10	●	Sun Jun 07 23:59:00 PDT 2009	
WeeklyJobRun - D2F	Orders_D2F_flow10	●	Thu Jun 04 23:59:00 PDT 2009	
MonthlyJobRun - D2D	Orders_D2D_flow10	●	Sun Jun 07 23:59:00 PDT 2009	

4 Click the Title of a job to edit it. The Schedule page displays.

5 Edit the values as desired.

6 Click **Save** to save the edited values for the scheduled job.

TIP To delete a job, click **Delete**.

Recovering or cancelling business flows and jobs

In general, it is best practice to never leave a business flow or job in one of the following states:

- Failed
- Suspended
- Not Responding

Otherwise, you may get unexpected results when you attempt to run the same or other, related business flows, or re-deploy the business flow.

TIP To check the status of a business flow or job, see Viewing a status overview (page 61).

If a business flow does not complete successfully, you can use the Web Console to:

- **Recover it.** Recovering the job or business flow reruns it using the same parameters as the first run. If possible, recovering is the best option.
- **Cancel it.** Cancelling the job or business flow should be your option of last resort. If you cannot recover for some reason and you need to redeploy the business flow, you can cancel the business flow. Cancelling updates the status of the business flow or job, but does not clean up any of the selection tables. Consequently, cancelling a job can lead to inconsistent results.

WARNING! If a business flow containing an advanced selection database to file cartridge does not complete successfully, see the troubleshooting guide for further information before rerunning the business flow.

Recovering business flows and jobs

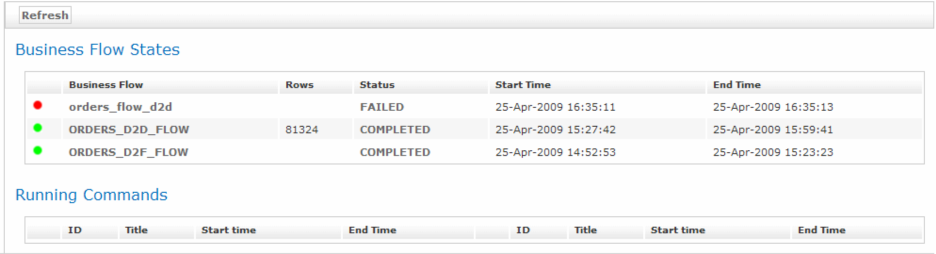
To recover a business flow or job:

1 Start the Web Console, if it is not already started.

See also Starting Web Console (page 35)

2 Click **Monitoring** from the menu at the top of the page.

3 Select a business flow in Failed, Suspended, or Not Responding status.



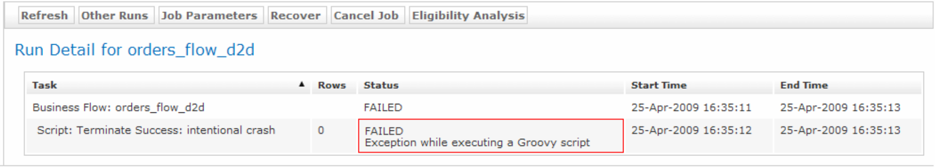
Business Flow States

	Business Flow	Rows	Status	Start Time	End Time
●	orders_flow_d2d		FAILED	25-Apr-2009 16:35:11	25-Apr-2009 16:35:13
●	ORDERS_D2D_FLOW	81324	COMPLETED	25-Apr-2009 15:27:42	25-Apr-2009 15:59:41
●	ORDERS_D2F_FLOW		COMPLETED	25-Apr-2009 14:52:53	25-Apr-2009 15:23:23

Running Commands

ID	Title	Start time	End Time	ID	Title	Start time	End Time

The Run Details page displays.



Run Detail for orders_flow_d2d

Task	Rows	Status	Start Time	End Time
Business Flow: orders_flow_d2d		FAILED	25-Apr-2009 16:35:11	25-Apr-2009 16:35:13
Script: Terminate Success: intentional crash	0	FAILED Exception while executing a Groovy script	25-Apr-2009 16:35:12	25-Apr-2009 16:35:13

4 Click **Recover**. The recovery job starts.

5 Click **Monitoring** to view the status of the job. If the recovery is successful, the job status will change to COMPLETED. If the recovery is unsuccessful, you will need to cancel the job.

Cancelling business flows and jobs

To cancel a business flow or job:

1 If the job to be cancelled is a database to file job that employed table parallel movement, then you must perform the following steps to cleanup incomplete transactions before proceeding:

   a Identify the rows that needs to be deleted from History by executing the following Groovy script:

   ```
   <install_dir>/obt/scripts/generateLaCleanupStmts.groovy
   ```

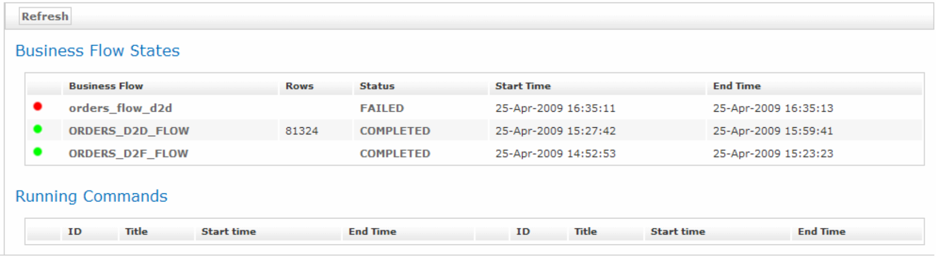
   The script generates delete statements with the appropriate row ranges specified.

   b Manually execute the delete statements against the history tables in the target database.

2 Start the Web Console, if it is not already started.

See also Starting Web Console (page 35)

3. Click **Monitoring** from the menu at the top of the page.
4. Select a business flow in Failed, Suspended, or Not Responding status.



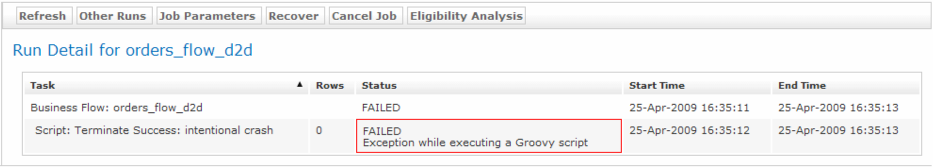
Business Flow States

	Business Flow	Rows	Status	Start Time	End Time
●	orders_flow_d2d		FAILED	25-Apr-2009 16:35:11	25-Apr-2009 16:35:13
●	ORDERS_D2D_FLOW	81324	COMPLETED	25-Apr-2009 15:27:42	25-Apr-2009 15:59:41
●	ORDERS_D2F_FLOW		COMPLETED	25-Apr-2009 14:52:53	25-Apr-2009 15:23:23

Running Commands

ID	Title	Start time	End Time	ID	Title	Start time	End Time

The Run Details page displays.



Run Detail for orders_flow_d2d

Task ▲	Rows	Status	Start Time	End Time
Business Flow: orders_flow_d2d		FAILED	25-Apr-2009 16:35:11	25-Apr-2009 16:35:13
Script: Terminate Success: intentional crash	0	FAILED Exception while executing a Groovy script	25-Apr-2009 16:35:12	25-Apr-2009 16:35:13

5. Click **Cancel Job**.
6. Click **Monitoring** to view the status of the job. The job status will change to CANCELLED.

Accessing the extracted data

After you copy your test data to a file, you still need to access it. From the Web Console, you can perform a variety of operations to facilitate access to the copied data:

- For database to file, you may relocate, copy, or delete the copied data from the Web Console.

This section includes

- Relocating extracted data (page 50)
- Accessing the structured data file (page 52)

Relocating extracted data

After you copy from database to file, you can relocate or copy the data to a variety of locations:

- another database
- another file location

To relocate data extracted to file from the Web Console:

1. Start the Web Console, if it is not already started.

See also Starting Web Console (page 35)

2 Click Launch from the menu at the top of the Web Console.

3 If you have extracted many times to file and your files are quite extensive, you probably need to select **Query** from the left navigation pane to query the indexes to find the exact data you want to relocate. Otherwise, you can select **Relocate** to see all of the available XML or CSV files.

TIP If you choose **Relocate**, you can use the date range search on the left navigation pane to narrow your choices. The date range search uses the format, `mm/dd/yy hh:mm:ss AM/PM`.

To perform a query:

a The Database to File Queries page displays the available indexes from each installed cartridge.

b Select the index you want to query. The query screen for the selected index displays.

c Enter values on which to query or leave the parameter fields blank to query on all values.

d Click **Run Query**. The cartridge runs that match the query display at the bottom of the page.

NOTE If you have not run a database to file extraction business flow at least once before querying, your extraction is empty and no rows are returned by the query.

4 Select the cartridge run that you wish to relocate and choose one of the following options:

NOTE The available options may vary slightly depending upon whether you chose **Relocate** or **Query** in step 3 (page 51).

- **Refresh** to refresh the list.
- **Lineage Details** displays more detailed history of the file.
- **Upload** uploads the data to a different database, checks for missing schema objects, and creates any schema objects that are missing. Refer to Schema mapping for upload (page 52) if you want to map the extracted data into a schema name that is different from the source.

TIP If you are planning to upload from an extraction file generated against Oracle to SQL Server, you should explicitly set the precision and scale on numeric columns, where appropriate, in the source database. Otherwise, if SQL Server has more restrictive precision and scale settings than Oracle, the upload may fail.

- **Copy** copies the data to an additional location.
- **Move** moves the data to a new location and delete it from the current location.

- **Delete from Backend** deletes the data from a backend.

5 If prompted, select the location where you want to move or copy the data.

> TIP If you do not see the location to which you want to relocate the data, you can create additional locations for the environment.

6 Click **Confirm** or **Cancel**.

Schema mapping for upload

When uploading from a structured data file to a database, you may want to perform some level of schema mapping. Doing so enables you to upload into a different schema name than the one from which you copied the data. For example, if you copied from a SQL Server database and are uploading to an Oracle database, you may prefer to use a different schema name.

To create and use a schema mapping for your database to file cartridges:

1 Create a schema mapping file in a location that is accessible to your Test Data Management instance. The contents of the mapping file should adhere to this format:

```
<src_db>.<src_schema>=<target_db>.<target_schema>
```

or, if the databases are heterogeneous, you might use this format instead:

```
<src_db>.<src_schema>=<target_schema>
```

2 In the Web Console, click **Parameters** from the menu at the top of the page.

3 In the left navigation area, click **Cartridges**.

4 Scroll down until you find the parameter named **Schema Mapping File for Upload**.

5 Enter the path and name of the schema mapping file that you created in step 1.

Accessing the structured data file

To access contents of your structured data files, you use the query server. Refer to Chapter 8, Using the query server to access database to file output.

5 Running business flows and jobs from the command line

To provide additional flexibility, HP Test Data Management enables you to run business flows and jobs from the command line as well as from the Web Console. This capability is useful for environments where you cannot easily access the Web Console or need to script the running of business flows or jobs.

In this chapter

Before you begin

Before you run business flows, you should perform some pre-flight checks, for example, confirming parameter settings. Beyond that, certain types of business flows require that you perform preparatory tasks before running them.

- The preparatory tasks for running business flows from the Web Console also apply to running on the command line. Refer to Before you begin (page 45).
- In order to run business flows or jobs from the command line, you may need some or all of the following information, which can be obtained from reviewing the deployed business flows in the Web Console:
  - environment name
  - environment ID
  - business flow name
  - job parameters
  - job name
  - job run ID
  - destination location

Running business flows

To run business flows or jobs from the command line:

1. Ensure that you have performed the relevant tasks from Before you begin (page 53).
2. Open a command window.
3. Navigate to the following directory:

   ```
   <install_directory>/obt/bin
   ```

   where <install_directory> is the location where you installed the software.
4. Run the business flow using the appropriate syntax:

For	Syntax
UNIX	./launch_businessflow.sh [-r] -e <environment_name> -j <business_flow_name> <parameters>
DOS	launch_businessflow.bat [-r] -e <environment_name> -j <business_flow_name> “<parameters>”

Where	Is
-r	The recover flag, -r, allows you to recover jobs and business flows. See Recovering business flows (page 55). Although the recover flag, -r, is optional, HP recommends that you use it whenever you run a job or business flow.
environment_name	The name of the environment. This is optional if you only have one environment.
business_flow_name	The name of the business flow.
parameters	The parameters as a space-separated list with each entry in the form <name>=<value>. For Windows, each individual entry must be enclosed by quotation marks. The parameter values for all cartridges included in the business flow are required. For a date or time policy parameter, use the following format: [YY]YY.MM.DD[HH24:MI:[SS[.SSS]]]

TIP To verify the usage of the job from the command line, enter `launch_businessflow.bat -h`.
The usage is displayed.
To verify the parameters for the job, enter `launch_businessflow.bat -j <business_flow_name> -e <environment_name> -h`.
After you enter the encryption key, the parameters are displayed.

DOS example

```
launch_businessflow.bat -e Oracle_env -j orders_flow_d2d
-r "Min_Months_to_Retain=24"
```

TIP If the parameter value is left blank, and a default value was defined in Designer, then the default value is used. If there is no default value, then NULL is used.

NOTE Running an undo business flow requires the Run ID for the business flow you want to undo. The Run ID for the business flow is displayed on the Run Summary page of the

5 Enter the encryption key at the prompt. The encryption key is case sensitive.

The job runs, and displays a success or failure message.

6 Optionally, start the Web Console to monitor the progress of the job.

See also Chapter 6, Monitoring jobs

Recovering and cancelling business flows and jobs

See also Recovering or cancelling business flows and jobs (page 48)

In this section

- Recovering business flows (page 55)
- Recovering jobs using the recovery script (page 56)
- Cancelling jobs from the command line (page 57)

Recovering business flows

See also Recovering business flows and jobs (page 49)

Using the recover flag, -r, allows you to rerun a business flow using the same parameters as the first time it was run. You can also use the -r flag to continue the business flow after a scheduled pause activity.

NOTE To recover a job, the job parameters must be exactly the same as the last time you ran it. If there are any differences in the syntax, even an extra space, then a new job is launched instead of the previous job being recovered.

1 Ensure that you have performed the relevant tasks from Before you begin (page 53).

2 Open a command window.

3 Navigate to the following directory:

```
<install_directory>/obt/bin
```

where <install_directory> is the location where you installed the software.

4 Enter the same job name and parameters as the job you want to recover.

DOS example

If you ran the business flow example given in Running business flows (page 54), you would run:

```
launch_businessflow.bat -e Oracle_env -j orders_flow_d2d
-r "Min_Months_to_Retain=24"
```

5 Optionally, start the Web Console to monitor the progress of the job.

See also

Chapter 6, Monitoring jobs

Recovering jobs using the recovery script

Using the recovery script to recover jobs enables you to select the specific job you want to recover.

To recover a job using the recovery script:

1 Ensure that you have performed the relevant tasks from Before you begin (page 53).

2 Locate the Group ID of the failed job you want to recover.

The Group ID is displayed on the Web Console History page.

See also

Viewing job history (page 63)

3 Open a command window.

4 Navigate to the following directory:

```
<install_directory>/obt/bin
```

where <install_directory> is the location where you installed the software.

5 Run the appropriate recovery script for the job you want to recover:

Job	For	Syntax
any business flow or database to file job	UNIX	./recover_job.sh -e <environment_name> -g <group_ID>
	DOS	recover_job.bat -e <environment_name> -g <group_ID>

Where	Is
environment_name	The name of the environment. This is optional if you only have one environment.
group_ID	The Group ID of the failed job as it appears in the Web Console.

DOS example

```
recover_job.bat -e "Oracle_env" -g "3"
```

6 Enter the encryption key at the prompt. The encryption key is case sensitive.

The job runs, and displays a success or failure message.

7 Optionally, start the Web Console to monitor the progress of the job.

See also Chapter 6, Monitoring jobs

Cancelling jobs from the command line

See also Cancelling business flows and jobs (page 49)

Cancelling jobs from the command line requires the `cancelJob API` Groovy script. The script allows you to update the status of a job to Cancelled.

1 Locate the Group ID of the business flow you want to cancel.

The Group ID is displayed on the Web Console History page.

See also Viewing job history (page 63)

2 Navigate to the following directory:

```
<install_directory>/obt/bin
```

where <install_directory> is the location where you installed the software.

3 Run the `cancelJob` API using the appropriate syntax:

For	Syntax
UNIX	./launch_groovyscript.sh -e MyEnvironment -f ../scripts/ cancelJob.groovy <groupRunId>
DOS	launch_groovyscript.bat -e MyEnvironment -f ..\scripts\cancelJob.groovy <groupRunId>

where <groupRunID> is the Group ID as defined on the Web Console.

4 Optionally, start the Web Console to monitor the progress of the job.

See also Chapter 6, Monitoring jobs

Accessing the copied data

See also

Accessing the extracted data (page 50)

This section includes

- Relocating database to file extracted data (page 57)
- Accessing the structured data file (page 59)

Relocating database to file extracted data

See also Relocating extracted data (page 50)

To relocate database to file data from the command line:

1 Ensure that you have performed the relevant tasks from Before you begin (page 53).

2. Open a command window.
3. Navigate to the following directory:

   ```
   <install_directory>/obt/bin
   ```

   where <install_directory> is the location where you installed the software.
4. Run the job from the command line using the following syntax:

For	Syntax
UNIX	./launch_ea_job.sh [-r] -e <environment_name> -j <jobName> sourceJobRunID=<runID> [destLocation=<dest_loc>]
DOS	launch_ea_job.bat [-r] -e <environment_name> -j <jobName> sourceJobRunID=<runID> [destLocation=<dest_loc>]

Where	Is	Description
-r	The recover flag, -r, allows you to recover jobs and business flows.	
environment_name	The name of the environment. This is optional if you only have one environment.	
jobName	One of the following:	
	XML_DATA@COPY_BE_TO_DB	Copies data from a backend to a database.
	XML_DATA@COPY_BE_TO_BE	Copies data from a backend to a backend.
	XML_DATA@UPLOAD_BE_TO_DB	Uploads data from a backend to a different database than the original source, checks for missing schema objects, and creates any schema objects that are missing.
	XML_DATA@MOVE_BE_TO_DB	Copies data from a backend to a database, and deletes it from the backend.
	XML_DATA@MOVE_BE_TO_BE	Copies data from one backend to another backend, and deletes it from the original.
	XML_DATA@DELETE_FROM_BE	Deletes the data from the backend.
	XML_DATA@DELETE_FROM_DB	Deletes the data from the database.
runID	The run ID of the job that placed the extraction at the current location.	
dest_loc	The name of the destination location as defined in the Web Console. Required for COPY_BE_TO_BE, MOVE_BE_TO_BE, and UPLOAD_BE_TO_DB only.	

DOS example

```
launch_ea_job.bat -e "MyEnvironment" -j
"XML_DATA@COPY_BE_TO_BE" -r "sourceJobRunId=3"
"destLocation=LOCAL_DATA_FS"
```

5. Enter the encryption key at the prompt. The encryption key is case sensitive.

   The job runs, and displays a success or failure message.

6. Optionally, start the Web Console to monitor the progress of the job.

See also Chapter 6, Monitoring jobs

Accessing the structured data file

To access contents of your structured data files, you use the query server. Refer to Chapter 8, Using the query server to access database to file output.

6 Monitoring jobs

In order to effectively manage your system, you need to be able to monitor activity in a variety of ways. All of the following information can be helpful to you in managing your system:

- status of currently running jobs
- historical information about previously run jobs
- eligibility analytics
- log files

In this chapter

Monitoring current and past jobs

The Web Console provides you with a number of facilities to monitor the status and history of your jobs.

In this section

Viewing a status overview

The Monitoring page in the Web Console provides you with an overview of the currently running and previously run jobs.

To view the Monitoring page:

1. Start the Web Console, if it is not already started.

   See also Starting Web Console (page 35)

2. Click **Monitoring** from the menu at the top of the page.

   The overview page displays the status of your business flows as well as any currently running commands.

In the Business Flow States section, the following information is displayed:

- Name of the business flow
- Number of rows affected
- Status of the business flow
- Start and end times of the business flow

NOTE Only the most recent run of each installed business flow is displayed.

In the Running Commands section, the following information is displayed about currently running jobs:

- Job ID number
- Name of the running job
- Start and end times of the running job

NOTE After a job has finished running, it no longer appears in the Running Commands section. Click **Commands** from the left navigation pane to view all commands that have run.

Viewing detailed status

From the Monitoring page, you can drill down to more detailed status information:

- From the Business Flow States section, you can drill down to the Run Details page, which displays a list of the individual tasks performed by the business flow and their status. If a business flow did not complete for some reason, the Run Details page can show you which task in the business flow was at fault.
- From the Running Commands section, you can drill down to the running log of the command.

Viewing run details

To access the Run Details page:

1 Navigate to the Monitoring page as per the instructions in Viewing a status overview (page 61).

2 Click a highlighted business flow name or status to navigate to the Run Details page.

3 From the Run Details page, you can perform the following operations:

   a Click **Command** to view the log file for the current run of the business flow.

   b Click **Other Runs** to view all runs of the business flow.

   c Click **Job Parameters** to view the runtime parameters and their values for the current run of the business flow.

d. Click **Eligibility Analytics** to view the eligibility analytics for the current run of the business flow. Refer to Viewing eligibility analytics (page 64) for more information.

e. Click **Row Counts** to view the detailed row count information for the current run of the business flow.

Viewing the log of a running command:

To view the logging of a running command:

1. Navigate to the Monitoring page as per the instructions in Viewing a status overview (page 61).
2. Click the highlighted job title to see information about the currently running job.

Viewing job history

The History page displays all business flows and jobs that have ever been run in the environment you are currently viewing.

1. Navigate to the Monitoring page as per the instructions in Viewing a status overview (page 61).
2. Click **History** in the left navigation pane.
3. Optionally filter the data by any of the following criteria:
   - group run id
   - run id
   - the date the job started (in yyyy-mm-dd hh:mm:ss or yyyy-mm-dd hh:mm:ss AM/PM format)
   - the date the job finished (in yyyy-mm-dd hh:mm:ss or yyyy-mm-dd hh:mm:ss AM/PM format)
   - the name of the job
   - the batch name of the job

   Each filter you add is displayed at the top of the page.
4. Click the red X to clear individual filters.
5. Click **CSV** or **XML** to export the filtered data to a CSV or XML file.

Viewing eligibility analytics

Eligibility analytics tell you which rows are selected for and excluded from movement. To see eligibility analytics, you must do all of the following:

- Paused your business flow after the selection task but before the movement task. Eligibility analytics are only available immediately after data selection. Refer to *HP Test Data Management Developer's guide* for information about how to insert a pause in your business flow.
- Enabled eligibility analytics on your rules in the model in Designer. Refer to *HP Test Data Management Developer's guide* for information about how to enable analytics on rules.
- Turned on the eligibility analytics parameter in the Web Console.

To view eligibility analytics:

1. Navigate to the Run Details page as per the instructions in Viewing detailed status (page 62).
2. Click **Eligibility Analytics**. The Eligibility Analytics page displays.
3. Select a cartridge. The eligibility analytics for that cartridge display.
4. Select a table from the left navigation page to view the rows for that table.
5. To filter the data, perform any of the following operations:
    - Select **Exclusion Only**, **Selection Only**, or **All Data** from the left navigation pane.
    - Select a table name to view the eligibility analytics for that table.
6. To export the filtered data to a CSV or XML file, click **CSV** or **XML**.

Viewing commands

In some cases, you may want to review the exact command that was used for a particular run of a business flow. For example, you may want this information for debugging purposes.

To view all commands that are running or have completed:

1. Navigate to the Monitoring page as per the instructions in Viewing a status overview (page 61).
2. Click **Commands** from the left navigation pane. The Commands page displays the commands.
3. To filter the list:
    - Click **Refresh** to refresh the list of commands.
    - Click **Show All Users** to show the jobs that belong to all users.
    - Click **Show Cleared** to show all cleared jobs.
    - Click **Clear Completed** to remove all completed jobs from the list.

4 Click a highlighted command to view details about the command and the log file.

Using log files

When reviewing job status or diagnosing an issue, it can sometimes be helpful to view the complete information provided in Test Data Management's log files. If you are regularly consulting the log files for a particular purpose, you may also want to adjust the logging properties to include or exclude certain information.

In this section

- Viewing log files (page 65)
- Editing the obt.log file logging properties (page 66)

Viewing log files

Error messages from the Web Console and Test Data Management are captured and appended to log files. By default, the log files are saved to the <install_directory>/obt/log/ directory and the logging level is set to INFO.

1 Navigate to the directory containing the log file.

Example

```
cd <install_directory>/obt/log/
```

where <install_directory> is the location where you installed the software.

2 Open one of the following log files using a text editor of your choice:

File name	Description
obt.log	The obt.log file captures all logging information for HP Test Data Management. New lines appended to the log file are displayed as they appear. Multiple log files are numbered in sequence, for example, obt.log1.
pdm_server_<runID>.log where <runID> is the run ID of the job as displayed on the Web Console.	The pdm_server_<runID>.log file is used for advanced data selection and partitioned data movement. The log file contains the generated statements, execution plans, and statistics from the execution of the advanced data selection and partitioned data movement jobs.

Editing the obt.log file logging properties

You can edit the log4j.properties file to change the following logging properties for the obt.log file:

- where the log files are kept.
- what information is logged.
- the maximum size of the log file.
- how much logging information is kept.

By default, the obt.log file is limited to 10 MB. When it exceeds the default limit, the obt.log file is renamed to a backup file, and a new obt.log file is created. By default, a total of nine backup files are kept, limiting the log files to a 100 MB maximum size.

1. Navigate to the directory that contains the log4j.properties file.

   Example

   ```
   cd <install_directory>/obt/config/
   ```

   where <install_directory> is the location where you installed the software.

2. Create a backup copy of log4j.properties that you can roll back to in case of any problems.
3. Open the log4j.properties file with a text editor of your choice.
4. Make your desired changes to the file.
5. Save the log4j.properties file. The changes are applied automatically.

7 Managing the repository and encryption keys

In certain situations, you may need to perform tasks such as locking the repository or changing its password. You might also need to change your environment encryption key for security purposes.

In this chapter

Locking and unlocking the repository

In some situations, you may need to lock the repository. When you lock the repository, you can:

- prevent business flows and cartridges from being deployed in all environments.
- prevent business flows and cartridges from being deployed in a particular environment.
- prevent any new environments from being created.
- prevent any existing environments from being deleted.

For example, you may want to stop all deployments while you perform some system maintenance or upgrade activities. Once you are ready, you can then unlock the repository to allow business flows and cartridges to again be deployed.

NOTE All new environments are unlocked by default.

In this section

Locking the repository

To lock the repository:

NOTE If a business flow or cartridge deployment is already in process, locking the repository has no effect on that deployment.

1 Navigate to the <install_directory>/obt/bin directory.

where <install_directory> is the location where you installed the software.

2 Run the lock_repository script using the appropriate syntax:

For	Syntax
UNIX	./lock_repository.sh [-m <lock_mode>] [-e <environment_name>] -u <yourName> [-c <comment>]
DOS	lock_repository.bat [-m <lock_mode>] [-e <environment_name>] -u <yourName> [-c <comment>]

Where	Is
lock_mode	Optional. If you want to specify a lock mode, enter NO_CREATE_REPOS. This lock mode prevents any environments from being created or deleted.
environment_name	Optional. The name of the environment.
yourName	The name of the person locking the repository. Used for auditing purposes only. If name contains spaces, you need to use double-quotes around the name.
comment	Optional. The reason why the repository is being locked.

3 Enter the encryption key at the prompt.

Windows examples

- To prevent business flows and cartridges from being deployed in all environments:

```
lock_repository.bat -u Joe
```

- To prevent business flows and cartridges from being deployed in the environment Oracle_Env:

```
lock_repository.bat -e Oracle_Env -u "Joe Smith"
```

- To prevent environments from being created or deleted:

```
lock_repository.bat -m NO_CREATE_ENV -u Paul -c "No new
environment creation"
```

Unlocking the repository

To unlock the repository:

1 Navigate to the <install_directory>/obt/bin directory.

where <install_directory> is the location where you installed the software.

2 Run the unlock_repository script using the appropriate syntax:

For	Syntax
UNIX	./unlock_repository.sh [-m <lock_mode>] [-e <environment_name>] -u <yourName> [-c <comment>]
DOS	unlock_repository.bat [-m <lock_mode>] [-e <environment_name>] -u <yourName> [-c <comment>]

Where	Is
lock_mode	Optional. If you want to specify a lock mode, enter NO_CREATE_REPOS. This lock mode enables environment creation and deletion in the repository.
environment_name	Optional. The name of the environment.
yourName	The name of the person unlocking the repository. Used for auditing purposes only. If name contains spaces, you need to use double-quotes around the name.
comment	Optional. The reason why the repository is being unlocked.

The script prompts for the encryption key.

3 Enter the encryption key at the prompt.

Windows examples

- To unlock all environments:

```
unlock_repository.bat -u Paul -c "Unlock to allow
check-ins"
```

- To unlock just the environment Oracle_Env, without affecting any other locked environments:

```
unlock_repository.bat -e Oracle_Env -u Mike -c "Unlock
Oracle_Env"
```

- To allow environments to be created or deleted:

```
unlock_repository.bat -m NO_CREATE_ENV -u Paul -c "Allow
environment creation"
```

Changing encryption keys and the repository password

The password manager utility is used to change the encryption key and repository password.

In this section

- Using password manager to change the encryption key (page 70)
- Using password manager to change the repository password (page 71)

Using password manager to change the encryption key

Use password manager to change the encryption key for the environment.

1 Open a command window.

2 Run the utility using the following syntax:

Operating system	Syntax
UNIX	<install_directory>/obt/bin/runPwManager.sh -m
DOS	<install_directory>/obt/bin/runPwManager.bat -m

where <install_directory> is the location where you installed the software.

3 Enter values for the prompts. Enter ? to see a list of available values.

— Enter environment to modify:

— Enter CURRENT encryption key:

— Re-enter CURRENT encryption key:

— Enter NEW encryption key:

— Re-enter NEW encryption key:

All encrypted passwords in the connection-sources.xml file are re-encrypted using the new key. The encryption key values are not displayed on the window.

Using password manager to change the repository password

If you need to change the password for your repository, use the password manager utility to update the password in the connection_sources.xml file. HP recommends using the Web Console to modify location passwords for all other databases.

NOTE You can use the password manager utility when required for automation purposes.

To change the password for the repository owner on the database:

1. If you are using the embedded repository, perform the following steps:
   a. Ensure the embedded repository has been shut down.
   b. Delete the following file:

      ```
      <install_directory>/obt/dbrep/access.cfg
      ```

      where <install_directory> is the location where you installed the software.
   c. Navigate to the following directory:

      ```
      <install_directory>/obt/install
      ```

      where <install_directory> is the location where you installed the software.
   d. Run the following script to reset the password:

      ```
      <install_dir>\obt\install\obt_deployer.bat -buildfile
      <install_dir>\foundation\components\install\deploy\
      build_repository.xml create.obtrep.embedded.user
       "-Drepository.rdbms.is.derbydb=true"
       "-Drepository.owner.name=obt_rep"
       "-Drepository.owner.password=<obt_rep_password>"
      ```

      where <install_dir> is the location where you installed the software, and <obt_rep_password> is the new password for the embedded repository.
   e. Restart the embedded repository.
2. Ensure the encryption key has not been changed.

   The master encryption key is required for verification purposes. You can change the encryption key after completing the steps in this section.
3. Open a command window.
4. Run the password manager utility using the following syntax:

Operating system	**Syntax**
UNIX	<install_directory>/obt/bin/runPwManager.sh -p
DOS	<install_directory>/obt/bin/runPwManager.bat -p

where <install_directory> is the location where you installed the software.

5 Enter values for the prompts. Enter a question mark (?) to see a list of available values.

- Enter environment to modify:
- Enter master encryption key:
- Re-enter master encryption key:
- Enter location name to modify:
- Enter NEW password:
- Re-enter NEW password:

The password is changed, and the values are saved to the following file:

```
<install_directory>/obt/config/connection-sources.xml
```

where <install_directory> is the location where you installed the software.

8 Using the query server to access database to file output

After your data has been copied to XML or CSV files, you can use the query server to upload the data to a test database and populate spreadsheets.

The query server uses standard JDBC and ODBC clients and servers to connect to your test data. The JDBC driver is installed during product installation, and requires no configuration. The ODBC driver and the Windows OLEDB driver are installed separately, and must be configured before use.

The collections of test data you created are available from a data source. The XML data source is created by default when you install the query server.

Depending on your needs, you can access all of your test data collections on the same data source, or create new data sources.

In this chapter

- Installing the query server (page 73)
- Starting the query server (page 74)
- Installing and configuring the ODBC or OLE DB driver (page 76)
- Installing and configuring the OLE DB database driver on Windows (page 78)
- Connecting to a test data source (page 80)
- Creating file collections (page 85)
- Viewing your collection (page 87)
- Uninstalling the query server (page 105)

Installing the query server

Installing the query server also installs the query server agent and the default XML data source. To use a different data source, or to create additional data sources, see Using multiple data sources (page 107).

By default, the query server is configured to run on port 19985, and the query server agent is configured to run on port 19988. If you want to change the ports, you must use the command line to install the query server.

In this section

- Installing the query server using the command line (page 74)
- Installing the query server on Windows (page 74)

Installing the query server using the command line

Use the command line to install the query server on UNIX or Windows.

1 Navigate to the directory that contains the setup file, oasetup.sh.

Example

```
cd <install_directory>/obt/bin
```

where <install_directory> is the location where the software was installed.

2 Run the setup file using the appropriate syntax:

For	Use
UNIX	`./oasetup.sh [server_port agent_port]`
DOS	`oasetup.bat [server_port agent_port]`

Where	Is
server_port	Optional. The port number you want to use for the query server. By default, the port number is 19985.
agent_port	Optional. The port number you want to use for the query server agent. By default, the port number is 19988.

The query server and the xmlData data source are installed.

Installing the query server on Windows

1 Navigate to the directory that contains the setup file, oasetup.bat.

Example

```
cd <install_directory>/obt/bin
```

where <install_directory> is the location where the software was installed.

2 Double-click the setup batch file, oasetup.bat, to install the query server.

The query server and the xmlData data source are installed.

Starting the query server

Clients communicate with the query server using the included JDBC database driver, or the ODBC or OLE DB drivers you install.

In this section

- Starting the query server on Windows (page 75)
- Starting the query server on UNIX (page 75)

Starting the query server on Windows

You can run the query server as a Windows service or as a program. The user that starts the server needs to be part of the Administrator group.

NOTE If you are using mapped network drives to store your collection information, you must run the query server as a program.

In this section

- Running the query server as a Windows service (page 75)
- Running the query server as a Windows program (page 75)

Running the query server as a Windows service

By default, the query server runs as a Windows service.

1. Open the Services window, for example, **Start > Administrative Tools > Services**.
2. Ensure the following services are started:
   - QueryServer
   - QueryServer_Agent

Running the query server as a Windows program

To run the query server as a Windows program, you must stop the query server Windows service before starting the program.

1. Open the Services window, for example, **Start > Administrative Tools > Services**.
2. Ensure the following services are stopped:
   - QueryServer
   - QueryServer_Agent

   TIP If you plan to continue using the Windows program, you can bypass this step by setting the Startup Type for both services to Manual.

3. Open a command window, and navigate to the directory that contains the query server.

Example

```
cd <install_directory>/obt/sqlxml/server/bin
```

where <install_directory> is the location where the software was installed.

4. Run the following commands:

```
oastrtr -n HPQueryServer -d ..\cfg\oadm.ini
```

Starting the query server on UNIX

1. Navigate to the directory that contains the query server.

Example

```
cd <install_directory>/obt/sqlxml/server/bin
```

where <install_directory> is the location where the software was installed.

2 Start the query server.

Syntax	Description
`./oaserver.sh -start`	starts the server.
`./oaserver.sh -stop`	stops the server.
`./oaserver.sh -status`	displays the status of the server.

Installing and configuring the ODBC or OLE DB driver

If you want to use an ODBC client on Windows or UNIX, or an OLE DB client on Windows, you need to install the appropriate driver.

In this section

- Installing and configuring the ODBC database driver on UNIX (page 76)
- Installing and configuring the ODBC database driver on Windows (page 76)
- Installing and configuring the OLE DB database driver on Windows (page 78)

Installing and configuring the ODBC database driver on UNIX

After you install the ODBC database driver on UNIX, you need to configure the odbc.ini file.

1 Navigate to the directory that contains the driver installation file, oadriver.sh.

Example

```
cd <install_directory>/obt/bin
```

where <install_directory> is the location where the software was installed.

2 Start the installation file.

Example

```
./oadriver.sh
```

3 Select Option 1 to install the driver.

The ODBC driver is installed.

Installing and configuring the ODBC database driver on Windows

After you install the ODBC database driver on Windows, you need to configure the ODBC database driver. You can configure the ODBC database driver before or after you start the query server process, but you can only test the connection if the query server process is running.

1 Navigate to the directory that contains the driver installation file, oadriver.bat.

Example

```
cd <install_directory>/obt/bin
```

where <install_directory> is the location where the software was installed.

2 Double-click the installation batch file, oadriver.bat.

3 Install the ODBC driver.

  a Select Option 1 to install the driver.

  b Select Option 1 to install the ODBC driver.

  The ODBC driver is installed.

> NOTE After you install the ODBC driver, you can repeat the process and select 2 to install the OLE DB driver.

4 Open the ODBC Data Source Administrator:

  a From the Windows Start menu, choose **Control Panel**.

  b From the Control Panel, choose **Administrative Tools** > **Data Sources (ODBC)**.

  The ODBC Data Source Administrator displays.

5 Using the ODBC Data Source Administrator, create a new data source.

  a Click the **System DSN** tab.

  b Click **Add**.

  The Create New Data Source window opens.

  c Select **DataDirect OpenAccess SDK 6.0** from the list of values.

  d Click **Finish**.

  The Data Direct OpenAccess SDK ODBC Driver Setup window opens.

  e Enter the following values:

Field	Description
Data Source Name	Enter the name for the ODBC data source. The name must match the name of the Service Data Source.
OpenAccess Service Host	Enter the name of the host machine or IP address on which the data source is running. If you are running on your local machine, enter localhost.
OpenAccess Service Port	Enter 19988.
OpenAccess Service Data Source	Enter the name of the data source. The default data source name is xmlData. If you have added a new data source, enter the name of the new data source.

6 If you have started the query server process, click **Test Connect** to test the connection.

See also Starting the query server (page 74)

The Login to OpenAccess Data Source window opens.

  a Enter the following information:

Field	Expected value
Data Source User Name	install
Data Source Password	OA
Custom Properties	leave blank

b Click **OK** to test the connection.

c Click **OK** to close the confirmation window.

7 Click **OK** to save the configuration.

Installing and configuring the OLE DB database driver on Windows

NOTE The OLE DB driver is supported on Windows 32-bit.

After you install the OLE DB database driver on Windows, you need to configure the database driver. You can configure the OLE DB database driver before or after you start the query server process, but you can only test the connection if the query server process is running.

1 Navigate to the directory that contains the driver installation file, oadriver.bat.

Example

```
cd <install_directory>/obt/bin
```

where <install_directory> is the location where the software was installed.

2 Double-click the installation batch file, oadriver.bat.

3 Install the ADO OLE DB driver.

a Select Option 1 to install the driver.

b Select Option 2 to install the ADO OLE DB driver.

The OLE DB driver is installed.

4 Start the DataDirect Configuration Manager from the DataDirect OpenAccess SDK 6.0 program group. For example, **All Programs** > **DataDirect OpenAccess SDK 6.0** > **Client for ADO** > **Configuration Manager**.

The Configuration Manager displays.

5 Configure the data source.

a Select **File > New > Data Source** from the menu bar. The New Data Source window opens.

b Enter a unique name for the data source.

c Select **DataDirect OpenAccess SDK for ADO 6.0 Provider** from the list of values.

d Click **Set Up Data Source**. The DataDirect OpenAccess SDK for ADO Provider Setup window opens.

e Enter the following values:

Field	Description
Description	Optionally enter a description for the data source name.
OpenAccess Service Host	Enter the name of the host machine or IP address on which the data source is running. If you are running on your local machine, enter localhost.
OpenAccess Service Port	Enter 19988, or the port number on which you started QueryServer.
OpenAccess Service Data Source	Enter the name of the data source. The default data source name is xmlData. If you have added a new data source, enter the name of the new data source.

6 If you have started the query server process, click **Test Connect** to test the connection.

See also Starting the query server (page 74)

The Login to OpenAccess Data Source window opens.

a Enter the following information:

Field	Expected value
Data Source User Name	install
Data Source Password	OA

b Click **OK** to test the connection.

c Click **OK** to close the confirmation window.

7 Click **OK** to save the configuration.

Connecting to a test data source

You can connect to the test data source using ODBC or JDBC clients. The JDBC database driver is included in the query server installation, and the JDBC driver does not need to be configured. The connection string syntax includes all of the connection information required to connect to any data source.

If you want to use an ODBC or OLE DB client, you need to install and configure the driver first.

See also Installing and configuring the ODBC or OLE DB driver (page 76)

By default, the data source is called xmlData. If you create a different data source, substitute the correct name.

This section assumes that the query server is installed and running.

In this section

- Connecting using the Interactive SQL (JDBC) client (page 80)
- Connecting using the Interactive JDBC client (ij) (page 81)
- Connecting using a user-supplied JDBC client (page 82)
- Connecting using the Interactive SQL (ODBC) client (page 83)
- Connecting using the Interactive SQL (OLE DB) client (page 84)

Connecting using the Interactive SQL (JDBC) client

1 Navigate to the directory that contains the oaisql program.

Example

```
<install_directory>/obt/bin/
```

where <install_directory> is the location where the software was installed.

2 Start the Interactive SQL(JDBC) program from the command line.

Platform	Use
UNIX	`oaisql.sh`
Windows	`oaisql.bat`

3 Connect to the server using the following syntax:

```
connect <username>*<password>@<host>:<port>;
ServerDataSource=<datasource_name>;
```

Where	Is
username	the name of the user with permissions to connect to the data source. The default value is install.
password	the password for the user. The default value is OA.

Where	Is
host	the name of the host machine or IP address on which the data source is running. If you are running on your local machine, enter localhost.
port	the port number for the host machine. The default value is 19988.
datasource_name	the name of the data source you are connecting to. The default value is xmlData.

Default xmlData example

```
connect install*OA@localhost:19988;
ServerDataSource=xmlData;
```

The install user account is created by default and must be used the first time you connect.

See also

- Creating file collections (page 85)
- Viewing your collection from the Interactive SQL or ij clients (page 88).

Connecting using the Interactive JDBC client (ij)

1 Start Interactive JDBC (ij).

Platform	Use
UNIX	./<install_directory>/obt/bin/dbarch_sql.sh
Windows	<install_directory>/obt/bin/dbarch_sql.bat

where <install_directory> is the location where the software was installed.

2 Connect to the driver using the following syntax:

```
driver 'com.ddtek.jdbc.openaccess.OpenAccessDriver';
```

3 Connect to the server using the following syntax:

```
connect 'jdbc:openaccess://
<host>:<port>;serverDataSource=<datasource_name>;user=<us
ername>;password=<password>';
```

Where	Is
host	the name of the host machine or IP address on which the data source is running. If you are running on your local machine, enter localhost.
port	the port number for the host machine. The default value is 19988.

Where	Is
datasource_name	the name of the data source you are connecting to. The default value is xmlData.
username	the name of the user with permissions to connect to the data source. The default value is install.
password	the password for the user. The default value is OA.

default xmlData example

```
connect 'jdbc:openaccess://
localhost:19988;serverDataSource=xmlData;user=install;passwo
rd=OA';
```

See also

- Creating file collections (page 85).
- Viewing your collection from the Interactive SQL or ij clients (page 88)

Connecting using a user-supplied JDBC client

To connect to the data source using a different JDBC client, you need to know the following information:

Connection information	Expected value
driver location	<install_directory>/obt/lib/oajc.jar where <install_directory> is the location where you installed the software.
driver name	com.ddtek.jdbc.openaccess.OpenAccessDriver
JDBC URL	jdbc:openaccess:<host>:<port>;serverDataSource=<datasource_name>;user=<username>;password=<password> Example: `jdbc:openaccess://localhost:19988;serverDataSource=xmlData;user=sample;password=samplePassword`

Where	Is
host	the name of the host machine or IP address on which the data source is running.
port	the port number for the host machine. The default value is 19988.
datasource_name	the name of the data source you are connecting to. The default value is xmlData.

Where	Is
username	the name of the user with permissions to connect to the data source. Optional.
password	the password for the user. Optional.

Connecting using the Interactive SQL (ODBC) client

This section assumes that the query server is installed and running, and that the ODBC driver has been installed and configured.

See also

- Installing the query server (page 73)
- Starting the query server (page 74)
- Installing and configuring the ODBC or OLE DB driver (page 76)

1 Navigate to the directory that contains the oaisql program.

Example

```
<install_directory>/obt/bin/
```

where <install_directory> is the location where the software was installed.

2 Start the Interactive SQL (ODBC) program or your preferred SQL client.

Platform	Use
UNIX	`oaisql.sh -odbc`
Windows	`oaisql.bat -odbc` Or from the Start menu, choose All Programs > DataDirect OpenAccess SDK 6.0 > Client for ODBC > Interactive SQL (ODBC)

NOTE When using Interactive SQL (ODBC), the create collection, drop collection, and drop schema statements require a prefix of 'exec' or '!'. This is not required for Interactive SQL (JDBC).

3 Connect to the xmlData data source as the install user.

Example

```
connect install*OA@<ODBC_DataSource_Name>
```

where <ODBC_DataSource_Name> is xmlData or the name you chose for the ODBC data source. For Unix ODBC, the default data source name is TDMQS.

The install user account is created by default and must be used the first time you connect.

NOTE The same connection string is used if you use the Interactive SQL (ODBC) program or your preferred SQL client.

See also

- Creating file collections (page 85)
- Viewing your collection from the Interactive SQL or ij clients (page 88).

Connecting using the Interactive SQL (OLE DB) client

This section assumes that the query server is installed and running, and that the OLE DB driver has been installed and configured.

See also

- Installing the query server (page 73)
- Starting the query server (page 74)
- Installing and configuring the ODBC or OLE DB driver (page 76)

1 Navigate to the directory that contains the oaisql program.

Example

```
<install_directory>/obt/bin/
```

where <install_directory> is the location where the software was installed.

2 Start the Interactive SQL (OLE DB) program or your preferred SQL client.

Platform	Use
DOS	`oaisql.bat -oledb`
Windows	From the Start menu, choose **All Programs > DataDirect OpenAccess SDK 6.0 > Client for ADO > Interactive SQL**

3 Enter **connect** at the prompt.The Data Link Properties window displays.

4 Select DataDirect OpenAccess SDK for ADO 6.0 Provider from the list of values.

5 Click **Next**. The Connection tab displays.

6 Test the connection using the following information:

Field	Expected value
Data Source	The name you supplied in the configuration manager.
User name	install
Password	OA

7 Click **Test Connection**.

8 Click **OK** to close the confirmation window.

9 Click **OK** to save the configuration and return to the ISQL prompt.

You can now type your queries at the prompt.

See also

- Creating file collections (page 85)
- Viewing your collection from the Interactive SQL or ij clients (page 88).

Creating file collections

A data collection is a file system location and pattern containing the files created when you copied your data using database to file copying. Data contained in CSV, XML and XSD files compressed using GZIP can also be read. Creating the collection enables the SQL access system to view the files. A collection is defined by a specific file system directory and a wild-card filename pattern.

Collection configuration information is stored in the OASYSTEM.oa_collections table.

You can create collections where the schema is based on the cartridge name, or create your own schema name. The collections can be located on a local file system. You can also create a name for the collection, or let the server create one for you automatically.

In this section

- Creating collections (page 85)
- Specifying the path for a local file system (page 86)
- Dropping collections, schemas, and connections (page 86)

Creating collections

You can create collections on a local file system, or on external file systems. Both local and external file systems use the same syntax to create the collection, but the local file systems require a path, and external file systems require a connection.

See also

- Specifying the path for a local file system (page 86)

1 Locate the folder or connection that contains the files created from running database to file.

NOTE If you plan to connect to an external file system, you must create the connection before you create the collection.

2 Ensure the server process is running.

See also

Starting the query server (page 74)

3 Start your preferred SQL client and connect to the xmlData data source.

4 To create a collection where the schema name is the same as the cartridge name, create the collection using a statement similar to the following:

```
create collection <collection_name> using pattern '<path
| connection>\*.xm*';
```

where <collection_name> is the name you choose for your collection, and <path | connection> is the location of the XML and XSD files from step 1.

Interactive SQL (JDBC) or ij example

```
create collection my_collection using pattern 'C:\Program
Files\HPTDM\data\ORDERS_D2F*.xm*';
```

Interactive SQL (ODBC) example

```
exec create collection my_collection using pattern
'C:\Program Files\HPTDM\data\ORDERS_D2F*.xm*';
```

5 To create a collection using a schema name you specify, create the collection using a statement similar to the following:

```
create collection <collection_name> in schema
<schema_name> using pattern '<path | connection>\*.xm*';
```

where <collection_name> is the name you choose for your collection, <schema_name> is the name of the schema, and <path | connection> is the location of the files from step 1.

Interactive SQL (JDBC) or ij example

```
create collection my_collection in schema my_schema using
pattern 'C:\Program Files\HPTDM\data\*.xm*';
```

Interactive SQL (ODBC) example

```
exec create collection my_collection in schema my_schema
using pattern 'C:\Program Files\HPTDB\data\*.xm*';
```

A collection is created containing all files in the directory that match the filename pattern, including those contained in gzip files.

TIP HP recommends using different directories for the XML and XSD files created when copying different business flows. If you have copied files from more than one business flow in the same directory, use wildcards to ensure that your pattern matches only files from a single business flow. Any files copied to this directory that match the pattern are automatically added to the collection.

Specifying the path for a local file system

You can use three different types of path notation when you create a collection on a local file system. Depending on where your files are located, the pattern you type will differ:

Notation	Pattern example
local mapped drive	'C:\Program Files\HPTDM\ data*.xm*'
UNC (Universal Naming Convention)	'\\192.168.1.11\NetFolder\HPTDM\data\*.xm*' where 192.168.1.11 is a valid host and NetFolder is the name of a valid share.
URI (Uniform Resource Identifier)	'file:///Z:/HPTDM/data/*.xm*'

NOTE If you are using mapped network drives to store your collection information, you must run the query server as a program.

Dropping collections, schemas, and connections

You can drop collections, schemas, and connections to clean up your data.

NOTE The last collection in a schema cannot be dropped. To remove the last collection, you need to drop the schema.

To drop a collection, use the following syntax:

```
drop collection <collection_name>;
```

where <collection_name> is the name of the collection.

To drop a schema use the following syntax:

```
drop schema <schema_name>;
```

where <schema_name> is the name of the schema.

To drop a connection use the following syntax:

```
drop connection <connection_name>;
```

where <connection_name> is the name of the connection.

Viewing your collection

After creating the collection, you can view it using different methods.

- Viewing your collection from the Interactive SQL or ij clients (page 88)
- Viewing your collection from Microsoft Office Excel (page 90)
- Viewing your collection from OpenOffice.org Base (page 92)
- Viewing database to file data from Oracle (page 94)
- Viewing extracted data from SQL Server (page 100)
- Viewing extraction metadata (page 102)
- Viewing the extraction summary table (page 103)
- Viewing CSV files without using the query server (page 104)
- Viewing limitations (page 105)

Common queries

This section summarizes some of the commonly used queries that you might run against your collections. These queries are described in greater detail in the sections that follow.

Querying schema names

To find out what schemas you have data in, use the following query:

```
select distinct schema from oa_collections;
```

Querying table names

To find out what tables have been created from your collections, use a query similar to the following:

```
select table_qualifier, table_owner, table_name from
oa_tables;
```

Querying from an Oracle database link

Query a database link in Oracle to the XML file:

```
SELECT * FROM order_header@xmlData;
```

where the following database link was previously created in Oracle:

```
create public database link xmlData connect to "install"
identified by "OA" using
'(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<HOST>)
(PORT=<PORT>))(CONNECT_DATA=(SERVICE_NAME=xmlData))
(HS=OK))';
```

Querying from a SQL Server ODBC connection

Query your data from within SQL Server using an ODBC connection to the file:

```
select * from XML_DATA.xmlData.SalesOrderOA.Customer
```

where `xmlData` is the remote link name.

`xmlData` is the XML server name.

`SalesOrderOA` is the cartridge name.

`Customer` is the table name.

Querying metadata

```
select workflowrunid, begintime from order_header where
orderid = 606

select orderid, memfreekb, memtotalkb from
vs.order_header where orderid in (1,100, 1000)

select distinct concat(workflowrunid,
concat(',',begintime)) as runString from vs.order_header
```

Querying summary table

```
select workflow_runid,param_name,param_value from
arcsum_multi where param_name in
('COMPRESSION_ALG','SOURCE_LOCATION','EXTRACT_FORMAT')
```

Viewing your collection from the Interactive SQL or ij clients

After connecting using Interactive SQL or ij, you can use different SQL queries to view the data in your collection.

To find out what schemas you have data in, use the following query:

```
select distinct schema from oa_collections;
```

You should see results similar to the following:

```
schema
MySchema
orders_d2f
```

To find out what tables have been created from your collections, use a query similar to the following:

```
select table_qualifier, table_owner, table_name from
oa_tables;
```

You should see results similar to the following:

Example

```
table_qualifier table_owner     table_name
SCHEMA          SYSTEM          OA_TABLES
SCHEMA          SYSTEM          OA_COLUMNS
SCHEMA          SYSTEM          OA_TYPES
SCHEMA          SYSTEM          OA_INFO
SCHEMA          SYSTEM          OA_STATISTICS
SCHEMA          SYSTEM          OA_FKEYS
SCHEMA          SYSTEM          OA_PROC
SCHEMA          SYSTEM          OA_PROCCOLUMNS
xmlData         MySchema        arcsum_MySchema
xmlData         MySchema        COMMPLAN
xmlData         MySchema        CUSTOMER
xmlData         MySchema        ORDER_ATTACHMENT
xmlData         MySchema        ORDER_HEADER
xmlData         MySchema        ORDER_LINE
xmlData         MySchema        ORDER_LINE_DIST
xmlData         MySchema        ORDER_PAYMENT
xmlData         MySchema        ORDER_PAYMENT_LINE
xmlData         MySchema        ORDER_STAR
xmlData         MySchema        ORDER_TAX
xmlData         MySchema        PRODUCT
xmlData         MySchema        SALESREP
xmlData         MySchema        STATUS
xmlData         MySchema        STORE
xmlData         OASYSTEM        OA_COLLECTIONS
xmlData         OASYSTEM        OA_CONNECTIONS
xmlData         OASYSTEM        OA_FILES
xmlData         OASYSTEM        OA_INDEXES
xmlData         OASYSTEM        OA_OPTIONS
xmlData         OASYSTEM        OA_PRIVILEGES
xmlData         OASYSTEM        OA_ROLES
xmlData         OASYSTEM        OA_USERS
xmlData         ORDERS_D2F      arcsum_ORDERS_D2F
xmlData         ORDERS_D2F      CUSTOMER
xmlData         ORDERS_D2F      ORDER_ATTACHMENT
xmlData         ORDERS_D2F      ORDER_HEADER
xmlData         ORDERS_D2F      ORDER_LINE
xmlData         ORDERS_D2F      ORDER_TAX
xmlData         ORDERS_D2F      PRODUCT
xmlData         ORDERS_D2F      SALESREP
xmlData         ORDERS_D2F      STATUS
```

- The table qualifier column is the name of the data source.
- The table owner column is the name of the cartridge used to extract the data, the schema name, or the user if you created a view.
- The SYSTEM and OASYSTEM tables are system tables that contain information on the contents of the searchable data.
- The arcsum_<schema_name> contains all of the configuration and runtime parameter information for all collections within that particular schema.

Viewing your collection from Microsoft Office Excel

You can view your collection in Microsoft Office Excel using the ODBC driver. You can use the Query Wizard to select what data is displayed, or you can use Microsoft Query to write your own queries.

NOTE Microsoft Office Excel does not support binary data or Unicode text. If your CSV file contains either of these types of data, you will not be able to import that data into Excel.

TIP If an object name contains special characters or punctuation marks, use quotation marks around the object names and end the query with a semicolon. For example, the following code:

```
SELECT ORDER_HEADER.ORDER&ID, ORDER_HEADER.DEPTNO+1
FROM xmlData.ORDER_OA.ORDER_HEADER
```

should be written as:

```
SELECT “ORDER_HEADER”.“ORDER&ID”, “ORDER_HEADER”.“DEPTNO+1”
FROM “xmlData”.“ORDER_OA”.“ORDER_HEADER” “ORDER_HEADER”;
```

In this section:

- Viewing your collection using the query wizard (page 90)
- Viewing your collection using Microsoft Query (page 91)

Viewing your collection using the query wizard

1 Ensure the ODBC database driver has been installed and configured.

See also Installing and configuring the ODBC or OLE DB driver (page 76)

2 Start Microsoft Office Excel.

3 Navigate to the Choose Data Source window.

Example Data > Import External Data > New Database Query

4 Select the name of your ODBC source from the list of values.

5 Use the Query Wizard to created your queries.

a Select the **Use the Query Wizard to create/edit queries** checkbox.

b Click **OK**.

The OpenAccess Login window opens.

c Enter the user name and password for your user account.

Field	Expected value
Data Source User Name	install
Data Source Password	OA
Custom Properties	leave blank

The Query Wizard - Choose Columns window opens.

d Select the tables or columns you want to include.

e Click **Next**. The Query Wizard - Filter Data window opens.

f Optionally, specify a filter to restrict the data.

g Click **Next**. The Query Wizard - Sort Order window opens.

h Optionally, specify the columns you want to use to sort the data.

i Click **Next**. The Query Wizard - Finish window opens.

6 Specify how the data should be returned into Excel.

7 Click **Finish**. The Import Data window opens.

8 Specify where you want the data returned.

9 Click **OK**. The data is loaded into the spreadsheet.

Viewing your collection using Microsoft Query

1 Ensure the ODBC database driver has been installed and configured.

See also Installing and configuring the ODBC or OLE DB driver (page 76)

2 Start Microsoft Office Excel.

3 Navigate to the Choose Data Source window.

Example Data > Import External Data > New Database Query

4 Select the name of your ODBC source from the list of values.

5 Ensure the **Use the Query Wizard to create/edit queries** checkbox is not selected.

6 Click **OK**.

The OpenAccess Login window opens.

7 Enter the user name and password for your user account.

Field	Expected value
Data Source User Name	install
Data Source Password	OA
Custom Properties	leave blank

The Microsoft Query window opens.

8 Enter a SQL query into Microsoft Query.

Example

```
select * from oa_users
```

The results of the query are displayed in Excel.

Viewing your collection from OpenOffice.org Base

You can view your connection in OpenOffice.org Base using either the ODBC or JDBC driver.

In this section

- Connecting to OpenOffice.org Base using ODBC (page 92)
- Connecting to Open Office.org Base using JDBC (page 93)

Connecting to OpenOffice.org Base using ODBC

1 If you plan on using the ODBC driver, ensure that the ODBC database driver has been installed and configured.

See also Installing and configuring the ODBC or OLE DB driver (page 76)

2 Ensure the server process is running.

See also Starting the query server (page 74)

3 Start the OpenOffice.org Base program.

The Database Wizard window opens.

4 Connect to the database.

a Click the **Connect to an existing database** radio button.

b Select ODBC from the list of values.

c Click **Next**.

d On the Set up a connection to an ODBC database page, click **Browse**. The Data Source window opens.

e Select the data source and click **OK** to close the window.

f Click **Next** to select the data source.

g Enter the user name and select the Password required checkbox. The default user name is install, and the password is OA.

h Click **Test Connection**.

The login window opens.

i Enter the password and click **OK** to test the connection.

j Click **OK** on the confirmation dialog.

k Click **Finish**.

5 Save the data source as an ODBC database.

The Base program is connected to the server and ready to use.

Connecting to Open Office.org Base using JDBC

1 Ensure the server process is running.

See also Starting the query server (page 74)

2 Start the OpenOffice.org Calc program.

3 Configure the Java options.

   a Select **Tools > Options**.

   The Options window opens.

   b Select Java from the navigation tree.

   c Click **Class Path**. The Class Path window opens.

   d Click **Add Archive**.

   e Navigate to the following directory:

   <install_directory>/obt/lib

   f Select oajc.jar and click **Open**.

   g Click **OK** on the Class Path window.

   h Click **OK** to return to the Calc program.

4 Exit all OpenOffice.org programs.

5 Connect to the database.

See also Connecting using the Interactive JDBC client (ij) (page 81)

   a Start the OpenOffice.org Base program.

   The Database Wizard window opens.

   b Click the **Connect to an existing database** radio button.

   c Select JDBC from the list of values.

   d Enter the following JDBC database connection values:

Field	Expected value
Datasource URL	`jdbc:openaccess://<host>:<port>;serverDataSource=<datasource_name>;`
JDBC driver class	`com.ddtek.jdbc.openaccess.OpenAccessDriver`

   e Click **Next**.

   f Enter the user name and select the **Password required** checkbox. The default user name is install, and the password is OA.

   g Enter the password and click **OK** to test the connection.

   h Click **OK** on the confirmation dialog.

   i Click **Finish**.

6 Save the data source as a JDBC database. The Base program is connected to the server and ready to use.

Viewing database to file data from Oracle

With Oracle's Heterogeneous Services, you can access XML data from within an Oracle database. This enables you to use advanced SQL functions against the data, and even join the data with your source data for advanced querying.

TIP Oracle treats all columns and table names as upper-case. If your collection contains columns or tables that are in mixed-case or lower-case, you need to use double-quotes in your query.
For example, `select column_name from table_name` returns a column called `COLUMN_NAME`, whereas `select "column_name" from table_name` returns a column called `column_name`.

In this section

- Configuring Heterogeneous Services (page 94)
- Configuring Heterogeneous Services for releases prior to Oracle 11g on 64-bit operating systems (page 96)
- Creating an ODBC database link (page 99)
- Querying your data from Oracle (page 99)

Configuring Heterogeneous Services

To configure the Heterogeneous Services agent process, you need to perform the following tasks:

- Creating an initialization file (page 94)
- Modifying the listener and tnsnames files (page 95)

NOTE If you are running a version of Oracle prior to 11g on a 64-bit operating system, follow the instructions in Configuring Heterogeneous Services for releases prior to Oracle 11g on 64-bit operating systems (page 96).

Creating an initialization file

To create the initialization file:

1 Navigate to the following directory:

```
$ORACLE_HOME/hs/admin
```

2 Create a new text file, `initxmlData.ora`, that contains the following text:

Mode	Use

UNIX	`HS_FDS_CONNECT_INFO = xmlData`
	`HS_AUTOREGISTER = TRUE`
	`HS_DB_NAME = hsodbc`
	`HS_FDS_SHAREABLE_NAME = <AQSODBC_install>/obt/sqlxml/client/lib/libodbc.so`
	`set ODBCINI=/<AQSODBC_install>/client/odbc.ini`
	`set LD_LIBRARY_PATH=/<AQSODBC_install>/client/lib:/<oracle-Home>/10.2.0.4/lib32`
	`set OASDK_ODBC_HOME=/<AQSODBC_install>/client/lib`

Windows	`set HS_FDS_CONNECT_INFO = xmlData`
	`set HS_AUTOREGISTER = TRUE`
	`set HS_DB_NAME = hsodbc`

where <AQSODBC_install> is the AQS ODBC drivers are installed.

NOTE For Oracle 11*g*, change the value of the `HS_DB_NAME` entry to `dg4odbc`.

If you use a different name than xmlData for the data source, edit the file and the file name accordingly. For example, if you used the name MyData, the file should be named `initMyData.ora`, and the value for HS_FDS_CONNECT_INFO should be MyData.

You can use the path variable appropriate to your environment to point to the necessary library for your operating system.

Operating System	Path variable
IBM AIX	LIBPATH
HP-UX	SHLIB_PATH
HP-UX on Intel Itanium	LD_LIBRARY_PATH
Linux	LD_LIBRARY_PATH
Solaris	LD_LIBRARY_PATH
MS Windows	PATH

Modifying the listener and tnsnames files

To set up the listener on the agent to listen for incoming requests from the Oracle Database server:

1 Navigate to the directory that contains the listener.ora file.

```
cd $ORACLE_HOME/network/admin
```

2 Edit the `listener.ora` file to add the entries for the data source, for example:

Example on MS Windows

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = xmlData)
      (ORACLE_HOME = C:\oracle\product\10.2.0\db_1)
      (PROGRAM = hsodbc)
   )
```

If you have a different Oracle home directory, edit the path accordingly.

NOTE For Oracle 11*g*, change the value of the `PROGRAM` entry to `dg4odbc`.

3 Ensure that the initialization parameter GLOBAL_NAMES is set to FALSE in the database initialization parameter file.

4 Edit the `tnsnames.ora` file to add the appropriate entry, for example:

TIP You do not need to add anything to `tnsnames.ora` on MS Windows.

Example on Unix

```
HSALIAS = (DESCRIPTION=
(ADDRESS=(PROTOCOL=tcp)(HOST=<oracle_host>)(PORT=1521))
(CONNECT_DATA=(SID=xmlData))
(HS=OK)
)
```

5 Restart the TNS Listener. When a request is received, the agent now spawns a Heterogeneous Services agent.

You are now ready to create a database link as per the instructions in Creating an ODBC database link (page 99).

Configuring Heterogeneous Services for releases prior to Oracle 11g on 64-bit operating systems

When configuring the query server with Heterogeneous Services in releases prior to Oracle 11g on a 64-bit operating system, you must specify the 32-bit library using the path variable appropriate to your environment.

TIP If you cannot locate the 32-bit driver appropriate for your platform, contact HP Support.

Example on Solaris

On Solaris with Oracle 10.2.0.4, your `initXMLData.ora` in `$ORACLE_HOME/hs/admin` would look similar to the following:

```
HS_FDS_CONNECT_INFO = xmlData
HS_FDS_SHAREABLE_NAME = /<AQSODBC_install>/client/lib/
libodbc.so
set ODBCINI=/<AQSODBC_install>/client/odbc.ini
set LD_LIBRARY_PATH=/<AQSODBC_install>/client/lib:/
<oracle_home>/10.2.0.4/lib32
set OASDK_ODBC_HOME=/<AQSODBC_install>/client/lib
```

where <AQSODBC_install> is the directory where the 32-bit AQS ODBC drivers for Solaris are installed.

Your `listener.ora` would look similar to the following:

```
SID_LIST_LISTENER =
 (SID_DESC =
  (SID_NAME = xmlData)
   (ORACLE_HOME = /remote/app/oracle/ra11510o/10.2.0.4)
   (PROGRAM = hsodbc)
   (ENVS=LD_LIBRARY_PATH=/<AQSODBC_install>/client/lib:
    /<Oracle_Home>/10.2.0.4/lib32)
 )
```

where <AQSODBC_install> is the directory where the 32-bit AQS ODBC drivers for Solaris are installed.

Your `tnsnames.ora` would look similar to the following:

```
HSALIAS = (DESCRIPTION=
 (ADDRESS=(PROTOCOL=tcp)(HOST=stingray)(PORT=1527))
          (CONNECT_DATA=(SID=xmlData))
          (HS=OK)
           )
```

Your `odbc.ini` would look similar to the following:

```
[xmlData]
Driver=/<AQSODBC_install>/client/lib/ivoa22.so
Description=DataDirect OpenAccess SDK 6.0
Host=stingray
Port=19988
ServerDataSource=xmlData
UseLDAP=0
DistinguishedName=
Encrypted=0
LoadBalancing=0
AlternateServers=
ConnectionRetryCount=0
ConnectionRetryDelay=3
CustomProperties=
[ODBC]
Trace=0
IANAAppCodePage=4
TraceFile=odbctrace.out
TraceDll=/<AQSODBC_install>/client/lib/odbctrac.so
InstallDir=/<AQSODBC_install>/client
```

Where <AQSODBC_install> is the directory where the 32-bit AQS ODBC drivers for Solaris are installed.

Example on AIX

On AIX with Oracle 10.2.0.4, your `initXMLData.ora` would look similar to the following:

```
HS_FDS_CONNECT_INFO=xmlData
HS_FDS_SHAREABLE_NAME=/home/verducci/TIGER/AQS_CLIENTS
/obt/sqlxml/client/lib/libodbc.so
set ODBCINI=/home/verducci/TIGER/AQS_CLIENTS/obt/sqlxml
/client/odbc.ini
```

```
set OASDK_ODBC_HOME=/home/verducci/TIGER/AQS_CLIENTS/obt
/sqlxml/client/lib
set LIBPATH=/home/verducci/TIGER/AQS_CLIENTS/obt/sqlxml
/client/lib:/remote/app/oracle/product/10.2.0.4/lib32
```

Your `listener.ora` would look similar to the following:

```
(SID_DESC =
 (SID_NAME = xmlData)
 (ORACLE_HOME = /remote/app/oracle/product/10.2.0.4)
 (PROGRAM = hsodbc)
 (ENVS=LIBPATH =/home/verducci/TIGER/AQS_CLIENTS
   /obt/sqlxml/client/lib:/remote/app/oracle/product
   /10.2.0.4/lib32)
    )
```

Your `tnsnames.ora` would look similar to the following:

```
HSALIAS = (DESCRIPTION=
          (ADDRESS=(PROTOCOL=tcp)(HOST=tiger)(PORT=1523))
          (CONNECT_DATA=(SID=xmlData))
          (HS=ok)
          )
```

Your `odbc.ini` would look similar to the following:

```
[ODBC Data Sources]
xmlData=DataDirect OpenAccess SDK 6.0
[xmlData]
QEWSD=40049
Driver=/home/verducci/TIGER/AQS_CLIENTS/obt/sqlxml
  /client/lib/ivoa22.so
Description=DataDirect OpenAccess SDK 6.0
Host=tiger
Port=19989
ServerDataSource=xmlData
UseLDAP=0
DistinguishedName=
Encrypted=0
LoadBalancing=0
AlternateServers=
ConnectionRetryCount=0
ConnectionRetryDelay=3
CustomProperties=
[ODBC]
Trace=0
IANAAppCodePage=4
TraceFile=odbctrace.out
TraceDll=/home/verducci/TIGER/AQS_CLIENTS/obt/sqlxml
  /client/lib/odbctrac.so
InstallDir=/home/verducci/TIGER/AQS_CLIENTS/obt/sqlxml
  /client
```

Example on MS Windows

On MS Windows with Oracle 10.2.0.4, your `initXMLData.ora` would look similar to the following:

```
HS_FDS_CONNECT_INFO = xmlData
HS_AUTOREGISTER = TRUE
HS_DB_NAME = hsodbc
```

Your listener.ora would look similar to the following:

```
(SID_DESC =
    (SID_NAME = xmlData)
    (ORACLE_HOME = C:\oracle\product\10.2.0\db_2)
    (PROGRAM = hsodbc)
    )
```

TIP You do not need to add anything to tnsnames.ora on MS Windows.

Creating an ODBC database link

1 Invoke SQL*Plus and log in as a user with privileges to create a database link. For example, SYSTEM.

2 Create a database link using the following syntax:

```
create public database link xmlData connect to "install"
identified by "OA" using
'(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<HOST>)
(PORT=<PORT>))(CONNECT_DATA=(SERVICE_NAME=xmlData))
(HS=OK))';
```

Where	Is
HOST	Enter the name of the machine where Oracle is installed.
PORT	Enter the port number for your Oracle database that the TSN Listener is connected to. For example, 1521.

Related information See your Oracle documentation.

Querying your data from Oracle

1 Invoke SQL*Plus and log in as a user with privileges to access the xmlData data.

2 Query your data using the database link.

Example

```
SELECT * FROM order_header@xmlData;
```

TIP In Oracle 10g and earlier, Unicode data types are not supported by Oracle Heterogeneous Services. To get support for Unicode data types, you must upgrade to Oracle Gateway 11g (DG4ODBC).

Viewing extracted data from SQL Server

Using a linked server configuration allows you to access extracted data from within SQL Server. This enables you to use advanced SQL functions against the data, and even join the data with your source data for advanced querying.

In this section

Creating the ODBC connection

1. Start the SQL Server Management Studio.
2. Expand the Server Objects node.
3. Right-click Linked Servers and select **New Linked Server**. The New Linked Server window opens.
4. On the General panel, enter or select the following information:

Field	Expected value
Linked server	XML_DATA
Server type	Other data source
Provider	Microsoft OLE DB Provider for ODBC Drivers
Product name	Choose a name for the product. For example, sqlXML.
Data source	xmlData
Provider string	leave blank
Location	leave blank
Catalog	leave blank

5. On the Security panel, click **Add**.
6. Enter or select the following information for the local server login:

Field	Expected value
Local Login	Enter the login for the server you want to access the XML server. For example, sa.
Remote User	Enter the username of the xmlData server. For example, install.
Remote Password	Enter the password for the user you selected. For example, OA.

Leave the **Impersonate** check box unchecked.

7. Click **OK**.

Querying your data from SQL Server

1. Start the SQL Server Management Studio.
2. Expand the Databases node.
3. Right-click on the database you configured and select **New Query**.
4. Query your data using the ODBC connection.

Example

```
select * from <remote_link_name>.<xml_server_name>.
<cartridge_name>.<table_name>
```

Where	Is
remote_link_name	The name of the linked server specified in Creating the ODBC connection (page 100). For example, XML_DATA.
xml_server_name	The name of the XML server. For example, xmlData.
cartridge_name	The name of the database to file cartridge.
table_name	The name of the table you want to query. For example, Customer.

Example

```
select * from XML_DATA.xmlData.SalesOrderOA.Customer
```

Viewing extraction metadata

The group XML file contains metadata about the extraction process, such as the start time and the workflow run ID. This information is available in queries using scalar functions.

The following functions expose group level metadata:

- workflowRunId
- sequence
- beginTime
- archiveGuid
- groupGuid
- productName
- productVersion
- runGuid
- cartridgeName
- cartridgeVersion
- sourceCategory
- sourceGuid
- sourceDatabase
- sourceDBServer
- sourceDriverType
- sourceHost
- sourcePort
- sourceUserID
- sourceStatus
- sourceType
- destinationCategory
- destinationGuid
- destinationDirectory
- destinationFile
- destinationHost
- destinationUserID
- destinationStatus
- destinationType

Example query

```
select workflowrunid, begintime from order_header where
orderid = 606
```

Example result

```
workflowrunid()  begintime()
---------------  ----------------------------------------
11               2007-11-07T18:56:41.000000000-08:00
11               2007-11-07T18:56:42.000000000-08:00
11               2007-11-07T18:57:24.000000000-08:00
11               2007-11-07T18:57:25.000000000-08:00
11               2007-11-07T18:57:26.000000000-08:00
```

The following functions read information from the Java virtual machine:

- MemFreeKB
- MemTotalKB

Example query

```
select orderid, memfreekb, memtotalkb from vs.order_header
where orderid in (1,100, 1000)
```

Example result

```
orderid memfreekb() memtotalkb()
------- ----------- ------------
1       3202        5184
100     3190        5184
1000    3190        5184
```

The following function concatenates the CHARACTER arguments into a single string:

- CONCAT(char1,char2)

Example query

```
select distinct concat(workflowrunid, concat(',',begintime))
as runString from vs.order_header
```

Example result

```
runString
---------------------------------------
11,2007-11-07T18:56:41.000000000-08:00
11,2007-11-07T18:56:42.000000000-08:00
11,2007-11-07T18:57:24.000000000-08:00
11,2007-11-07T18:57:25.000000000-08:00
11,2007-11-07T18:57:26.000000000-08:00
```

Viewing the extraction summary table

Each summary XML file is a summary of the extraction process and data movement history for one run. The summary table contains much of the data from the summary XML file, including:

- summary of extraction groups,
- summary of data movements,
- what the user parameters were set to when the extraction was run
- the source location
- the destination location

The summary table contains the following columns:

- workflow_Runid
- task_GUID
- archive_GUID
- appsPack_Name
- appsPack_Version
- group_GUID
- group_sequence
- group_instance_count
- mv_Type
- mv_Product_Version
- mv_Workflow_Runid
- mv_Task_GUID
- mv_Begin_Time
- source_GUID
- source_Category
- source_Type
- source_Host
- source_User_Id
- source_Desc
- dest_GUID
- dest_Category
- dest_Type
- dest_Hostdest_User_Id
- dest_Desc
- param_name
- param_scope
- param_type
- param_datatype
- param_default_value
- param_value

Example query

```
select workflow_runid,param_name,param_value from
arcsum_multi where param_name in
('COMPRESSION_ALG','SOURCE_LOCATION','EXTRACT_FORMAT')
```

Example result

```
workflow_runid  param_name       param_value
--------------  ---------------  -----------
7               COMPRESSION_ALG  NONE
7               SOURCE_LOCATION  OBTINTF_DB
7               EXTRACT_FORMAT   XML
9               COMPRESSION_ALG  GZIP
9               SOURCE_LOCATION  OBTINTF_DB
9               EXTRACT_FORMAT   CSV
```

Viewing CSV files without using the query server

Although you can import CSV files directly into Microsoft Office Excel, many of the datatypes available in relational databases are not supported. Because your HP Test Data Management CSV files were created from a database, they may not be interpreted correctly if you use a spreadsheet to read the CSV files directly. In addition, the CSV files do not contain column headers. You have to add these manually yourself.

The following limitations apply to CSV files containing binary data and unicode text:

- You cannot open CSV files created by HP Test Data Management directly with Excel. They must be imported before you can view them.
- Excel does not support reading, manipulating, or storing binary data. If you import a CSV file with binary fields, the data in those fields will not be usable or recognizable in Excel.
- Excel 2003 requires a byte order marker (BOM) be added to CSV files with unicode text before importing. See your Professional Services representative for more information.

To import CSV files into Microsoft Excel 2007:

1. From Excel 2007, select **Data > Get External Data > From Text**.
2. Select the file you want to import.
3. In the Text Import wizard, ensure the following are set before you import:
   - File Origin is set to 65001: Unicode (UTF-8)
   - Delimiter is set to comma

To import CSV files into OpenOffice.org Calc 3.1:

1. From OpenOffice.org 3.1 Calc, select **File > Open**.
2. Select the file you want to import.
3. In the Text Import window, change the Character Set to Unicode (UTF-8).
4. Click **OK**.

Viewing limitations

The following limitations exist for viewing the extracted data:

Large object limitations in the xmlData

Because large objects are stored in memory differently than other data, there are limitations when accessing them. Large objects in Oracle include CLOBs, BLOBs, LONG and LONG RAW datatypes. Large objects in SQL Server include image, text, ntext, nvarchar(max), varchar(max), and XML datatypes.

Large object columns can only be used in a WHERE clause if used in conjunction with an IS NULL or IS NOT NULL operator. The following functions cannot be used to query columns with large objects:

- group by
- distinct
- order by
- set

Extracting binary and varbinary data to XML

When binary and varbinary data is extracted, it is stored in base64 format in the CSV or XML data files generated by running database to file extraction.

Example

original column	**original row value**	**XML column and value**
binary_type	0x1111111111	<binary_type>ERERERE= </binary_type>

The data is returned to its original form when restored to a database or accessed by the query server.

NOTE Not all clients display binary data properly.

Multi-byte characters in views

Views cannot be created from tables with multi-byte characters in the tablename, and multi-byte characters cannot be used in the name of the view.

Uninstalling the query server

Use the oauninstall script to uninstall the query server and the client drivers. If you want to uninstall only the query server, or only the client drivers, add -server or -client to the syntax.

NOTE If you uninstall HP Test Data Management, the query server is uninstalled as well.

1 Navigate to the directory that contains the uninstall file.

Example

```
cd <install_directory>/obt/bin
```

where <install_directory> is the location where the software was installed.

2 Run the appropriate script:

To	On	Use the following syntax
uninstall both the query server and the client drivers	DOS	`oauninstall.bat`
	UNIX	`./oauninstall.sh`
uninstall the query server only	DOS	`oauninstall.bat -server`
	UNIX	`./oauninstall.sh -server`
uninstall the client drivers only	DOS	`oauninstall.bat -client`
	UNIX	`./oauninstall.sh -client`

TIP If any files remain in the <install_directory>/obt/sqlxml/client/ or <install_directory>/obt/sqlxml/server/ directories, you can delete them manually.

If you have installed either the ODBC driver or the OLE DB driver, you can use the oadriver script to uninstall the drivers.

3 Navigate to the directory that contains the ODBC driver installation file.

Example

```
cd <install_directory>/obt/bin
```

where <install_directory> is the location where the software was installed.

4 Run the appropriate script:

For	Syntax
DOS	`oadriver.bat`
UNIX	`./oadriver.sh`

5 Select Option 2 to uninstall the driver.

6 For DOS, select Option 1 to uninstall the ODBC driver or Option 2 to uninstall the OLE DB driver.

The ODBC or OLE DB database driver is uninstalled.

9 Query server administrative tasks

The following administrative tasks can help you to manage your AQS environment:

- Using multiple data sources (page 107)
- Understanding users, roles, and privileges (page 111)
- Managing indexes (page 113)
- Managing server options (page 115)
- Viewing query server log files (page 119)

Using multiple data sources

The xmlData data source is created by default when you install the query server. Adding additional data sources allows you to:

- change the name of the data source from xmlData
- split your test data to improve access performance
- create separate access to test data along organizational lines

In this section

- Adding a data source (page 108)
- Dropping a data source (page 108)
- Configuring a new data source for ODBC on UNIX (page 109)
- Relocating a data source (page 110)

Adding a data source

You can add one or more data sources. All data sources must have a unique name, and must be stored in a unique directory.

1 Navigate to the directory that contains the add data source script.

Example

```
<install_directory>/obt/bin
```

where <install_directory> is the location where the software was installed.

WARNING! Use of the following special characters with the `addoadb` scripts may result in an operating system error:
` $! ^ &) (\ ' = | ; " < > ,
You should avoid these characters wherever possible when adding a data source.

2 Run the script using the appropriate syntax:

For	Syntax
DOS	`addoadb.bat <datasource_name> [datasource_root_path]`
UNIX	`./addoadb.sh <datasource_name> [datasource_root_path]`

Where	is
datasource_name	the name of the new data source you want to create.
datasource_root_path	The optional path where you want to place the data source. By default, the data source is placed in the following path: <install_directory>/obt/sqlxml/<datasource_name> where <install_directory> is the location where you installed the software.

Dropping a data source

Dropping a data source removes the following:

- the data source
- the data source root directory
- the collection configuration created for this data source

All XML data related to the data source is retained.

1 Navigate to the directory that contains the drop data source script.

Example

```
<install_directory>/obt/bin
```

where <install_directory> is the location where the software was installed.

2 Run the script using the appropriate syntax:

For	Syntax
DOS	`dropoadb.bat <datasource_name>`
UNIX	`./dropoadb.sh <datasource_name>`

where <datasource_name> is the name of the data source you want to drop.

Configuring a new data source for ODBC on UNIX

The following instructions are for adding a data source called MyDataSource. Change the name as appropriate to your environment.

To configure a new data source for ODBC on Windows, follow the instructions in Installing and configuring the ODBC database driver on Windows (page 76).

For JDBC, there is no need to configure the data source. Following the instructions in Connecting using the Interactive SQL (JDBC) client (page 80) or Connecting using the Interactive JDBC client (ij) (page 81).

1 Navigate to the directory that contains the odbc.ini file.

Example

```
<install_directory>/obt/sqlxml/client/
```

where <install_directory> is the location where the software was installed.

2 Open the odbc.ini file with a text editor, and locate the [ODBC Data Sources] and [ODBC_datasource_name] sections.

Example of odbc.ini file

```
[ODBC Data Sources]
ODBCDataSource=DataDirect OpenAccess SDK 6.0
[ODBCDataSource]
Driver=/home/myhome/sqlxml/client/lib64/ivoa22.so
Description=DataDirect OpenAccess SDK 6.0
Host=localhost
Port=19988
ServerDataSource=xmlData
UseLDAP=0
DistinguishedName=
Encrypted=0
LoadBalancing=0
AlternateServers=
ConnectionRetryCount=0
ConnectionRetryDelay=3
CustomProperties=
```

3 Add the following entry to the data sources block:

```
MyDataSource=Customer ODBC Driver 6.0
```

Example

```
[ODBC Data Sources]
ODBCDataSource=DataDirect OpenAccess SDK 6.0
MyDataSource=Customer ODBC Driver 6.0
[ODBCDataSource]
```

4. Duplicate the [ODBCDataSource] block and replace the existing bold values with the appropriate values for the MyDataSource data source:

Example

```
[ODBC Data Sources]
ODBCDataSource=DataDirect OpenAccess SDK 6.0
MyDataSource=Customer ODBC Driver 6.0
[ODBCDataSource]
Driver=/home/myhome/sqlxml/client/lib64/ivoa22.so
Description=DataDirect OpenAccess SDK 6.0
Host=localhost
Port=19988
ServerDataSource=xmlData
UseLDAP=0
DistinguishedName=
Encrypted=0
LoadBalancing=0
AlternateServers=
ConnectionRetryCount=0
ConnectionRetryDelay=3
CustomProperties=
[MyDataSource]
Driver=/home/myhome/sqlxml/client/lib64/ivoa22.so
Description=Customer ODBC Driver 6.0
Host=<host_name>
Port=<port_name>
ServerDataSource=MyDataSource
```

Where	Is
host_name	The name of the host machine or IP address on which the query server is running.
port_name	The port number used by the remote query server you are connecting to. The default value is 19988.

5. Save the odbc.ini file.

Relocating a data source

You can relocate a currently existing data source to a different directory by dropping the existing data source and creating a new one.

NOTE All data in your collections must be reindexed on the new data source. See Managing indexes (page 113) for more information.

1. Drop the data source using the instructions in Dropping a data source (page 108).
2. Move the data source files to the new location.
3. Add the new data source using the instructions in Adding a data source (page 108).

4 Configure the data source to point to the new location using the instructions in Configuring a new data source for ODBC on UNIX (page 109).

Understanding users, roles, and privileges

The query server uses users, roles, and privileges to establish individual permission to access the xmlData.

users	an individual with specific privileges and a password.
roles	a grouping of users with the same privileges. You can assign privileges to each individual user, or grant the privileges to a role, and assign that role to each user.
privileges	granting of access to a database table.

By default, the query server is installed with the install user that has the DBA role.

NOTE When using Interactive SQL (ODBC), the privileges and roles statements require a prefix of 'exec' or '!'. This is not required for Interactive SQL (JDBC) or Interactive SQL (OLE DB).

Managing users

Use standard SQL statements to manage users.

NOTE Do not use multi-byte characters when you create the user or password.

- To view existing users:

```
select * from oa_users;
```

- To create a new user:

```
create user <user_name> identified by <password>;
```

- To change the password of an existing user:

```
alter user <user_name> identified by <password>;
```

- To grant a role to an existing user:

```
grant <role_name> to <user>;
```

- To revoke a role from an existing user:

```
revoke <role_name> from <user_name>;
```

Where	Is
user_name	the name of the user you want to create or alter. By default, all new users are created with the NO_ACCESS role.
password	the password for the user.
role_name	the role you want to grant to the user.

Managing roles

The following roles exist at installation:

Role	Description
DBA	Grants all permissions on all objects, including: • creating and dropping users and roles • creating and dropping collections • changing user passwords • creating and dropping indexes • setting and removing privileges
NO_ACCESS	The default role granted to new users. The NO_ACCESS role can connect to the xmlData, but has no other permissions.

- To create a new role:

```
create role <role_name>;
```

- To drop an existing role:

```
drop role <role_name>;
```

Where	Is
role_name	the name of the role you are creating or dropping.

Managing privileges

By default, the following privileges are available:

READ	grants read access to a single table.
READ_ANY	grants read access to all tables in any schema.
READ_SYSTEM	grants read access to all system tables.

If a user has access to all of the tables in a particular view, then the view can also be accessed.

- To grant read access for one or more tables:

```
set privilege READ on {<object_name>|ALL} (object_type)
to (<user_name | role_name>);
```

Example

```
set privilege READ on CUSTOMER table to USER1;
```

- To grant read access to all tables:

```
set privilege READ_ANY to <user_name | role_name>;
```

- To grant read access to all system tables:

```
set privilege READ_SYSTEM to <user_name | role_name>;
```

- To remove a privilege from a user or role:

```
remove privilege READ_SYSTEM from <user_name |
role_name>;
```

Where	Is
user_name	the name of the user you want to grant privileges to.
role_name	the name of the role you want to grant privileges to.
object_name	the name of the table you want to grant read access to.
object_type	the type of object you want to grant read access to. The expected value is table.

NOTE Because ij has a system use for the remove command, when removing privileges in ij, use single quotes to separate the statement.

```
'remove privilege READ_SYSTEM from USER1';
```

Managing indexes

The query server uses indexes to optimize your SQL queries. Any index defined when you created the cartridge is included in the collection by default. The pre-created indexes can be disabled or dropped. Standard range indexes can be created at any time by a user with the DBA role. The indexes are built when the table is queried, or the administrator runs the build range index command. Indexes do not modify the XML or XSD files

To see information about existing indexes, query the oa_indexes table. The oa_indexes table displays the following information:

Field	Description
table	The table affected by the index.
index_name	The name of the index.
builtin	Displays T if the index was created as part of the cartridge. Displays F if the index was created after the table was extracted.
enabled	Displays T if the index is currently enabled, and F if the index is disabled. Disabled indexes are not visible to the query optimizer. To see information about the index that is available to the query optimizer, query against the oa_statistics table.

NOTE When using Interactive SQL (ODBC), the create index statements require a prefix of 'exec' or '!'. This is not required for Interactive SQL (JDBC) or Interactive SQL (OLE DB).

- To add an index:

  ```
  create range index <index_name> on <table_name>(<Column
  list>);
  ```

- To drop an index:

  ```
  drop range index <index_name>;
  ```

- To enable an index:

  ```
  enable index <index_name>;
  ```

- To disable an index:

  ```
  disable index <index_name>;
  ```

- To build an index:

  ```
  build range index on [all | schema <schema_name> | table
  <table_name>]
  ```

  — Use `all` to build all indexes currently defined on the system.

  — Use `schema <schema_name>` to build all indexes defined within that schema.

  — Use `table <table_name>` to build all indexes defined for that table.

  NOTE You cannot create indexes on system tables or system schemas.

Managing server options

Users with the DBA role can modify the behavior of the server through the use of options.

NOTE Overriding data types and server options is global to the database, and all occurrences are converted.

In this section

Understanding the override options

Users with the DBA role can modify the behavior of the server through the use of options. The server features that can be controlled through options include the number of threads used for searching data, how to treat character data with embedded nulls, and memory usage for column storage.

The following options are available:

- threads

  A positive whole number representing the number of independent threads that should be used for reading data. The default value, 8, works well on systems with up to four processors. If your system has more than four processors, this number can be increased up to two times the number of physical processors.

  NOTE Larger thread numbers can decrease performance.

- type

  You can override source database data types in order to report a different data type to the client.

  See Understanding database data type override options (page 116).

- colMaxSize

  This value is the maximum internal memory size used for storage of a single column value. The default value, 32768, should only be changed if you experience memory issues. The colMaxSize option defines when to stop using physical memory and start using virtual memory. The minimum size is 1024 and the maximum size is 4194304.

- ignoreNull

  When set to true, ignores any null values in a column with character data without raising an error. If set to false, returns an error when null values exist in a column with character data. The default value is false.

- duplicateObjectRename

The duplicateObjectRename option controls whether or not duplicate table names are renamed in a multi-cartridge schema. The default value, false, returns an error message. Set the duplicateObjectRename option to true to rename the duplicate tables. This option can only be changed by a DBA user, and is global for the entire database.

- showOverrideNames

  When set to false, the original source database column names are reported back to the client. When set to true, if column names were overridden in order to make valid XML tags, then the new column name is reported to the client. This is global for the entire database and takes effect immediately. The default value is false.

- searchSchema

  The searchSchema option allows you to specify which schema you want to search. Any user can use the searchSchema option, and it is only valid for that individual user.

If you query on:	Default behavior	Behavior with searchSchema option
A table name that is fully-qualified	Run the query.	Run the query.
A table that is not fully-qualified, but only table with that name exists	Run the query.	Qualify the table name with the schema, and run the query. If the table does not exist in that schema, return a “table not found” error.
A table that is not fully-qualified, and more than one table with that name exists	Return the following error: `Multiple matches of <table> found.`	Qualify the table name with the schema, and run the query. If the table does not exist in that schema, return the following error: `Table not found.`

Understanding database data type override options

Users with the DBA role can override how data types are reported to the client. If your preferred client does not work well with a particular option, you can change how the data type is reported so that the columns can still be read.

Example

- If multi-byte characters are stored in a database column defined as holding single-byte ascii characters, then you can use the set option type statement to override the data type in order for the data to display correctly when accessed from the query server.
- For certain numeric data types you can set the scale and precision of the number. For example, for a decimal data type, you could add a precision of 2 to represent US dollars.
- For CHAR and BINARY data types you can set the maximum length for the data type. For example, you can limit the CHAR field to 50 characters by adding a maximum length of 50.

Overriding options does not change the XSD or XML information, or how the query server interprets the XML data. After you override the data type, the type is reported differently, and, if possible, the XML data is converted to the desired type.

For data type overrides, the following java SQL data types are supported:

Data type	**Attributes you can set**
BINARY	• maxLen between 1 and 255
BIT	
CHAR	• maxLen between 1 and 254
DATE	
DOUBLE	
FLOAT	
INTEGER	
LONGVARBINARY	• maxLen between 1 and 2147483647
LONGVARCHAR	• maxLen between 1 and 2147483647
NUMERIC	• precision between 1 and 40 • scale between 0 and value of precision
REAL	
SMALLINT	
TIME	
TIMESTAMP	• scale values of 0, 3, 6, or 9
TINYINT	
VARBINARY	• maxLen between 1 and 4000
VARCHAR	• maxLen between 1 and 4000
WCHAR	• maxLen between 1 and 254
WLONGVARCHAR	• maxLen between 1 and 2147483647
WVARCHAR	• maxLen between 1 and 4000

Overriding database data types and server options

You can override data types and server options with the set option SQL statement.

1 Invoke SQL*Plus and log in as a user with privileges to access the xmlData data.

2 Override the data type or server option using the following syntax:

```
SET OPTION <option_type> <option_value>
```

Where	Is
option_type	The option you want to override, for example, TYPE.
option_value	The value you want to use for the option. Do not include any spaces or tabs in the option_value.

Examples

Statement	Expected result
`set option type clob=wlongvarchar`	All CLOB data types are converted to WLONGVARCHAR.
`set option type date=varchar, maxLen=12`	All DATE data types are converted to VARCHAR with a maximum length of 12 characters in the field.
`set option type var2=numeric, precision=5, scale=15`	All VAR2 data types are converted to NUMERIC with precision of 5 and scale of 15.
`set option type text=varchar, maxLen=64000`	All SQL Server TEXT data types are converted to VARCHAR with a maximum length of 64000.
`set option colmaxsize 2222`	Changes the default buffer size for all columns to 2222 bytes.
`set option ignorenull=true`	Ignores any null values in a column with character data without raising an error.

NOTE When using Interactive SQL (ODBC), the set option statements require a prefix of 'exec' or '!'. This is not required for Interactive SQL (JDBC) or Interactive SQL (OLE DB).

3 To revert back to the original default value, use the following syntax:

```
SET OPTION <option_type> <option_value>=default
```

Viewing existing database data type overrides

You can view existing data types and options that have been overridden by querying the OA_OPTIONS table.

1 Invoke SQL*Plus and log in as a user with privileges to access the xmlData data.

2 Use the following query to view existing overrides.

```
SELECT * FROM OA_OPTIONS;
```

The query displays the option, value and description columns for each override for the current server.

Column	Example value
`OPTION`	`type`
`VALUE`	`var2=numberic`
`DESCRIPTION`	`override var2 database types, treat as numeric`

NOTE If you need to verify what server you are using, you can use a query similar to: `select distinct table_qualifier from oa_tables where table_qualifier != 'SCHEMA'`

Viewing query server log files

Depending on the type of interaction, the query server logs are stored in different files.

Log file	Description
oaserver.log	Windows server installation log file.
oaclient.log	Windows client installation log file.
oaserveruninstall.log	Windows server uninstall log file.
oaclientuninstall.log	Windows client uninstall log file.
oaerror.log	Configuration or runtime error log for both Windows and UNIX.
obt.log	Query server interaction log for both Windows and UNIX.

All log files are located in the following directory:

```
<install_directory>/obt/log/
```

where <install_directory> is the directory where you installed the software.

See also Using log files (page 65)

A Configuration and runtime parameters

The following parameters govern the running of business flows and jobs:

- Database to file configuration parameters (page 121)
- File naming parameters (page 124)
- Runtime parameters (page 124)

Database to file configuration parameters

Selecting Database to file on the Parameters page displays the configuration parameters applied to all business flows and jobs that employ database to file extraction.

The configuration parameters are divided into the following sections:

- Core parameters (page 121)
- Performance parameters (page 122)
- Validation parameters (page 123)
- PDM parameters (page 123)

Core parameters

The following parameters are located in the Core section:

Parameter name	Description
Allow masked data on upload	By default, prevents any data that has been masked from being uploaded into the database. Set to true if you want to upload masked data instead of the original values.
Compression algorithm	Specifies the compression algorithm to apply to the XML and CSV files created after running the database to file extraction. Valid values are NONE and GZIP. • None—places the files in the specified directory without compression. • GZIP—compresses the files into GZIP format and places them in the specified directory.
Extract file format	Specifies whether to create XML denormalized or CSV normalized output.
Job engine SQL tracing enabled	Turns on SQL tracing for the job engine repository connection.

Parameter name	Description
Preserve temporary files	Saves the temporary files generated when running the job.
Primary key index location	Displays the location where the primary key indices will be stored.
Source database location	Specifies the name of the active database.
User index location	Displays the location where the user indexes will be stored.
Verify Row Counts	Set to True to perform verification of row counts between the current job and its corresponding selection job. Set to False to bypass verification. True is the default value.

Performance parameters

The following parameters are located in the performance section:

Parameter name	Description
Combined statement count	Defines the maximum number of database statements which can be combined in a single query.
Data movement Batch size	Defines the number of driving table rows per transaction. This is used for data movement operations that operate on related parent and child rows in the same transaction, which includes the following: • copies in the database to file transactional data movement option • all database to file operations. The total number of rows operated on can be larger that the value entered, and depends on the characteristics of the data.
Denormalize lookup records	Indicates whether to combine retrieval of multiple lookup table records with the referencing record in a single statement. Set to true or false.
Eligibility Analytics Configuration	By default, eligibility analytics is disabled. If you want to enable eligibility analytics for a business flow, you must enable it before you run the business flow. Enabling eligibility analytics allows querying of the analytics tables for information on record eligibility. • Select “Disable the eligibility analytics” to improve performance • Select “Enable the eligibility analytics” to enable eligibility analytics. NOTE If you select this option, you must include an interrupt step in your business flow to view the data.
Maximum number of parallel workers	Defines the default maximum number of job workers for tasks that can take advantage of parallelism.
Selection batch size	Defines the number of driving table rows per transaction. This is used for selection operations that select related parent and child row IDs from the source database into the selection tables.

Parameter name	Description
Type of indexes to create on Uploaded tables	Specifies how to populate index tables on upload.
Units of work	Defines the number of units amongst which the total amount of work will be divided. Each worker picks up a whole unit at a time to ensure clear progress indication and manage the total work in units for the job engine.

Validation parameters

The following parameters are located in the validation section:

Parameter name	Description
Check cardinality constraints	Indicates whether to validate that the extracted data does not violate cardinality constraints in the model instance definition.
Checksum algorithm	Indicates whether to run the checksum algorithm on created files.
Match rowcounts	Indicates whether to verify that rowcounts in the XML files match those in the database.
XML file checksum validation	Indicates whether to validate that the XML file checksums have not changed.
XML Schema validation	Indicates whether to validate that the XML files do not violate their XML schema. TIP For database to file copying, if you have large BLOB, CLOB, LONG, or LONG RAW fields, HP recommends leaving the value of the XML schema validation parameter set to false.

PDM parameters

The following parameter is located in the PDM section:

Parameter name	Description
Unify MTU Selections	Unify selections in multiple table uses (MTU) into one selection table, and remove duplicate rows. For database to file copying, the default value for this parameter is false, enabling multiple table users to retain the selected rows. There may be duplication of rows in the result. Set this parameter to true to remove duplicate rows.

File naming parameters

The following fine naming parameters are used only database to file extraction, and can be set for each individual cartridge:

Parameter	Description
Group File Prefix	The prefix for the group XML or CSV files. The default value is <cartridge_name>_, where cartridge_name is the name of the cartridge.
Group File Suffix	The suffix you want to use for the group XML or CSV files.
Group XML Schema File Prefix	The prefix for the group XML schema files. The default value is <cartridge_name>_, where cartridge_name is the name of the cartridge.
Group XML Schema File Suffix	The suffix you want to use for the group XML schema files.
Schema Mapping File for Upload	Specifies a schema mapping file to be applied on upload. For information about schema mapping, refer to Schema mapping for upload (page 52)
Summary File Prefix	The prefix for the summary files. The default value is <cartridge_name>_, where cartridge_name is the name of the cartridge.
Summary File Suffix	The suffix you want to use for the summary files.
Summary XML Schema File Prefix	The prefix for the summary XML schema files. The default value is <cartridge_name>_, where cartridge_name is the name of the cartridge.
Summary XML Schema File Suffix	The suffix you want to use for the summary XML schema files.

Runtime parameters

The following runtime parameters are used only for advanced selection and partitioned data movement:

Parameter	Description
START_ TABLE_ ALIAS	Type the unique table alias of the table for which you want to extract partitions, as specified by Designer. Defaults to null if missing or left blank. This parameter is only used with advanced selection or partitioned data movement.
START_ PARTITION_LIST	Type the comma-separated list of partitions to be extracted. The partitions must exist in the table specified in START_TABLE_ALIAS. No spaces can be used in the list. Defaults to null if missing or left blank. This parameter is only used with advanced selection or partitioned data movement.
RUN_ OPTION	Choose from the following list to define the logging detail: • RUN executes the job and reports the minimum amount of information. • LIST_SQL executes the job, provides the same logging detail as RUN, and also prints every SQL statement that is run. • SHOW_PLAN executes the job, provides the same logging detail as LIST_SQL, and also prints the execution plan for every SQL statement. • LIST_SQL_ONLY prints all the SQL statements that are generated. The SQL statements are not executed. • SHOW_PLAN_ONLY prints all the SQL statements and the execution plan. The SQL statements are not executed. Defaults to LIST_SQL if the RUN_OPTION parameter is missing or left blank. This parameter is only used with advanced selection or partitioned data movement.

NOTE The parameters are set on the Launch page for business flows that contain advanced selection.

B Advanced tasks

This chapter covers advanced tasks that are less frequently performed.

In this chapter

SQL tuning

Test Data Management uses SQL hints, session variables, and custom pre-statement execution code to allow you to configure the SQL statements in your Oracle or SQL Server installation, and improve the performance of your selection and data movement statements.

In this section

About SQL tuning

You can create hints, set session variables, or create custom pre-statement execution code that affect the selection and data movement statements for your installation.

Each modification is applied to a specific SQL statement for an individual cartridge in a business flow. For example, if the ORDERS_BF business flow contains two cartridges, ORDERS_DOM and ORDERS_INTL, you would have to write separate hints for each cartridge, or use wildcards.

Each cartridge contains the following seed statement files in the cartridge directory:

Seed statement files	Description
seed-selection-statements.xml	Contains SQL statements that apply to the cartridge selection process. Data selection statements are supported for database to file cartridges and database to file cartridges.
seed-dm-sql-statements.xml	Contains SQL statements that apply to the cartridge data movement process. Data movement statements are supported for database to file cartridges.

With wildcards, each modification can affect one or more statements in each seed statement file. The modifications are defined in the sql_tuning.properties file, which is stored in the following directory:

```
<install_directory>/obt/config
```

where <install_directory> is the location where you installed the software.

About statements

When you deploy database to file copying, the SQL statements in the seed-selection-statements.xml and seed-dm-sql-statements.xml files are generated.

Each statement contains the following information:

- source environment
- context
- appspack_name
- table_identifier
- statement_name

Example

```
<SRC_ENV>Oracle_OLTP</SRC_ENV>
<CONTEXT>OLTP_SELECTION</CONTEXT>
<APPSPACK_NAME>d2d_trans</APPSPACK_NAME>
<TABLE_IDENTIFIER>ORDER_HEADER</TABLE_IDENTIFIER>
<STATEMENT_NAME>INSERT_SELECTION_NO_ANALYTICS
</STATEMENT_NAME>
```

In this section

- About context values (page 129)
- About statement_name values (page 129)
- About appspack_name and table_identifier values (page 130)

About context values

The following context values are valid:

Context	Description
OLTP_SELECTION	The extract selection step.
ELIGIBILITY_ ANALYTICS	Eligibility analytics step.

About statement_name values

Each context has one or more of the following statement names.

In this section

- Selection statement names (page 129)
- Copy and move statement names (page 129)
- Eligibility analytics statement name (page 130)

Selection statement names

The following statement values are used when inserting into the selection tables:

Statement	Description
INSERT_SELECTION_ ANALYTICS	Used when the Eligibility Analytics Configuration configuration parameter in the Web Console is set to “Enable the eligibility analytics”. Eligibility Analytics must be enabled in both Designer and the Web Console before eligibility analytics occurs.
INSERT_SELECTION_ NO_ANALYTICS	Used when the Eligibility Analytics Configuration configuration parameter in the Web Console is set to “Disable the eligibility analytics”.

Copy and move statement names

The following statement values are used when moving data between the source database and the target. Each move operation consists of an INSERT statement.

Statement	Description
COPY_TRX_RANGE	Used for fully transactional data moment when RANGE is applicable.
COPY_TRX_NO_RAN GE	Used for fully transactional data moment when NO_RANGE is applicable.
COPY_TP_RANGE	Used for table parallel data moment when RANGE is applicable.

Statement	Description
COPY_TP_NO_RANGE	Used for table parallel data moment when NO_RANGE is applicable.
DELETE_RANGE	Used for deletion when RANGE is applicable.
DELETE_NO_RANGE	Used for deletion when NO_RANGE is applicable.

NO_RANGE will be executed when one of the following configuration combinations is set:

Configuration parameter	Required value
Use database parallelism to extract data	TABLE_PARALLEL
Number of rows per commit	0

or

Configuration parameter	Required value
Use database parallelism to extract data	FULLY_TRANSACTIONAL
Data movement batch size	0

If neither of these two combinations is set, RANGE will be used.

Eligibility analytics statement name

The following statement value is used for eligibility analytics:

Statement	Description
ELIGIBILITY_ANALYTICS	Used for eligibility analytics.

About appspack_name and table_identifier values

The appspack_name and table_identifier values depend on your cartridge and the tables included in that cartridge. The appspack_name corresponds to the cartridge name designated in Designer, and the table_identifier corresponds to the table alias designated in the model in Designer.

See the seed-selection-statements.xml and seed-dm-sql-statements.xml files for the valid values.

About using session variable statements

Session variable statements require different commands depending on the database you are using.

Database	Command
Oracle	Uses the ALTER SESSION command arguments.
SQL Server	Uses the SET command arguments.

Because Test Data Management automatically wraps it with the appropriate command (ALTER SESSION or SET), the variable value is just the arguments for the database command.

SQL Server example

For SQL Server, if you want to set the deadlock priority to low, and you would normally change the settings with:

```
SET deadlock_priority low
```

set the value to:

```
deadlock_priority_low
```

Oracle example

For Oracle, if you want to set optimizer index caching to 10, and you would normally change the optimizer settings with:

```
ALTER SESSION set optimizer_index_caching=10
```

set the value to:

```
set optimizer_index_caching=10
```

See also

Editing the sql_tuning.properties file (page 132)

About using custom pre-statement execution code

The PreExec code enables you to specify custom code to be executed before the selection or data movement statements.

Pre-statement execution code is executed in different locations depending on the type of statement.

Statement	Location
Selection statement	The PreExec code is executed on the source local database.
Copy statement	The PreExec code is executed on the target local database.

NOTE If the statement you are modifying is for a Reload job, the locations are reversed.

Example

```
Oracle_OLTP.OLTP_SELECTION.INVENTORY.ITEM.INSERT_SELECTIO
N_ANALYTICS.PREEXEC_NAME=apps_initialization
```

where `apps_initialization` is the name of a SQL block that you want to execute.

Editing the sql_tuning.properties file

Use the sql_tuning.properties file to add SQL hints or set session variables.

1. Navigate to the directory that contains the seed statements for the cartridge you want to modify.

Example

<install_directory>/obt/artifacts/businessflow/<business_flow_name>/cartridge/<product>/<cartridge_name>

Where	Is
<install_directory>	The location where you installed the software.
<business_flow_name>	The name of the business flow that contains the cartridge you want to modify.
<product>	The copy method the cartridge uses. • oa—database to file
<cartridge_name>	The individual cartridge you want to query.

NOTE Database to file cartridges only support selection statement hints.

2. Open the seed statement file you want to view.
   - seed-selection-statements.xml—contains selection statements
   - seed-dm-sql-statements.xml—contains data movement statements

NOTE Do not edit the seed statement files.

3. Search the seed statement file for the statement you want to modify.

   You need to note the following information:
   - source environment
   - context
   - appspack_name
   - table_identifier
   - statement_name

Example

```
<SRC_ENV>Oracle_OLTP</SRC_ENV>
<CONTEXT>OLTP_SELECTION</CONTEXT>
<APPSPACK_NAME>d2d_trans</APPSPACK_NAME>
<TABLE_IDENTIFIER>ORDER_HEADER</TABLE_IDENTIFIER>
```

```
<STATEMENT_NAME>INSERT_SELECTION_NO_ANALYTICS
</STATEMENT_NAME>
```

4 Open the following file in a text editor:

```
<install_directory>/obt/config/sql_tuning.properties
```

where <install_directory> is the location where you installed the software.

NOTE For Oracle, the sql_tuning.properties file has default hints. The SQL Server sql_tuning.properties file has no defaults.

To add a hint

5 Add a new hint using the following format:

```
<SRC_ENV>.<CONTEXT>.<APPSPACK_NAME>.<TABLE_IDENTIFIER>.
<STATEMENT_ NAME>.<hintType>=<hint>
```

Where	Is
<SRC_ENV>	The contents of the <SRC_ENV> tag in the seed statement file.
<CONTEXT>	The contents of the <CONTEXT> tag in the seed statement file.
<APPSPACK_NAME>	The contents of the <APPSPACK_NAME> tag in the seed statement file.
<TABLE_IDENTIFIER>	The contents of the <TABLE_IDENTIFIER> tag in the seed statement file.
<STATEMENT_NAME>	The contents of the <STATEMENT_NAME> tag in the seed statement file.
<hintType>	INSERT_HINT, SELECT_HINT or DELETE_HINT.
<hint>	The SQL hint.

Example hint

```
OLTP_SELECTION.d2d_trans.ORDER_HEADER.INSERT_SELECTION_NO
_ANALYTICS.<hintType>=<hint>
```

TIP You can use wildcards to apply the changes to more than one statement. `*.*.*.*.<hintType>=<hint>` applies the hint to all statements in all installed cartridges.

Example SQL in seed file

A SQL statement in a seed file looks something like the following sample. The items between the ## symbols are the available hintTypes for that statement.

```
INSERT ##INSERT_HINT##  INTO "CUSTOLTP_HIST"."ORD"
("ORDERDATE", "ORDID", "OBT_WF_RUN_ID", "OBT_ROW_SEQ",
"OBT_TIMESTAMP", "OBT_SAVED_ROWID" ) SELECT
##SELECT_HINT##    "S"."ORDERDATE",
    "S"."ORDID",
    "S"."OBT_WF_RUN_ID",
    "S"."OBT_ROW_SEQ",
    sysdate,
    "S"."OBT_SAVED_ROWID"
FROM    "OBT_IF"."ORD_SVO"@ORCL3 "S"
WHERE   ("S"."OBT_WF_RUN_ID"=?)
AND ("S"."OBT_ROW_SEQ" BETWEEN ? AND ?)
```

6 Fill in the appropriate SQL hint.

Oracle example

```
Oracle_OLTP.OLTP_SELECTION.d2d_trans.ORDER_HEADER.INSERT_
SELECTION_NO_ANALYTICS.SELECT_HINT=use_nl
```

SQL Server example

```
MSOLTP.OLTP_SELECTION.d2d_trans.ORDER_HEADER.INSERT_SELEC
TION_NO_ANALYTICS.SELECT_HINT=FAST 5
```

7 Save the sql_tuning.properties file.

To add a session variable

8 Add a new session variable using the following format:

```
<SRC_ENV>.<CONTEXT>.<APPSPACK_NAME>.<TABLE_IDENTIFIER>.<S
TATEMENT_ NAME>.<SESSION_VARn>=<var>
```

Where	Is
<SRC_ENV>	The contents of the <SRC_ENV> tag in the seed statement file.
<CONTEXT>	The contents of the <CONTEXT> tag in the seed statement file.
<APPSPACK_NAME>	The contents of the <APPSPACK_NAME> tag in the seed statement file.
<TABLE_IDENTIFIER>	The contents of the <TABLE_IDENTIFIER> tag in the seed statement file.
<STATEMENT_NAME>	The contents of the <STATEMENT_NAME> tag in the seed statement file.
<SESSION_VARn>	The identifier for the session variable.
<var>	The session variable.

Example

```
Oracle_OLTP.OLTP_SELECTION.d2d_trans.ORDER_HEADER.INSERT_
SELECTION_NO_ANALYTICS.<SESSION_VARn>=<var>
```

TIP You can use wildcards to apply the changes to more than one statement. `*.*.*.*.<SESSION_VARn>=<var>` applies the session variable to all statements in all installed cartridges.
For example, if the default statement is as follows:
`DefEnv.OLTP_SELECTION.*.*.*.INSERT_HINT=APPEND`
changing the statement to:
`DefEnv.OLTP_SELECTION.INVENTORY.ITEM.*.INSERT_HINT=PARALLEL`
will override the APPEND hint for only the ITEM table in INVENTORY.

9 Fill in the appropriate session variable.

Oracle example

```
Oracle_OLTP.OLTP_SELECTION.d2d_trans.ORDER_HEADER.INSERT_
SELECTION_NO_ANALYTICS.SESSION_VAR3=set
optimizer_index_caching=10
```

SQL Server example

```
MSOLTP.OLTP_SELECTION.d2d_trans.ORDER_HEADER.INSERT_SELEC
TION_NO_ANALYTICS.SESSION_VAR3=deadlock_priority low
```

10 Save the sql_tuning.properties file.

To add a preexec statement

11 Add a new pre-statement execution code using the following format:

```
<SRC_ENV>.<CONTEXT>.<APPSPACK_NAME>.<TABLE_IDENTIFIER>.
<STATEMENT_NAME>.PREEXECNAME=<preexecname>
```

Where	Is
<SRC_ENV>	The contents of the <SRC_ENV> tag in the seed statement file.
<CONTEXT>	The contents of the <CONTEXT> tag in the seed statement file.
<APPSPACK_NAME>	The contents of the <APPSPACK_NAME> tag in the seed statement file.
<TABLE_IDENTIFIER>	The contents of the <TABLE_IDENTIFIER> tag in the seed statement file.
<STATEMENT_NAME>	The contents of the <STATEMENT_NAME> tag in the seed statement file.
<preexecname>	The name of the custom code you want to execute.

Example

```
Oracle_OLTP.OLTP_SELECTION.INVENTORY.ITEM.INSERT_SELECTIO
N_ANALYTICS.PREEXEC_NAME=apps_initialization
```

where `apps_initialization` is the name of the custom code you want to execute.

TIP You can use wildcards to apply the changes to more than one statement. `*.*.*.*.<PREEXEC_NAME>=<preexecname>` applies the modification to all statements in all installed cartridges.

12 Save the sql_tuning.properties file.

Using views for customized reporting

HP Test Data Management supports the following views of the product metadata:

View	Definition
OBTWC_BF_RUNS_V	Displays business flow runs, but does not display the detailed steps.
OBTWC_RUN_DETAILS_V	Displays business flow runs and the detailed steps of the business flow.
OBTWC_RUN_PARAMETERS_V	Displays the name & value of each parameter and maps them to an individual job run_id.

You can use these views to create customized reports for your installation.

Setting up email access from the Web Console

Before you can send email using the Web Console, you need to configure the webConsole.properties file.

1. Navigate to the directory that contains the webConsole.properties file.

   Example

   ```
   cd <install_directory>/obt/config/
   ```

   where <install_directory> is the location where you installed the software.

2. Open the webConsole.properties file with a text editor.

   TIP To ensure that any absolute links (for example, for a lost password reset) work correctly, you should set `grails.serverURL` to your application server. For example, `grails.serverURL= http://localhost:8080` or `grails.serverURL=http://tiger.internal.outerbay.com:8080`.

3. Edit the mail properties as appropriate for your environment.
4. Remove the comment marks for the mail properties you edited.
5. Save the webConsole.properties file.
6. Start the Web Console service.
7. Open the following web page:

   ```
   http://localhost:8080/WebConsole/main/mailTest
   ```

8. Test the email settings.

Enabling SQL trace for Oracle

You can enable or disable Oracle SQL trace using properties files. Because SQL trace can impact performance, HP recommends that you only enable it for performance tuning.

- To enable SQL tracing for the database to file selection, use the sql_tuning.properties file.
- To enable SQL tracing for database to file data movement, use the outerbay.properties file.

The trace files are saved to the user_dump_dest directory. For more information, see your Oracle documentation.

Enabling SQL trace using the sql_tuning.properties file

Use the sql_tuning.properties file to enable SQL trace for database to file selection and data movement.

1 Open the sql_tuning.properties file.

See also SQL tuning (page 127)

2 Add a session variable to enable SQL trace.

Example

```
SESSION_VAR1=set events '10053 trace name context
forever, level 1'
```

The trace begins before the statement that the session variable is attached to.

For example, if you attach it to the following statement:

```
OLTP_SELECTION.INVENTORY.ITEM.INSERT_SELECTION_ANALYTICS.
SESSION_VAR1=set events '10053 trace name context
forever, level 1'
```

the trace command is issued before the `OLTP_SELECTION.INVENTORY.ITEM.INSERT_SELECTION_ANALYTICS` SQL statement is run.

Enabling SQL trace using the outerbay.properties file

Use the outerbay.properties file to enable SQL trace for database to file data movement.

1 Navigate to the directory that contains the outerbay.properties file.

Example

```
<install_directory>/obt/config
```

where <install_directory> is the location where you installed the software.

2 Open the outerbay.properties file with a text editor.

3 Locate the SQL trace section.

4 Update the properties file. By default, the following properties are set:

```
DASL.sqlTrace.level=1
DASL.sqlTrace.components=ALL
```

5 Save the outerbay.properties file.

C Scripted product setup and business flow deployment

If you plan to set up and deploy HP Test Data Management multiple times, you can use scripted deployment. In this section

- Understanding properties files (page 139)
- Scripting the repository setup (page 140)
- Scripting the environment creation (page 143)
- Scripting business flow deployment (page 146)

See also Chapter 3 of the *HP Test Data Management Installation guide* for information about scripted installation of the product.

Understanding properties files

Scripted deployment requires that you create the following properties files to define the installation.

Property file	Description
repository.install.properties	Contains properties to create your repository.
product.install.properties	Contains properties to create environments with database to file
businessflow.install.properties	Contains properties to deploy your business flows.

You can create these files in one of two ways:

- Use the Web Console to generate the properties files.
- Use a text editor to manually create the files.

The instructions in this chapter assume that you have used the Web Console at least once to create the properties files in the install directory.

NOTE Properties files should contain property values with ISO 8859-1 character encoding only. For more information on using characters that cannot be expressed using ISO 8859-1 encoding, see Using special characters in properties files (page 140).

Using special characters in properties files

Properties files should contain property values with ISO 8859-1 character encoding only. Unicode escapes can be used for characters that cannot be directly represented in this encoding. One u character can be used per escape sequence.

To convert properties files that contain non-ISO 8859-1 character encoding:

1 Run the Native-to-ASCII Converter, native2ascii, with the following syntax:

```
native2ascii -encoding <encoding_name> <input_file>
<output_file>
```

Where	Is
<encoding_name>	A valid Java encoding. For more information, see the Sun Java website at http://java.sun.com.
<input_file>	The original properties file with non-ISO 8859-1 character encoding.
<output_file>	The new properties file with only ISO 8859-1 character encoding.

2 Use the new properties file for your scripted installation.

Scripting the repository setup

To script your repository installation, you need to create a file with the properties you want to use, and then run the script. The following sections contain the scripting procedure, the properties in the properties file, and examples of the properties file.

NOTE If you are extracting data from Sybase, you must install your repository on either an Oracle or SQL Server database, or use the embedded repository.

In this section

- Scripted repository installation procedure (page 140)
- Scripted repository installation properties (page 141)
- Scripted repository installation properties file example (page 142)

Scripted repository installation procedure

If you choose to include passwords or encryption.key properties in the properties file, they are saved in the text file and can be accessed. To ensure that the passwords are not saved, omit the entire password property line from the properties file. You will then be prompted for the passwords.

1 Ensure that HP Test Data Management has been installed.

See also *HP Test Data Management Installation guide*

2 Navigate to the directory that contains the installation scripts:

Example

```
<install_directory>/obt/install
```

where <install_directory> is the location you installed the software.

3 Open the repository.install.properties file in a text editor, or create the file if it does not exist.

4 Edit the properties file as necessary for your installation.

See also Scripted repository installation properties (page 141)

5 Open a command window.

6 Install the repository using the appropriate syntax:

For	Syntax
UNIX	`./deploy_repository.sh <properties_file_name>`
DOS	`deploy_repository.bat <properties_file_name>`

Where `properties_file_name` is the name of the properties file you are using for the repository.

Scripted repository installation properties

The following properties are used for installing the repository:

property name	Expected value
encryption.key	The encryption key to be created. If you omit the line, you will be prompted for the encryption key.
repository.conn.rdbmstype	sqlserver or oracle. For the embedded repository, the value should be derbydb_10.
repository.conn.dbserver	Database SID or server name for the repository.
repository.conn.host	Host name for the repository.
repository.conn.port	Port number for the repository.
repository.dbadmin.username	Repository database or database server administrator username. (Oracle, SQL Server only)
repository.dbadmin.password	Repository database or database server administrator password. (Oracle, SQL Server only) If you omit the line, you will be prompted for the password.
repository.owner.name	Repository user to be created by the installer.
repository.owner.password	Repository user password to be created. If you omit the line, you will be prompted for the password.

property name	Expected value
repository.default.storagelocation	Default tablespace to use when creating repository users. (Oracle only)
repository.temp.storagelocation	Temporary tablespace to use when creating repository users. (Oracle only)
repository.dbname	Repository database name. (SQL Server, embedded repository only)
repository.default.storagelocation.size	Data device size to be allowed to the repository database in MB. (SQL Server, embedded repository only)
repository.lot.storagelocation.size	Log device size to be allocated to the repository database in MB. (SQLServer, embedded repository only)
console.admin.username	The username for the Web Console administrator.
console.admin.passwd	The password for the Web Console administrator.
console.admin.email	Optional. The email address of the Web Console administrator.
console.admin.userRealName	Optional. The real name of the Web Console administrator.
console.admin.phone	Optional. The phone number of the Web Console administrator.
console.admin.description	Optional. The description of the Web Console administrator.

Scripted repository installation properties file example

The following examples demonstrate the use of the properties file.

Oracle

```
encryption.key=myEncryptionKey
repository.conn.rdbmstype=oracle
repository.conn.dbserver=orcl1
repository.conn.host=myHost
repository.conn.port=1521
repository.dbadmin.username=system
repository.dbadmin.password=myPassword
repository.owner.name=obt_rep
repository.owner.password=myPassword
repository.default.storagelocation=USERS
repository.temp.storagelocation=TEMP
console.admin.username=admin
console.admin.passwd=admin
```

SQL Server

```
encryption.key=myEncryptionKey
repository.conn.rdbmstype=sqlserver
repository.conn.dbserver=MSOLTP
repository.conn.host=myHost
repository.conn.port=1433
repository.dbadmin.username=sa
repository.dbadmin.password=myPassword
repository.owner.name=obt_rep
repository.owner.password=myPassword
repository.dbname=obt_rep
repository.default.storagelocation.size=50
repository.log.storagelocation.size=50
```

```
console.admin.username=admin
console.admin.passwd=admin
```

Embedded repository

```
encryption.key=myEncryptionKey
repository.conn.conntype=default
repository.conn.rdbmstype=derbydb
repository.conn.dbserver=obt_rep
repository.conn.host=myHost
repository.conn.port=1527
repository.owner.name=obt_rep
repository.owner.password=myPassword
repository.dbname=obt_rep
console.admin.username=admin
console.admin.passwd=admin
```

Scripting the environment creation

To script your environment creation, you need to create a file with the properties you want to use, and then run the script. The following sections contain the scripting procedure, the properties in the properties file, and examples of the properties file.

Each environment requires a separate product.install.properties file.

In this section

- Scripted environment creation procedure (page 143)
- Scripted environment creation properties (page 144)
- Scripted environment creation properties file example (page 145)

Scripted environment creation procedure

The sample product.install.properties file is installed with HP Test Data Management.

If you choose to include passwords or encryption.key properties in the properties file, they are saved in the text file and can be accessed. To ensure that the passwords are not saved, omit the entire password property line from the properties file. You will then be prompted for the passwords.

NOTE Database to file extraction is installed with all environments.

1 Ensure that HP Test Data Management has been installed.

See also *HP Test Data Management Installation guide*

2 Navigate to the directory that contains the installation scripts:

Example

```
<install_directory>/obt/install
```

where <install_directory> is the location you installed the software.

3 Open the product.configure.properties file in a text editor.

4 Edit the properties file as necessary for your installation.

See also Scripted repository installation properties (page 141)

5 Open a command window.

6 Create the environment and deploy the product using the appropriate syntax:

For	Syntax
UNIX	`./deploy_product.sh <properties_file_name>`
DOS	`deploy_product.bat <properties_file_name>`

Where `properties_file_name` is the name of the properties file you are using.

Scripted environment creation properties

The following properties are used for creating the environment and deploying database to file:

Property name	Expected value
encryption.key	Encryption key. If you omit the line, you will be prompted for the encryption key.
environment.id	The name you want to use for the environment.
environment.description	The optional description of the environment.
source.rdbms.name	Acceptable values are oracle, sqlserver, or sybase.
source.conn.dbserver	The name of the Oracle database service, SQL Server or Sybase server name, or data source for the source database.
source.conn.host	The host for the source database.
source.conn.port	The port number for the source database.
source.dbadmin.username	The source database or database server administrator username.
source.dbadmin.password	The source database or database server administrator password. If you omit the line, you will be prompted for the password.
source.interface.dbname	The interface database name. (SQL Server, Sybase only)
source.interface.default.storagelocation.size	The data device size to be allocated to the interface database in MB. (SQL Server only)
source.interface.log.storagelocation.size	The log device size to be allocated to the interface database in MB. (SQL Server only)
source.interface.default.storagelocation	The location for the data device. (Sybase only)
source.interface.log.storagelocation	The location for the log device. (Sybase only)

Property name	Expected value
source.interface.owner.name	Interface user to be created by the installer.
source.interface.owner.password	Interface user password. If you omit the line, you will be prompted for the password.
source.interface.default.storagelocation	The default tablespace to use when creating the interface user for the source database. (Oracle only)
source.interface.temp.storagelocation	The temporary tablespace to use when creating the interface user for the source database. (Oracle only)
user.index.location	The location for the user index. Possible values are source or repository.

Scripted environment creation properties file example

The following examples demonstrate the use of the properties file.

SQL Server example

SQL Server with database to file only:

```
#Product Information
encryption.key=myEncryptionKey
environment.id=myEnvironment
environment.description=My Default Environment

#Source Database Information
source.rdbms.name=sqlserver
source.conn.dbserver=MSOLTP
source.conn.host=myHost
source.conn.port=5001
source.dbadmin.username=sa
source.dbadmin.password=myPassword

#Interface User Information
source.interface.dbname=obt_if
source.interface.owner.name=obt_if
source.interface.owner.password=myPassword
source.interface.default.storagelocation.size=50
source.interface.log.storagelocation.size=25
```

Sybase example

Sybase with database to file only:

```
#Product Information
encryption.key=myEncryptionKey
environment.id=MySybaseEnv
environment.description=Sybase environment

#Repository Information
rdbms.name=derbydb
repository.dbname=obt_rep
repository.conn.rdbmstype=derbydb_10
repository.conn.dbserver=xxx
repository.conn.host=localhost
repository.conn.port=1527
repository.dbadmin.username=obt_rep
repository.owner.name=obt_rep

#Source Database Information
source.rdbms.name=sybase
source.conn.dbserver=SYBASE_OLTP125
source.conn.host=myHost
source.conn.port=5000
source.dbadmin.username=sa
source.interface.owner.name=obt_if
source.interface.dbname=obt_if
source.interface.default.storagelocation=my_device
source.interface.log.storagelocation=my_log_device
```

Scripting business flow deployment

To script your business flow deployment, you need to create a file with the properties you want to use, and then run the script. The following sections contain the scripting procedure, the properties that can be used, and an example of the properties file.

In this section

- Scripted business flow deployment procedure (page 146)
- Scripted business flow deployment properties (page 148)
- Scripted business flow deployment properties file example (page 148)

Scripted business flow deployment procedure

1 Ensure the repository and environment has been created.

NOTE If you plan to use eligibility analytics, you must enable eligibility analytics in the model for your business flow before deploying, and the model must contain a pause after the selection step. For more information, see the *HP Test Data Management Developer's guide*.

2 Navigate to the directory that contains the generated business flows and the businessflow properties files.

Business flows located in the main businessflow directory can be accessed by all environments.

Example

```
<install_directory>/obt/businessflow
```

where <install_directory> is the directory where you installed the software.

Business flows located in an environment directory can only be accessed by that environments.

Example

```
<install_directory>/obt/businessflow/<environment_name>
```

where <install_directory> is the directory where you installed the software, and <environment_name> is the name of the environment.

3 Ensure that the business flow properties file has the same name as the generated business flow that you want to deploy.

Example If the full name of the .businessflow file in the business flow directory is ORDERS_D2F_BF.1_0_0_0.busflow, the properties file should be called ORDERS_D2F_BF.1_0_0_0.properties.

4 Edit the properties file with the appropriate property values from Scripted business flow deployment properties (page 148).

If you choose to include passwords or encryption.key properties in the properties file, they are saved in the text file and can be accessed. To ensure that the passwords are not saved, omit the entire password property line from the properties file. You will then be prompted for the passwords.

5 Save the file.

6 Open a command window.

7 Navigate to the following directory:

```
<install_directory>/obt/install
```

where <install_directory> is the directory where you installed the software.

8 Deploy the business flow using the appropriate command:

For	Syntax
UNIX	`./deploy_businessflow.sh <environment_name> <businessflow_full_name>`
DOS	`deploy_businessflow.bat <environment_name> <businessflow_full_name>`

where <businessflow_full_name> is the full name of the business flow you want to deploy, and <environment_name> is the name of the environment.

Example

```
deploy_businessflow.bat MyEnvironment ORDERS_D2F_BF.1_0_0_0
```

TIP If you want to remove the job run history of any existing business flows, add the following command: `-Dforce.job.history.drop=true`

Scripted business flow deployment properties

Name	Values
environment.id	Environment ID
encryption.key	Encryption key. If you omit the line, you will be prompted for the encryption key.
businessflow.install.configuration	Possible values are: • PrepareOnly • DeployOnly • PrepareAndDeploy
rdbms.name	The rdbms.name of the source database. Acceptable values are sqlserver, oracle, sybase, or neoview.
source.dbadmin.username	The source database administrative username.
source.dbadmin.password	The password for the source database administrative user. If you omit the line, you will be prompted for the password.
source.location.name	The location of the source database. Required for business flows containing database to file cartridges.

Scripted business flow deployment properties file example

The following examples demonstrate the use of the properties file.

Oracle example

Oracle database to file business flow

```
#Business Flow Install Properties
businessflow.install.configuration=PrepareAndDeploy
environment.id=MyEnv
rdbms.name=oracle
source.dbadmin.username=system
source.location.name=OBTINTF_DB
```

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 CVS 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