QAD Enterprise Applications

Installation Guide

QAD Data Warehouse Designer

78-0935-6.6.2
Data Warehouse Designer Version 6.6.2
September 2012

DataWarehouseDesigner_IG_v0662.pdf/r9m

QAD Inc.
100 Innovation Place
Santa Barbara, California 93108
Phone (805) 566-6000
http://www.qad.com

Contents

Change Summary

The following table summarizes significant differences between this document and the previous version.

Date/Version	Description	Reference
September 2012/6.6.2	New version of DWD; changes throughout	
March 2012/6.5.6	Minor updates throughout	
	Expanded List/Load Application chapter with additional metadata options	
September 2011/QAD BI 3.5	Rebranded for QAD BI 3.5	--

CHAPTER 1

Installation

This manual guides you through the installation and setup of the QAD Data Warehouse Designer product. It is also used as a reference for the creation of additional metadata repositories.

- The installation of the product onto a PC.
- An introduction to the Setup Administrator utility.
- The setup of the PC environment to support QAD Data Warehouse Designer.
- The Installation of the metadata repository
- The installation of the tutorial data.
- The installation of the QAD Data Warehouse Designer Scheduler.
- The installation of any third party data warehouse applications.
- The quick installation and setup of a Data Warehouse Designer data warehouse
- The quick installation and setup of a Data Warehouse Designer data warehouse and application.
- The creating and loading of language files.

In This Chapter

Documentation Roadmap

The QAD Data Warehouse Designer Installation and Administration Guide, the online help, and the QAD Data Warehouse Designer User Guide assume that you are proficient in the use of the Windows operating system.

QAD Data Warehouse Designer often provides multiple ways to accomplish a task. In some cases, you can use the main menu, the right-mouse button menu, or a toolbar, or a key combination (e.g. Alt/M and Ctrl/M to raise menus). Instructions in this documentation generally include only the most convenient method of accomplishing a task.

The following sources of information are available with QAD Data Warehouse Designer:

QAD Data Warehouse Designer Installation and Administration Guide

The Installation and Administration Guide is available either as online help, as a PDF, or in printed format. The Installation and Administration Guide provides the information needed to:

- Install the QAD Data Warehouse Designer software
- Validate the various software components required by QAD Data Warehouse Designer
- Install the QAD Data Warehouse Designer metadata
- Install a Scheduler
- Optionally install third-party data warehouse applications.

QAD Data Warehouse Designer User Guide

The QAD Data Warehouse Designer User Guide is available either as online help, as a PDF, or in a printed manual. The User Guide provides information on how to use QAD Data Warehouse Designer to build a data warehouse. It includes a set of tutorials to assist in the familiarization of the terms and methodologies embodied in QAD Data Warehouse Designer.

QAD Data Warehouse Designer Teradata User Guide

The QAD Data Warehouse Designer Teradata User Guide is available either as online help, as a PDF, or in a printed manual. The Teradata User Guide provides information on how to use QAD Data Warehouse Designer to build a data warehouse in Teradata. It includes a set of tutorials to assist in the familiarization of the terms and methodologies embodied in QAD Data Warehouse Designer.

Sql Admin User Guide

The Sql Admin User Guide is available either as online help, or as a PDF. It provides documentation in the use of the stand alone SQL query tool shipped as part of the QAD Data Warehouse Designer product.

QAD Forum

A web forum is available at http://www.qad.com. This forum contains information on the latest version, and bug reports that may be relevant for installation. In addition the QAD Blog is available at http://www.qad.com which may provide additional information.

Installation RoadMap

QAD Data Warehouse Designer is designed to build data warehouse analysis areas within a database environment. The installation and configuration process is very rarely a simple task as this involves a number of layered products external to QAD Data Warehouse Designer. For the more complex cases a number of hours or days should be set aside to complete this task. Therefore, it is important that the reader begins the installation process with the correct expectations. Highlighted below are the major steps in the installation and setup of QAD Data Warehouse Designer. In many environments many of these steps will have been completed independently of the installation process, or may need to be completed by a third party.

Note: This document does not cover the creation, and configuration of the database required to run the data warehouse. It does not cover the procedures required to create and maintain database users. Nor does it cover the installation of the database client software. Refer to the database vendor manuals for assistance.

This manual provides a logical path for the installation and setup of QAD Data Warehouse Designer. Work through each chapter in order to complete the installation.

The steps provided here are an indication of the normal process required to install and configure QAD Data Warehouse Designer. In some environments additional processes will be required.

SQL Server Data Warehouse

1. Designate a database that is to be used as the data warehouse database.
2. Install the QAD Data Warehouse Designer software onto a PC.
3. Add the definition of an ODBC data source to connect to the data warehouse database.
4. Create the QAD metadata repository. This procedure is performed in the metadata section of the Setup Administrator utility.
5. If required load the QAD tutorial tables into a separate database. This step is performed via the Setup Administrator utility. It is highly recommended that the tutorials be undertaken.
6. Install a QAD Scheduler as defined in the chapter in this manual.

Note: A Quickstart option is available for SQL Server. This option will perform all the required tasks (except license key) when provided a valid database server.

Oracle Data Warehouse

1. Install the Oracle 8.1.7, 9i, 10g or 11g database client software on a suitable PC.
2. Designate a database that is to be used as the data warehouse database. For testing or evaluation purposes this may be a PC based database. QAD Data Warehouse Designer requires the Oracle database version 8.1.7 or greater as the data warehouse database. Record the SID for this database.
3. Establish the Oracle Net connections required to allow access to the data warehouse database from the PC. Record the 'Server' name for the connection.
4. Install the QAD Data Warehouse Designer software on the same PC.
5. Step through the tasks in the PC Setup section of the QAD Data Warehouse Designer Setup Administrator utility. Important specific tasks are the entering of the license key, the confirmation or installation of the 'Microsoft ODBC for Oracle' driver, and the definition of an ODBC data source to connect to the data warehouse database.
6. Create the database user (schema) that is to be used to hold the metadata and data warehouse tables. Allocate the privileges required by QAD Data Warehouse Designer.
7. Create the QAD metadata repository under the data warehouse user created in step (6). This procedure is performed in the metadata section of the Setup Administrator utility.
8. If required, load the QAD tutorial tables into a suitable database schema. This step is performed via the Setup Administrator utility. It is highly recommended that the tutorials be undertaken.
9. Install a QAD Scheduler as defined in the chapter in this manual.

IBM DB2 Data Warehouse

1. Install the IBM DB2 9.5, 9.7 or 10.1 client access software on a suitable PC.
2. Designate a database that is to be used as the data warehouse database. For testing or evaluation purposes this may be a PC based database. QAD Data Warehouse Designer requires the DB2 database version 9.5 or greater as the data warehouse database.
3. Install the QAD Data Warehouse Designer software on the same PC.
4. Step through the tasks in the PC Setup section of the QAD Data Warehouse Designer Setup Administrator utility. Important specific tasks are the entering of the license key and the definition of an ODBC data source to connect to the data warehouse database.
5. Create the database schema that is to be used to hold the metadata and data warehouse tables. Allocate the privileges required by QAD Data Warehouse Designer.
6. Create the QAD metadata repository under the data warehouse schema created in step (5). This procedure is performed in the metadata section of the Setup Administrator utility.
7. If required load the QAD tutorial tables into a suitable database schema. This step is performed via the Setup Administrator utility. It is highly recommended that the tutorials be undertaken.
8. Install a QAD Scheduler as defined in the chapter in this manual.

Teradata Data Warehouse

1. Install Teradata Tools and Utilities 12 (or later) on a suitable PC. These tools and utility components are required:

   - CLI

   - Teradata ODBC Driver

   - BTEQ

   - Fast Load

   - Multiload

   - Fast Export

   Optionally install TPT (with LOAD and UPDATE operators) if you would like to use TPT file loads or TPT ODBC loads.

2. Designate a database or user that is to be used as the meta data database. For testing or evaluation purposes this database or user may be a Teradata Express database server. QAD Data Warehouse Designer requires the Teradata database version 12 or greater as the data warehouse database server. Record the database and server names for this database.

   Note: A Teradata user is preferred (over a database) as the metadata includes some procedures that use dynamic sql. Procedures containing dynamic sql may only be compiled under a user in Teradata. In the scheduler, these procedures are used for several features including refreshing statistics. If they fail to compile, these features will not be available.

3. Add appropriate entries to the hosts file on the PC for the server.

4. Install the QAD Data Warehouse Designer software on the same PC.

5. Step through the tasks in the PC Setup section of the QAD Data Warehouse Designer Setup Administrator utility. Important specific tasks are the entering of the license key and the definition of an ODBC data source to connect to the data warehouse database server.

6. Create the database or user that is to be used to hold the metadata tables. Allocate the privileges required by QAD Data Warehouse Designer.

7. Create the QAD metadata repository under the data warehouse database or user created in step (6). This procedure is performed in the metadata section of the Setup Administrator utility.

8. If required load the QAD tutorial tables into another database or user. This step is performed via the Setup Administrator utility. It is highly recommended that the tutorials be undertaken.

9. Install a QAD Scheduler as defined in the chapter in this manual.

Installation Prerequisites

QAD Data Warehouse Designer requires certain client and server components.

Client Prerequisites

- Windows 2003, Windows XP SP2, Windows 2008, Windows 2008 R2, Windows Vista or Windows 7.
- 100MB available disk space
- ODBC Version 3.0 (normally shipped as part of the Operating System).
- For building an Oracle data warehouse a version 9.2 database client or greater is required. A 32-bit Oracle client installation will be necessary.
- For building a SQL Server data warehouse the appropriate Microsoft ODBC driver for SQL Server is required.
- For building a DB2 data warehouse a version 9.5 database client or greater is required.
- For building a Teradata data warehouse Teradata Tools and Utilities 12 or greater is required.

Note: QAD Data Warehouse Designer only supports the Microsoft ODBC for Oracle. See ***Check/Add Odbc Entry(s)*** (see "***Check / Add Odbc Entry(s)***" on page 22)

Server Prerequisites

One of the following data warehouse database platforms is required:

- Oracle version 9.2, 10g or 11g
- Microsoft SQL Server 2005, 2008, 2008 R2 or 2012
- IBM DB2 9.5.x, 9.7.x or 10.1.x
- Teradata 12, 13.x or 14.x

To use the QAD Data Warehouse Designer Scheduler, a UNIX, LINUX or Windows based server is also required (for SQL Server, Oracle and DB2 this may be the database server). QAD Data Warehouse Designer only supports sh and ksh under Unix and bash under Linux.


TIP: When building a Teradata data warehouse, the Teradata database server must have a c++ compiler installed. This is used by Teradata to compile stored procedures.

Oracle and 64-bit Windows

The QAD Data Warehouse Designer installer selects the following installation directory as the default location for Data Warehouse Designer on a 64-bit windows server:

- C:\Program Files (x86)\QAD

This is where 32-bit applications normally reside on a 64-bit Windows platform.

However, if Oracle 10g is involved, either as the data warehouse repository, or as a source system, installing Data Warehouse Designer in “Program Files (x86)” can cause an issue:

Oracle does not like directories that have a parenthesis in their name. Applications using Oracle cannot be located in such a directory. This includes QAD Data Warehouse Designer using ODBC to connect to Oracle. The result is that QAD Data Warehouse Designer will not be able to connect to Oracle.

This issue does not usually occur with Oracle 11g.

To prevent this, change the installation directory of QAD Data Warehouse Designer. Options include:

- C:\QAD
- D:\Program Files\QAD (if a second drive is available)

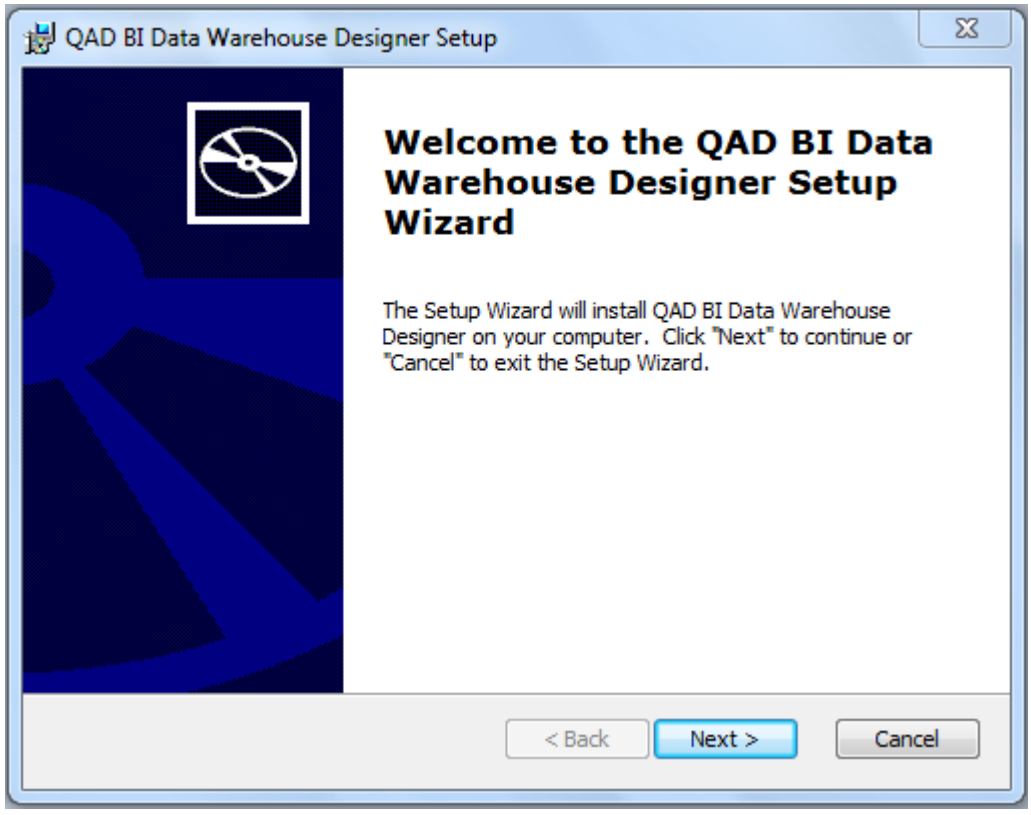
Note: It is not good practice to install 32-bit applications in C:\Program Files\QAD on a 64-bit machine.

Installing QAD Data Warehouse Designer

Use the following procedure to install the QAD Data Warehouse Designer software onto a PC.

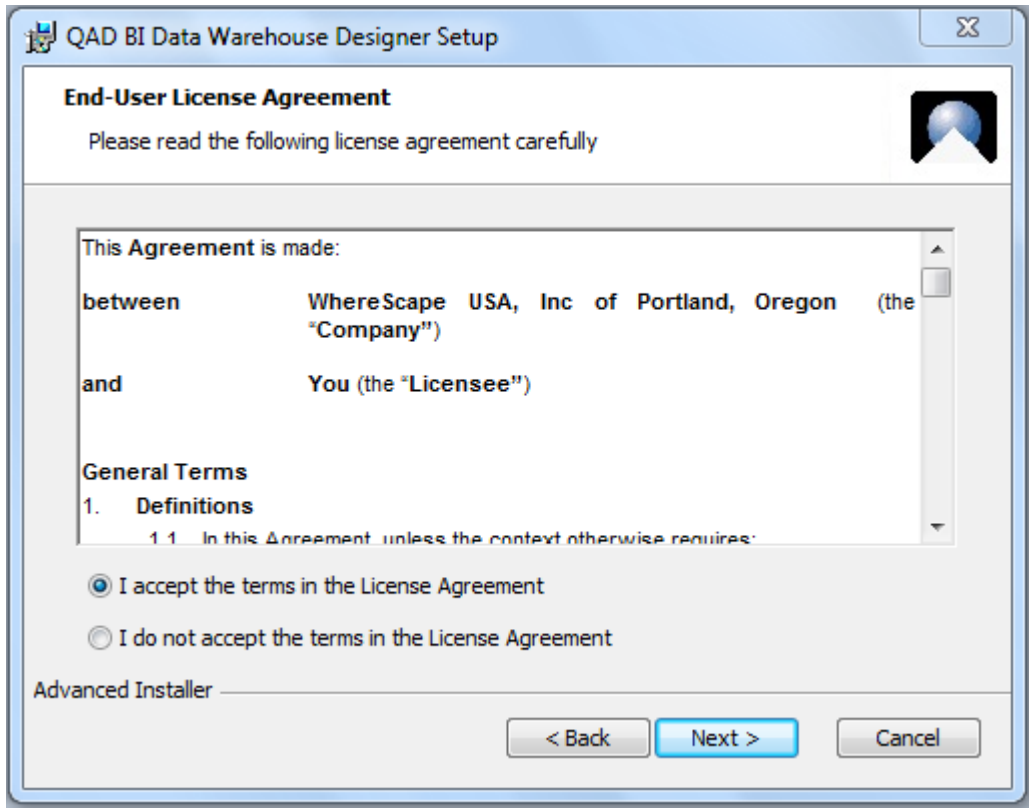
1. Stop any QAD Data Warehouse Designer schedulers running on the PC.
2. Close any other applications that are running.

Insert the QAD Data Warehouse Designer CD into your CD drive or double click the downloaded file Data Warehouse Designer_xxxxx.msi (For example: the downloaded file for version 5.6.4, would be Data Warehouse Designer_56400.msi). If using a CD, the setup program will launch automatically, displaying a dialog box that welcomes you to the QAD Data Warehouse Designer installation process.



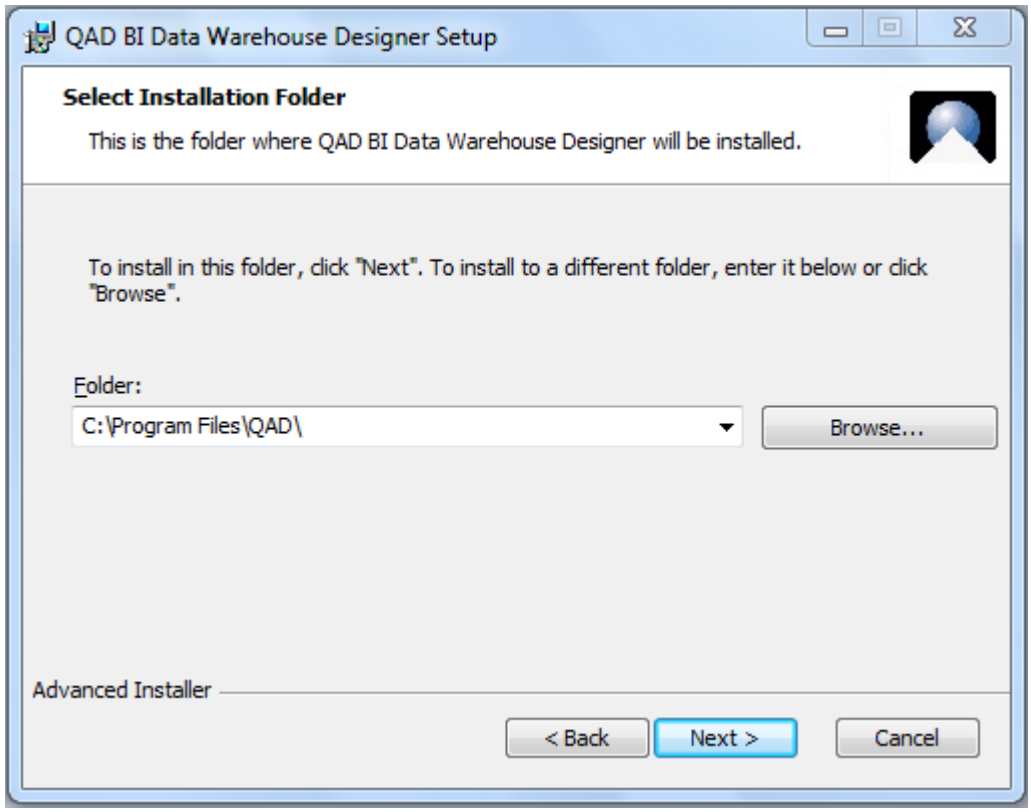
If you're using an installation CD and the setup program does not launch automatically, use Windows Explorer to browse to the CD, then open the file Data Warehouse Designer_xxxxx.msi.

Click the Next button. The License Agreement dialog box appears.

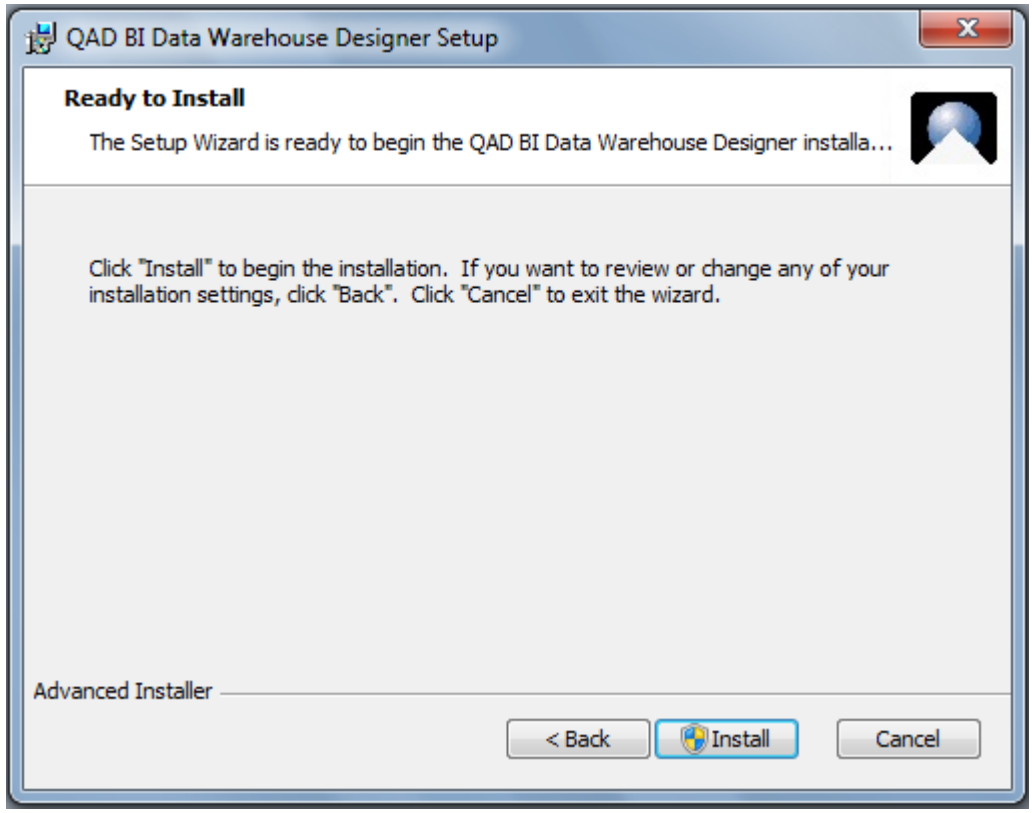


Read the license terms and select the I Accept the terms in the License Agreement option then click Next. If you do not agree to the license agreement, you cannot continue with this installation.

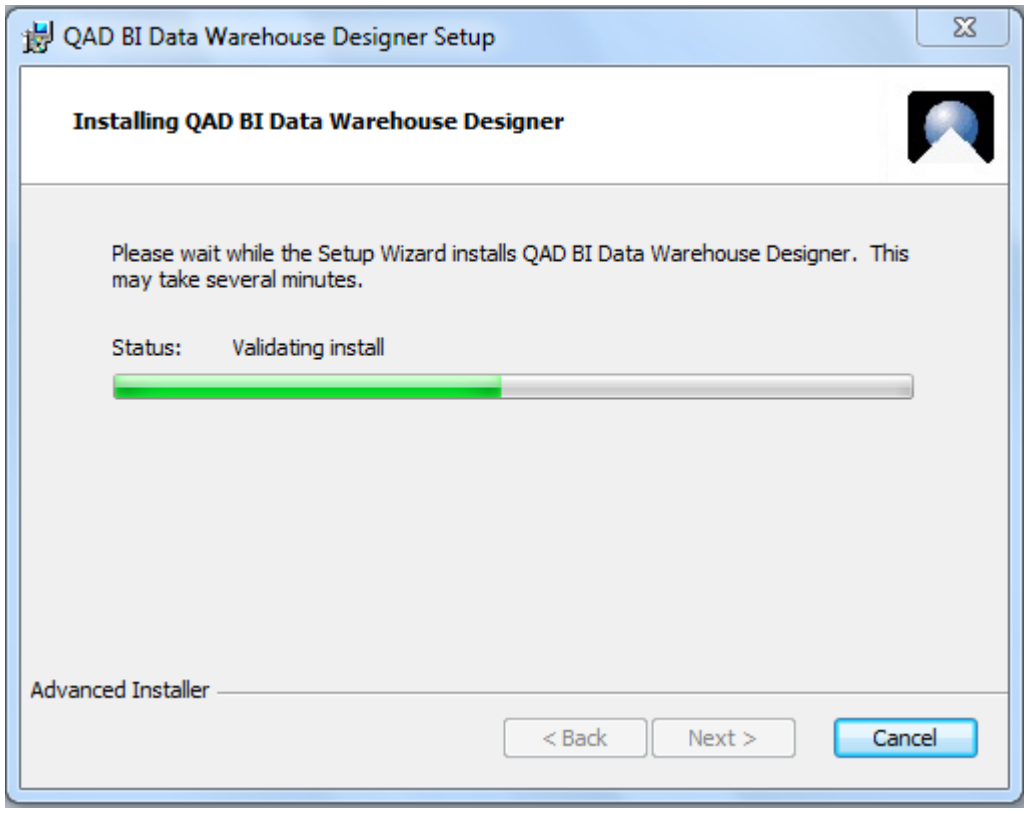
The Destination Location dialog box appears, showing the directory where the QAD Data Warehouse Designer software will be installed. To change this directory, click Browse...



Click the Next button. The Ready to Install dialog box appears, which provides a last chance to go back and change any of the information you have entered so far.



Click the Install button to install the QAD Data Warehouse Designer software. When installation begins the Installing dialog box opens and shows you the progress as files are copied.



When the process is complete, another dialog box tells you that QAD Data Warehouse Designer has been installed successfully.

Under normal circumstances the dialog displayed will be as follows. If, however, a scheduler was running or a program open (when doing an upgrade) then a dialog will appear advising that a reboot is required to finish the install.

Click the Finish button to complete the installation.

3 Remove the CD from your CD drive and store it in a safe place.

4 Proceed to the introduction to the Setup Administrator utility that was installed as part of QAD Data Warehouse Designer. The installation will continue with this utility.

CHAPTER 2

Upgrading QAD Data Warehouse Designer

QAD versions

QAD Data Warehouse Designer has a four part version number normally shown as xx.xx.xx.xx. An example of this may be 6.0.6.0. The first number represents the major release. The second number represents the metadata repository version. The third and fourth numbers relate to application specific releases.

From the example above we see that the current version is version 6 of QAD Data Warehouse Designer. We are on version 6.0 of the metadata repository.

Metadata Changes

A change from a 6.0.. release to a 6.5.. release would indicate a change in the metadata tables. All metadata changes are non destructive. They simply add new columns or new meta tables. In this way they can be applied without harming an existing metadata repository. The impact of a metadata change is that the associated applications (namely the Data Warehouse Designer executable, the Scheduler, security module etc.) will need to be at the same metadata version. Therefore a 6.0.6.0 version of Data Warehouse Designer may not successfully run against a version 6.5 metadata repository.

Application changes

The final two numbers in the version represent application releases. Applications are deemed to be all of the executable images supplied with QAD Data Warehouse Designer as well as the UNIX scheduler scripts and the stored procedures. Application changes reflect enhancements and bug fixes. A change in the first number indicates a major enhancement in one of the application components.

Upgrading Data Warehouse Designer

Upgrading QAD Data Warehouse Designer consists of the following steps:

1. Allow any active jobs to complete. Halting active jobs will allow running tasks to complete with no new tasks starting. Aborting active jobs will kill any running tasks and stop running jobs.
2. Stop all schedulers. Windows schedulers can be stopped with the QAD Setup Administrator. To stop a UNIX or Linux scheduler, kill the active scheduler process and comment their crontab entries (to stop the scheduler re-starting itself).
3. Close any QAD programs that are running on your machine.
4. Install the new version of Data Warehouse Designer on your machine.

5. Back up your metadata.
6. Validate the metadata by selecting the 'Validate metadata' option in Setup Administrator. For an *Application* upgrade, this will compile any new metadata procedures and re-compile existing metatdata procedures. For a *Metadata* upgrade, some tables may be altered, new tables created, new metadata procedures compiled and existing metadata procedures recompiled.
7. Back up your metadata again (just in case).
8. If using a UNIX or Linux scheduler and this is a major application enhancement then rename the wsl/bin directory to say wsl/bin_versionxxxxxx. Create a new bin directory and ftp over the files under the unix directory (see the main install instructions). Change the protections on the files (chmod 750 *.sh).
9. Restart all schedulers. Windows schedulers can be restarted with the QAD Setup Administrator. For any UNIX or Linux schedulers, uncomment any commented out crontab entries - this is enough to restart the schedulers.
10. Restart any halted or aborted jobs.

Notes:

1. There are different versions of the scripts for each database and for unix versus linux. You should also be looking in the sub directory with the highest version number.

2. Metadata tables do not change between minor releases, but metadata procedure often and usually do change.

3. Data Warehouse Designer will not let you sign into an old repository version using a newer version of Data Warehouse Designer

4. Data Warehouse Designer will let you sign into a new repository version using an older version of Data Warehouse Designer, but it will warn you that this may potentially cause issues

5. It is very important when using the windows scheduler to have the installed Data Warehouse Designer version on the scheduler server EXACTLY matching the stored procedures.

6. Side by side installations are possible (two versions in two directories on the same machine), but be careful with schedulers. If you install the new version of Data Warehouse Designer in a new directory, you will have to remove and reinstall all windows schedulers in order for the scheduler(s) to use the new version of Data Warehouse Designer.

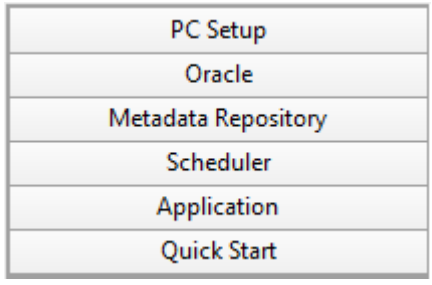
7. As a general rule, the UNIX and Linux scheduler scripts, metadata tables/procedures and Data Warehouse Designer front end should all be in sync.

8. If the scripts for the UNIX or Linux scheduler have changed, you should replace files in your old and new shell scripts in the QAD Program Files directory and see if any have changed.

CHAPTER 3

Setup Administrator Introduction

The remaining chapters of this installation document follow the flow of the 'Setup Administrator' utility found in the QAD program group. If installing QAD Data Warehouse Designer for the first time then you will need to work through each of these chapters. The main task groups (chapters) in this utility are as follows:



The top to bottom flow encompasses the following tasks:

- Validating the ***PC*** (see "***PC Setup***" on page 19) configuration, recording the license key and setting up any ODBC connections.
- Creating a data warehouse user in the database. Testing ODBC access to that user and validating the privileges required to run QAD Data Warehouse Designer.
- Creating a ***QAD metadata repository*** (see "***Create/Validate Meta Repository***" on page 37) under the data warehouse user.
- Creating and populating the ***tutorial tables*** (see "***Create/Validate Tutorial Tables***" on page 41) in the database if required.
- Installing a ***scheduler*** (see "***Install/Maintain Scheduler***" on page 43). (Note: Enterprise only)
- Loading Applications
- Performing a quick start if using SQL Server

The chapter on ***Applications*** (see "***List/Load Applications***" on page 73) relates to the loading of a specific data warehouse application into a metadata repository. It may not be part of the normal installation process unless a specific application model has been shipped with the QAD Data Warehouse Designer product suite. The Quick Start option provides a one click install and load onto a SQL Server server.

In This Chapter

Users of QAD Data Warehouse Designer

It is envisaged that QAD Data Warehouse Designer will be used by experienced data warehouse designers and implementers. QAD Data Warehouse Designer provides an environment to facilitate the rapid production of prototype data warehouse analysis areas. It also provides the functionality to move that prototype into a production environment and support the day-to-day running of a data warehouse.

Problem solving

Problem Solving

If problems occur in your install that cannot be resolved proceed as follows when running the QAD 'Setup Administrator' utility:

1. Create a log file by selecting Log/Start log from the menu bar. Enter a file name.
2. Step through at least:
   - PC Setup --> Check Windows version
   - PC Setup --> Check ODBC driver
   - PC Setup --> Validate/Add Odbc source
   - DSS User --> Validate Oracle user privileges (if an Oracle DataWarehouse)
   - Create/Validate Meta Repository --> Validate meta repository
3. Select Log/Stop log to close off the log
4. Edit the log and remove any passwords that appear.
5. Mail the log file to http://support.qad.com

CHAPTER 4

PC Setup

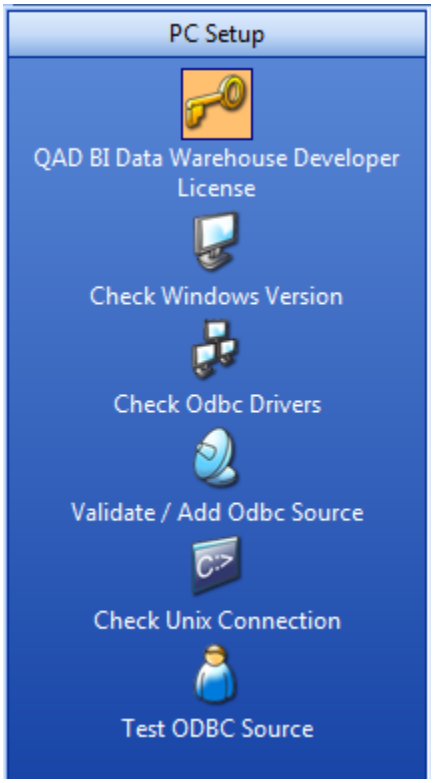
In This Chapter

PC Setup Introduction

The PC Setup component of the QAD 'Setup Administrator' attempts to ensure that the PC environment is suitable for the running of QAD Data Warehouse Designer. This includes validation of the environment, and the checking of the ODBC connectivity.

The normal procedure is to step down the tasks in order.

Click on the 'PC Setup' icon to start. Click on any of the options under this level to action them. The PC Setup menu is as follows:



- ***QAD License*** (see "***License***" on page 31) allows the entry of the license key.
- ***Check Windows Version*** (see "***Validate Windows Setup***" on page 21) simply checks to ensure that the PC is running an acceptable OS and has a screen resolution of at least 1024x768.
- ***Check ODBC drivers*** (see "***Validate/Install ODBC Driver***" on page 21) checks that a suitable version of the ODBC drivers are installed..
- ***Validate/Add Odbc Entry(s)*** (see "***Check / Add Odbc Entry(s)***" on page 22) checks and allows the adding of ODBC sources.
- ***Check Unix connection*** (see "***Check Unix System Connectivity***" on page 33) will test the connection to a Unix server. (if Unix is being used)
- Test ODBC Source - will test the ODBC connection.

Validate Windows Setup

QAD Data Warehouse Designer requires a Windows 2003, Windows XP, Windows 2008, Windows 2008 R2, Windows Vista or Windows 7 environment with a screen size set to at least 1024x768. This step simply validates that the above requirements have been met. The QAD Data Warehouse Designer front end will run in other Windows environments, but the scheduler may not. QAD Data Warehouse Designer will not run under Windows 95, 98 or NT. Please note that if the minimum screen size of 1024x768 is used then the standard font size must also be used.

Validate/Install ODBC Driver

Both the SQL Server and Microsoft for Oracle ODBC drivers are shipped as normal in Windows 2003, Windows XP, Windows 2008, Windows 2008 R2, Windows Vista and Windows 7. This section will normally be a validation of the pre installed drivers.

Oracle

QAD Data Warehouse Designer connects to the data warehouse database using ODBC. This step checks that the ODBC driver is present and up to date.

A list of ODBC drivers will be displayed provided the PC user has sufficient privileges to access the registry.

Check / Add Odbc Entry(s)

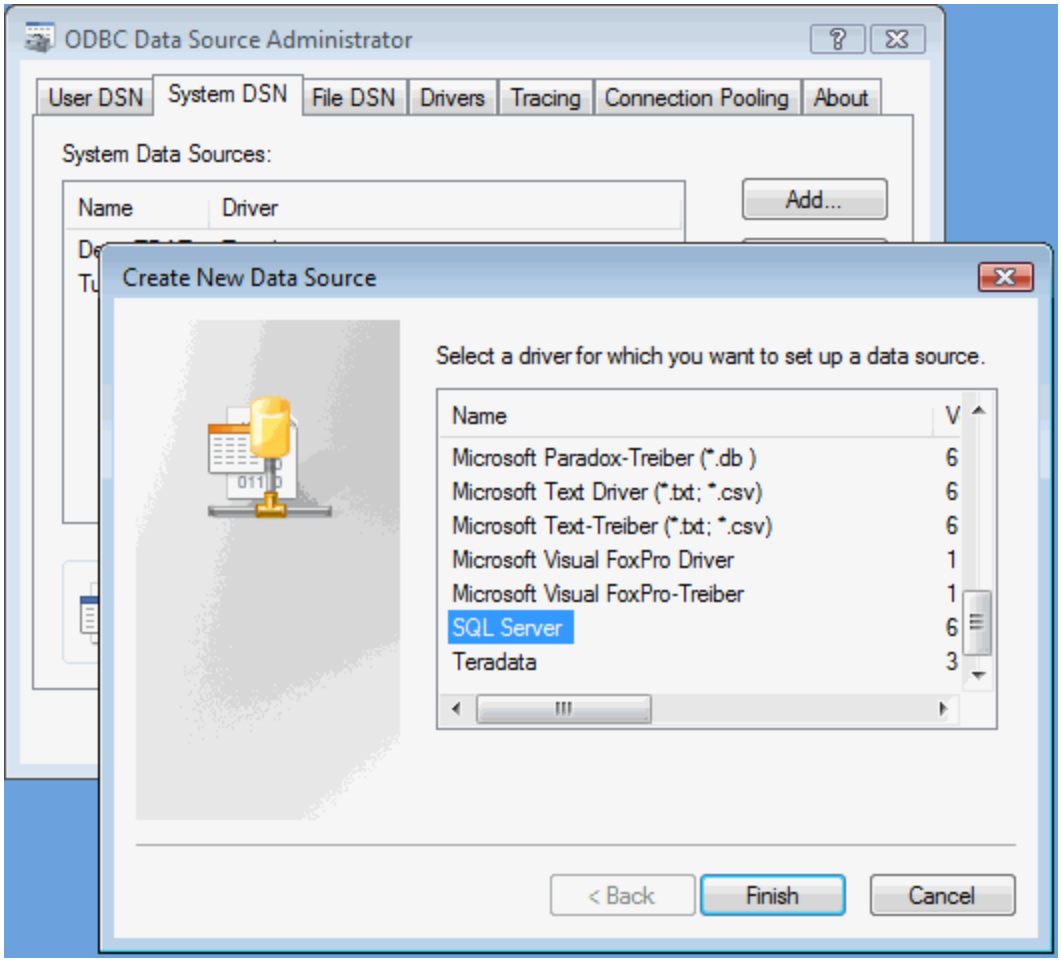
An entry needs to exist in the ODBC tables for the data warehouse database. This step lists all entries found that utilize an ODBC driver for Oracle, SQL Server or Teradata. If no entry exists for the data warehouse database then it will be necessary to add one. Right mouse in the right hand frame to bring up the 'Add Odbc entry' menu option.

Adding a SQL Server ODBC Source

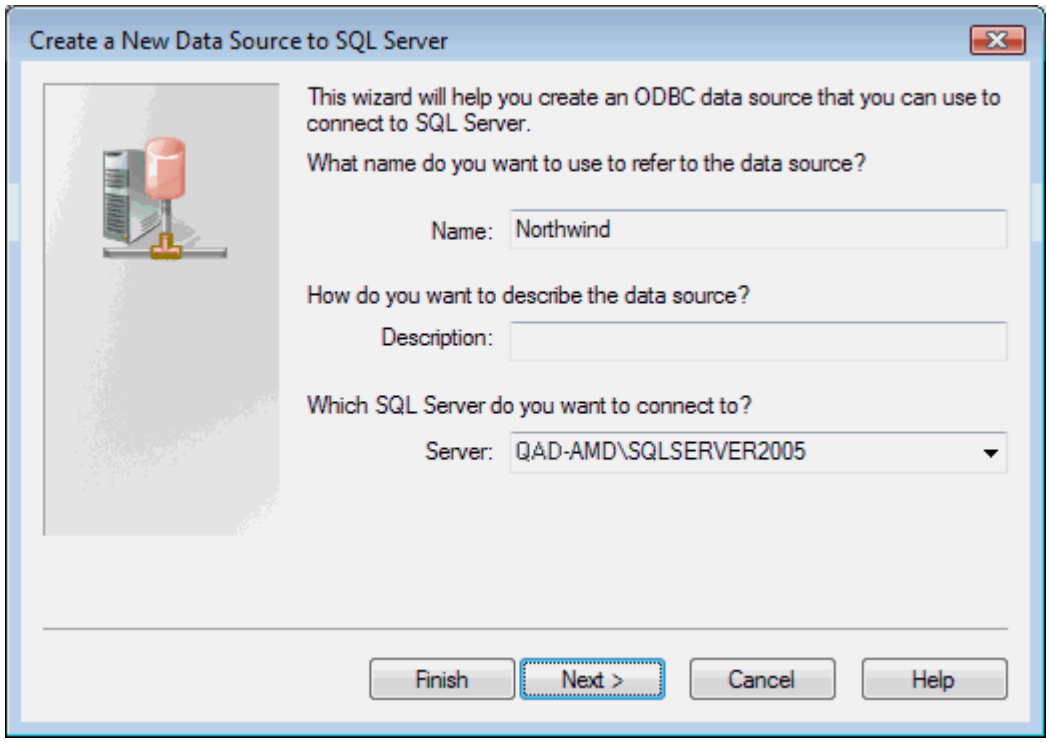
The normal process for adding a SQL Server ODBC source is as follows:

Select the 'System DSN' tab to view the system wide ODBC entries

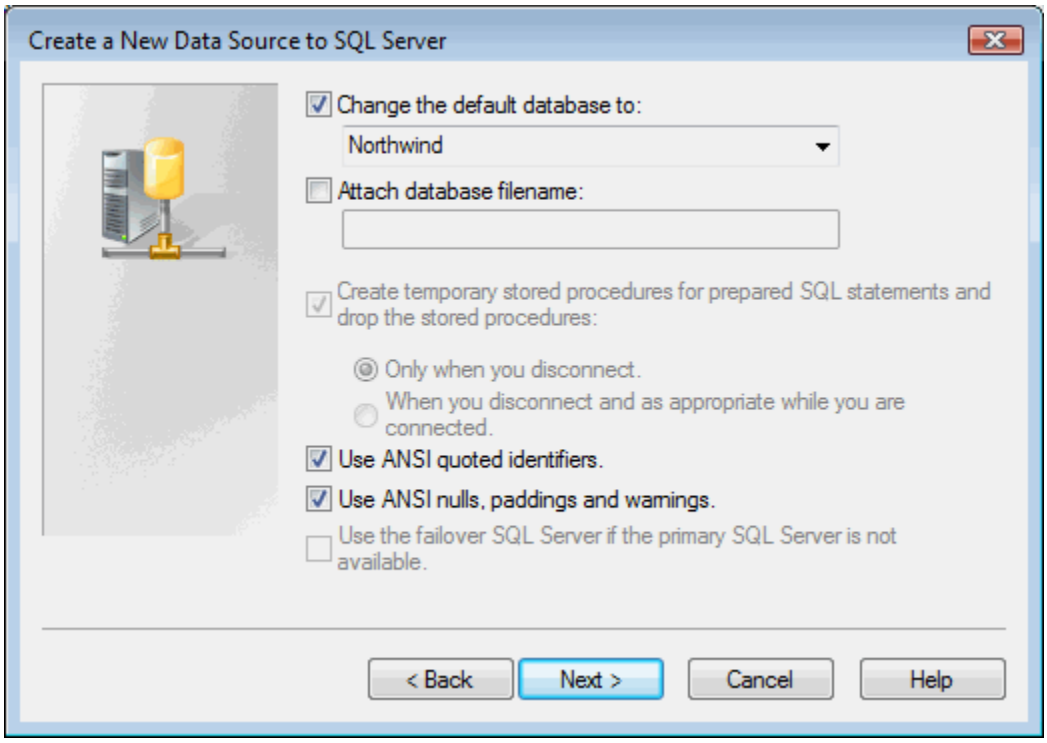
Click on the 'Add...' button to add a new entry



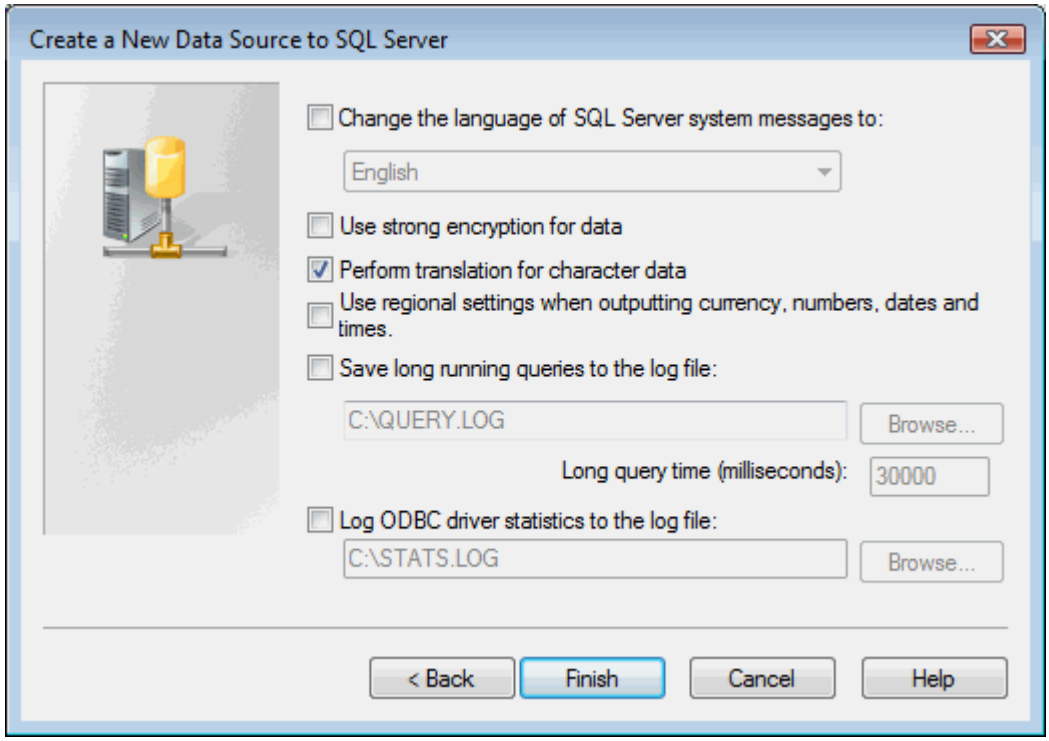
Select a 'SQL Server' driver from the driver list. A popup as shown below will appear. Enter a name for the ODBC Source and select or enter the server and click 'Next'.



A new popup will appear with authentication details. Change as required or leave as shown if you are unsure. When 'Next' is clicked a popup will provide the opportunity to select a database. The following example shows the selection of the NorthWind database. Click 'Next' when the database has been selected.



A further popup will appear to allow the setting of other features if desired.



Ensure that the regional settings check box is not used. If this is used then date and number formats will be in regional standards rather than native form.

Click 'Finish'.

Adding an Oracle ODBC Source

The normal process would be

Select the 'System DSN' tab to view the system wide ODBC entries

Click on the 'Add..' button to add a new entry

Select the 'Microsoft ODBC for Oracle' driver from the driver list. Other Oracle drivers may exist.

Click the 'Finish' button. At this stage the connection to the Oracle client is checked, so if a failure occurs it is probably due to the Oracle client not being present or installed incorrectly.

A small dialog box will appear asking for a 'Data Source name', 'Description', 'User Name' and 'Server'. The Server is the Oracle client connection to the data warehouse database, and is typically (but not necessarily) the SID. The Data Source Name and Description can be set to the SID for simplicity. The 'User Name' is not required.

Note: The Server must be a valid Tnsnames entry

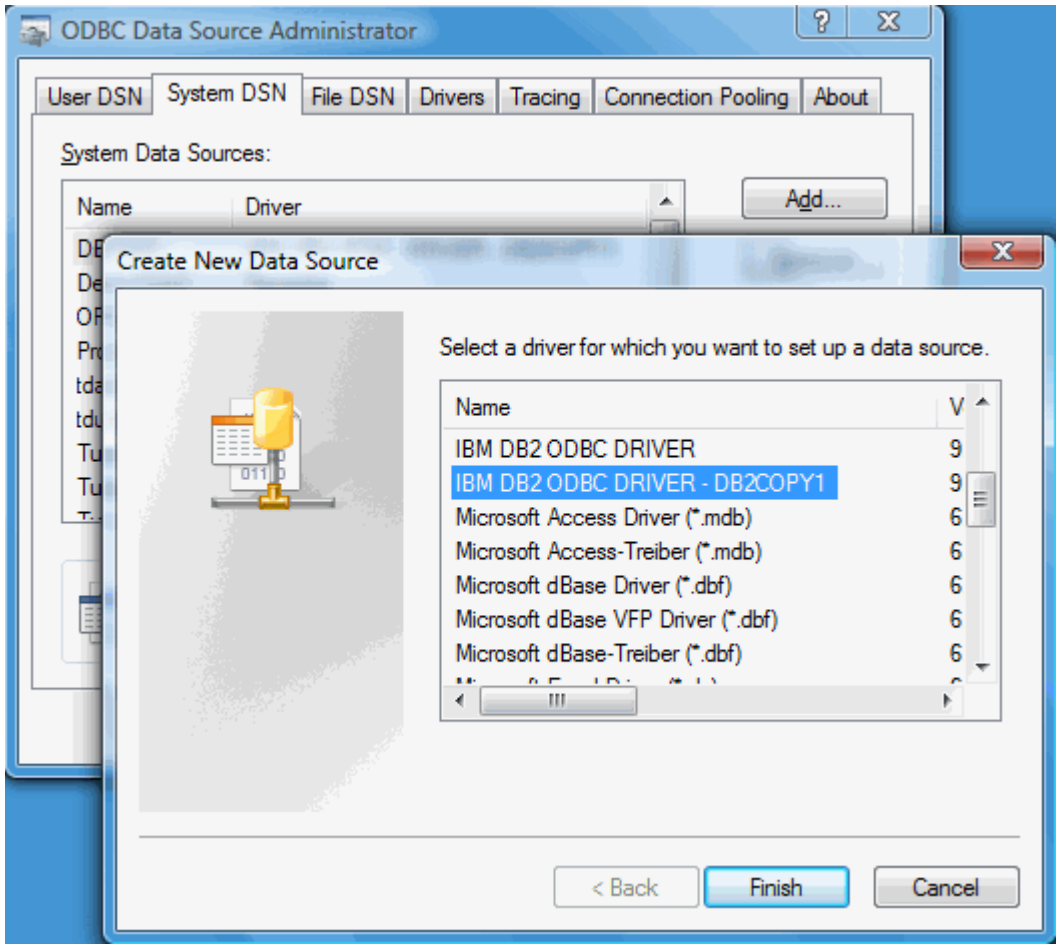
Adding a DB2 ODBC Source

Note: A DB2 script to create a database and data warehouse schema needs to be run before the ODBC driver can be configured. See Create DB2 Dss Schema

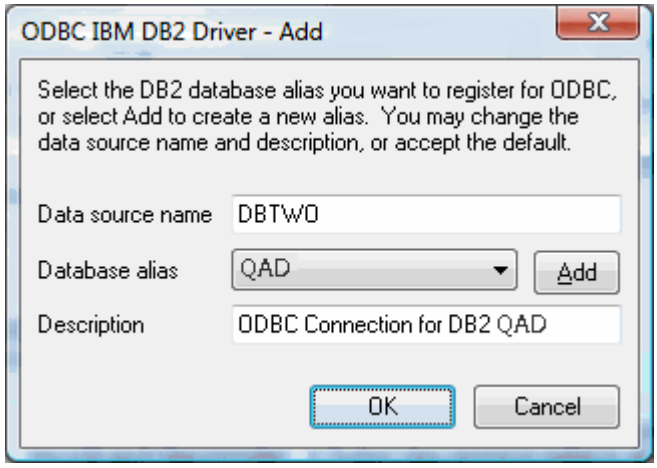
The normal process for adding a DB2 ODBC source is as follows:

Select the 'System DSN' tab to view the system wide ODBC entries

Click on the 'Add...' button to add a new entry



Select a 'IBM DB2 ODBC DRIVER' driver from the driver list. A popup as shown below will appear. Enter a name for the Data Source Name. Choose the database alias from the drop down list. Click OK.



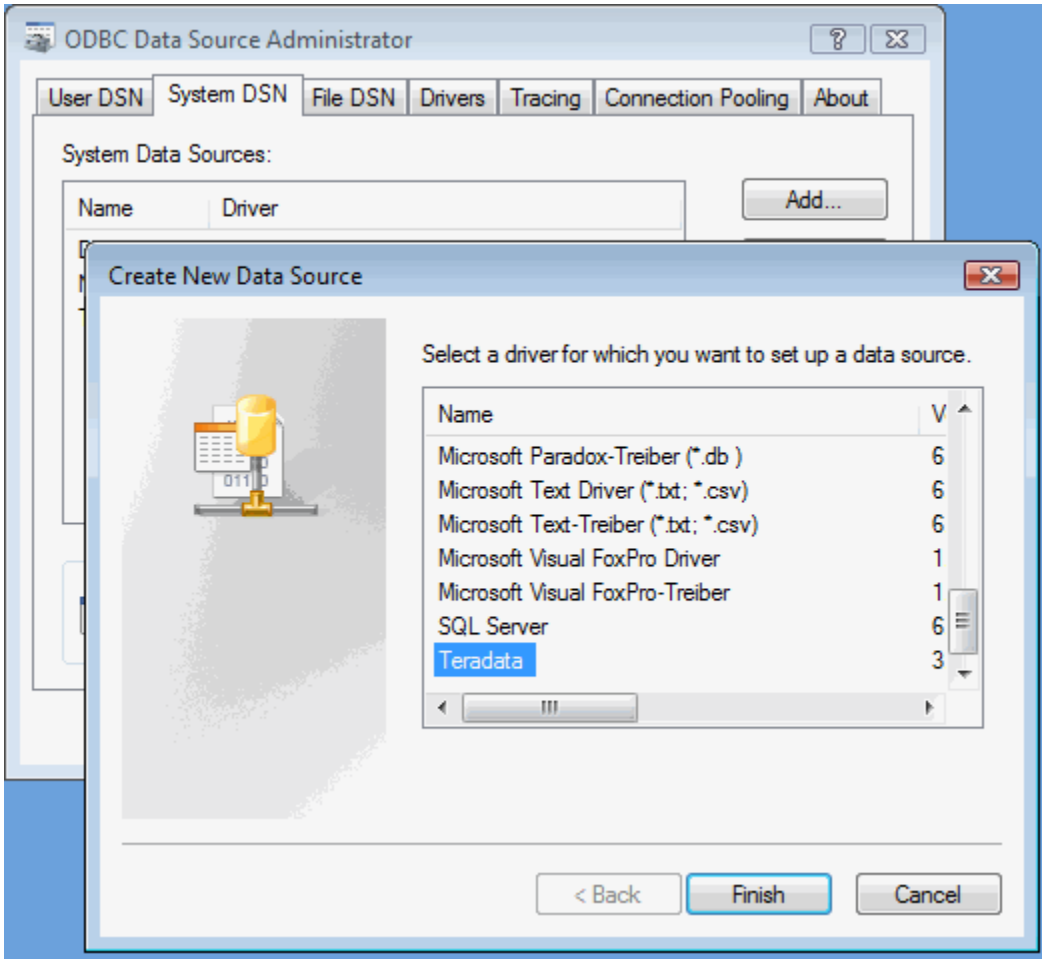
If using the ODBC source for a Data Warehouse Designer Scheduler, click on 'Run in Quiet Mode'. Click OK. Click OK again.

Adding a Teradata ODBC Source

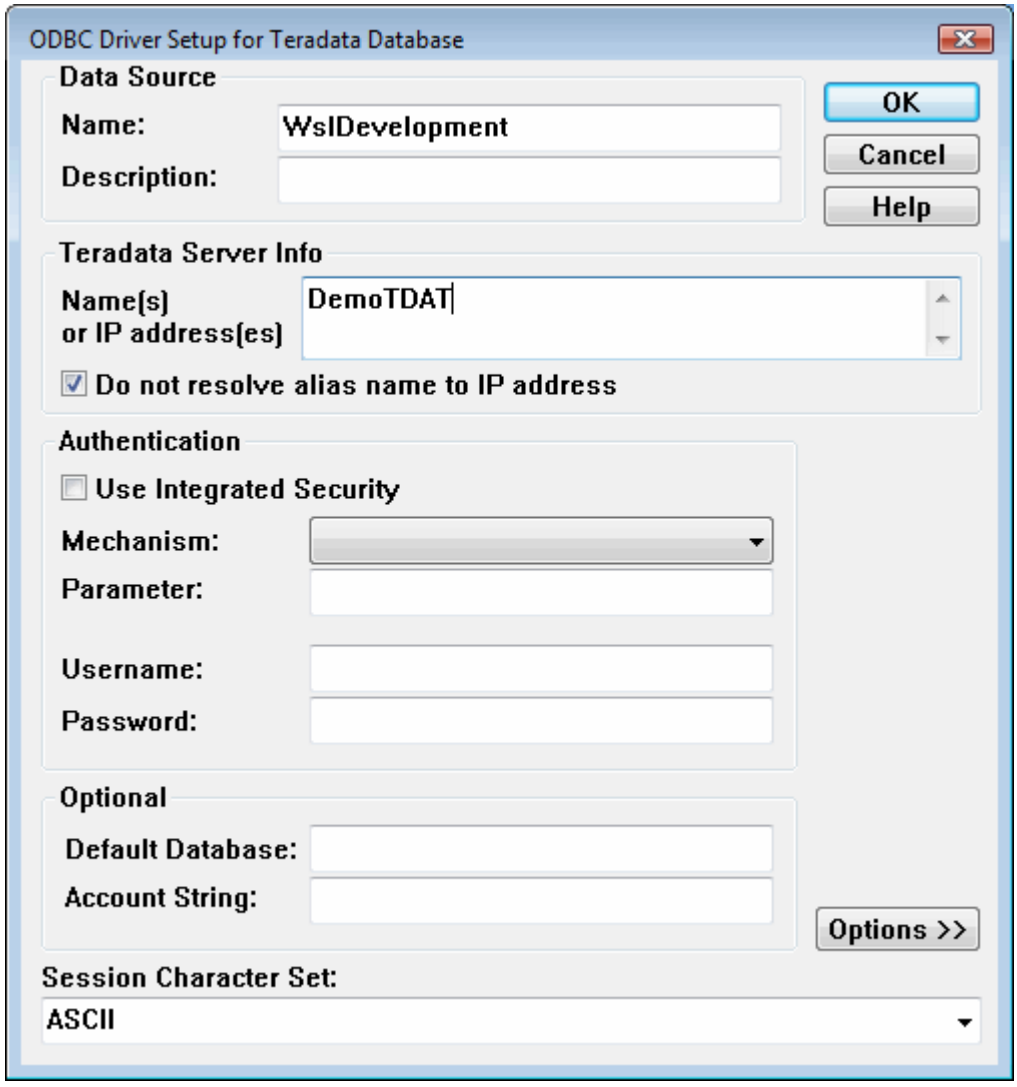
The normal process for adding a Teradata ODBC source is as follows:

Select the 'System DSN' tab to view the system wide ODBC entries

Click on the 'Add...' button to add a new entry



Select a 'Teradata' driver from the driver list. A popup as shown below will appear. Enter a name for the Data Source Name. Next, enter the Name or IP Address for the Teradata Server Info Name and check on the 'Do not resolve name to IP address' check box. Click Options.



Note: The name entered for the 'Teradata Server Info Name' should appear in the 'hosts' file on the PC in this form:

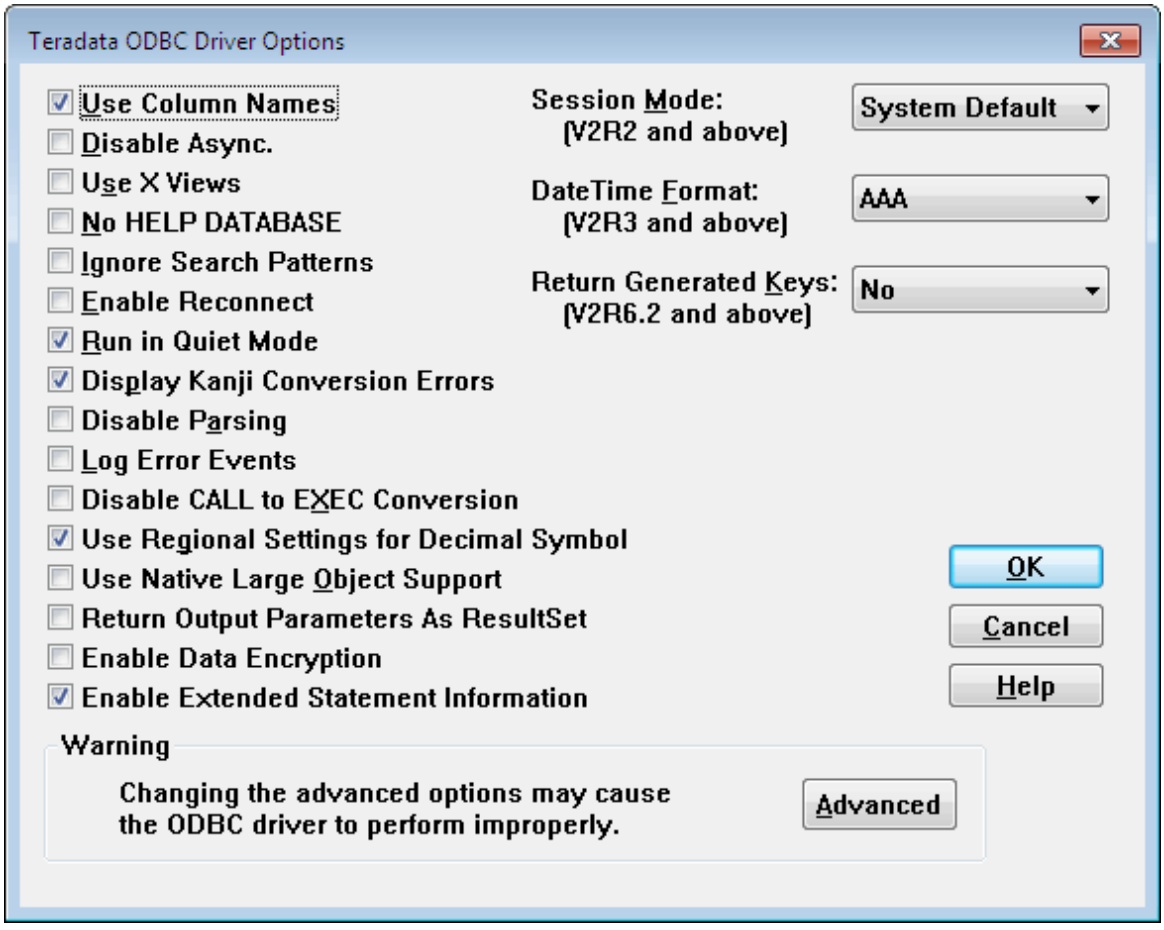
(ip address) (Teradata Server Info Name)Cop1

For the example in the ODBC screen shot above for DemoTDAT, this entry would be present if the server had an ip address of 10.12.123.60:

10.12.123.60 DemoTDATCop1

An alternative approach is to add the same entry to a dynamic name server (DNS).

If using the ODBC source for a Data Warehouse Designer Scheduler, click on 'Run in Quiet Mode'. Click OK. Click OK again.

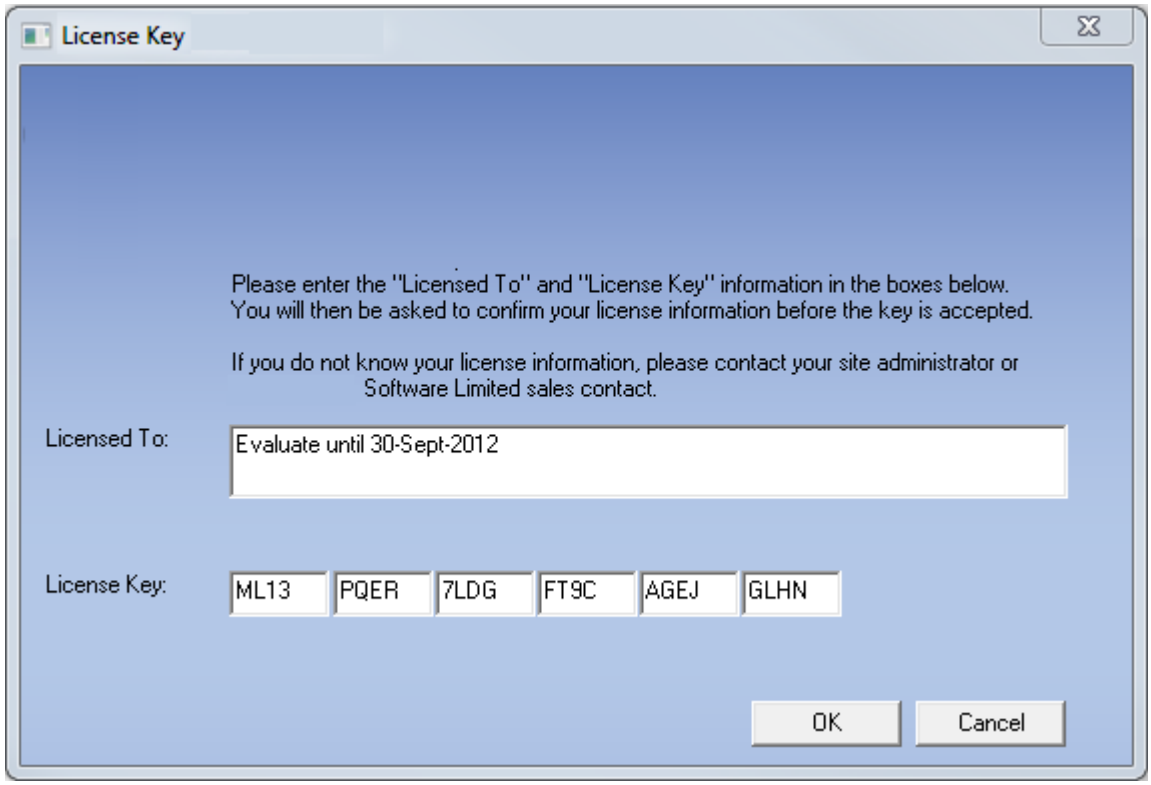


TIP: If multiple PCs will access the data warehouse they must all use the same data source name as entered here.

License

QAD Data Warehouse Designer requires a valid licensee and license key. This information and key are normally supplied in a separate email. This menu option allows the entry of the license information. Once validated the license information is stored in the registry and validated by the QAD Data Warehouse Designer program on startup.

Note: If you have not been supplied a license key then you can still use QAD Data Warehouse Designer. You will however be limited in the number of objects that you can create. The first four tutorials can be performed without a license key.

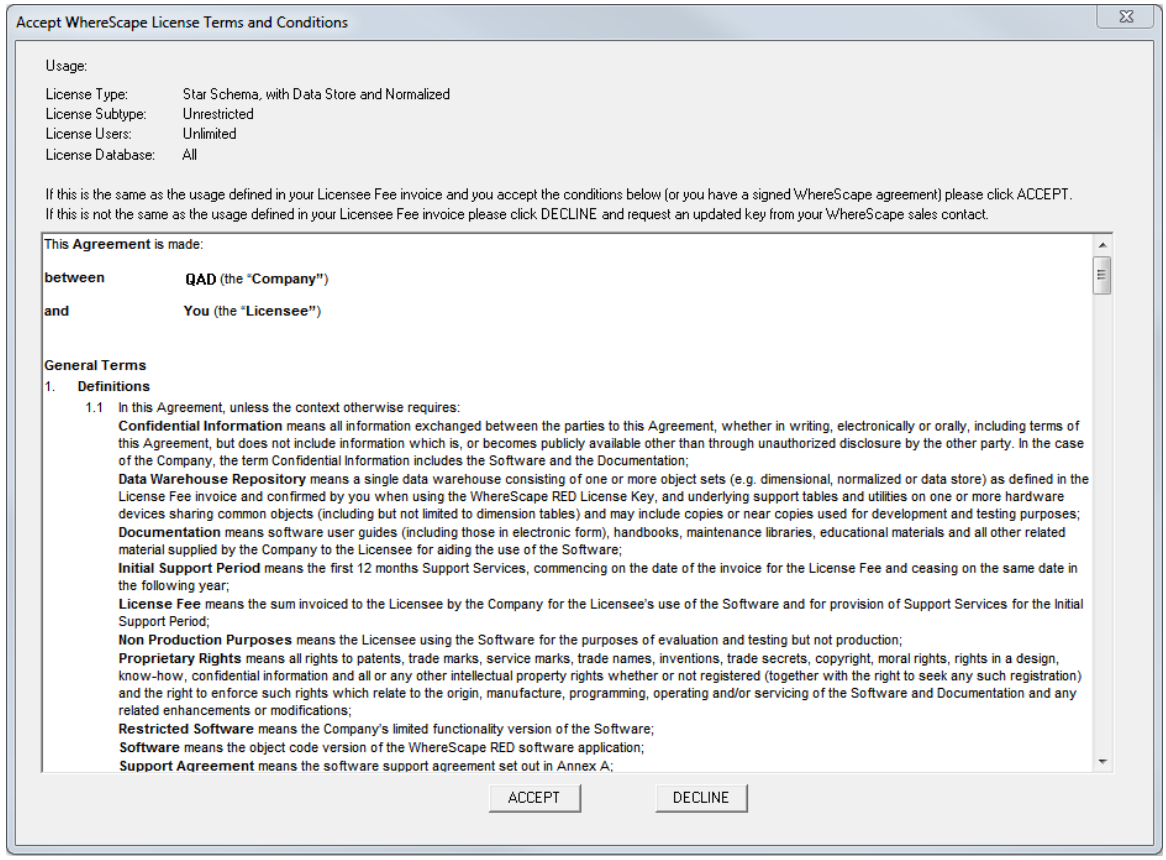


The example above shows the entry of an evaluation license and key.

If the key fails to validate contact http://support.qad.com.

Enter the license fields and click OK.

The license type dialog now displays:



Click ACCEPT to confirm the type of license.

Check Unix System Connectivity

Note: IBM DB2 and Oracle data warehouses only

QAD Data Warehouse Designer can allow the user to browse a Unix file system and utilize the drag and drop functionality to setup the loading of files from a Unix file system. It can also perform interactive file loads. To facilitate this and to obviate the need to port code to every Unix environment the Telnet protocol is used for the Unix connectivity.

This step is not necessary for the successful running of a Unix based data warehouse environment. As mentioned above it is provided to simplify the process of setting up the loading of Unix files. This step is totally unnecessary for a Windows based data warehouse environment.

If you wish to utilize these Unix drag and drop assistants then consult the QAD Data Warehouse Designer User Guide for an explanation of Unix connections and perform the following tasks:

1. Identify a Unix user that allows direct logon to a Unix shell prompt without passing though a menu system.
2. Ensure that the Unix user has the Oracle 'sqlldr' program in the path after a normal logon. If not then interactive loads will not be possible. Scheduled loads should still function correctly.
3. Complete the dialog box presented specifying the Unix user and password identified in step (1).
4. Press the 'OK' button to proceed. A timeout of 20 seconds is enabled, so that if connection is not achieved within that time the process will terminate. The Telnet window will be displayed to assist in debugging any problems. It will be closed on completion or timeout.
5. Review the results and attempt to address any problems. A file called 'WslTelnet.log' is created in the installation directory containing the full dialog of the connection attempt. The message 'Telnet connection worked. All OK' will be displayed if the connection worked. If successful record the parameters used as these will need to be defined when setting up a Connection within QAD Data Warehouse Designer itself. If a connection cannot be achieved then remove all reference to a username and password from the file mentioned above and mail it to http://support.qad.com.

CHAPTER 5

SQL Server

In This Chapter

SQL Server Permissions

The minimum SQL Server permissions are as follows:

Server roles:

- bulkadmin - this is required if you want to load files or use native odbc loads

Database roles (on the database with the RED metadata in it):

- db_datareader
- db_datawriter
- db_ddladmin

Database grants (on the database with the RED metadata in it):

- SELECT
- INSERT
- UPDATE
- DELETE
- EXECUTE

Permissions to kill process when killing jobs in teh scheduler:

- VIEW SERVER STATE
- ALTER ANY CONNECTION

NOTE: When creating a new repository or validating the metadata in an existing repository, you're required to log into SQL Server with a user who has SYSADMIN or the ALTER TRACE permission.

This is because almost all the QAD procedures have a catch all with log file option exception handler.

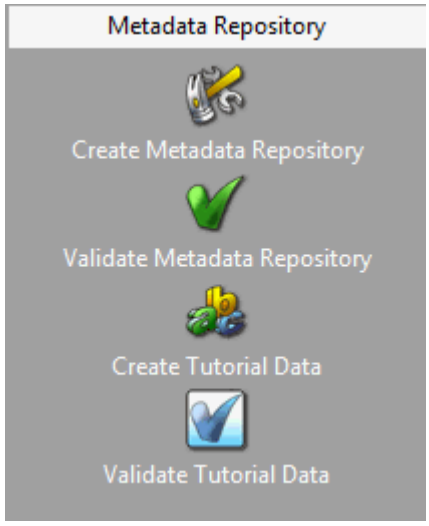
This insures that in the event that the scheduler cannot log errors to the database, it can log errors to SQL Server logs and Application logs.

CHAPTER 6

Create/Validate Meta Repository

The Meta Repository component of the QAD 'Setup Administrator' attempts to validate, upgrade or perform an initial install of the QAD metadata tables.

Click on the 'Create Metadata Repository' or 'Validate Metadata Repository' icon to start. Click on any of the options under this level to action them. The metadata repository menu is as follows:



- The ***validate meta repository*** (on page 38) option can be used at any time to ensure that the meta repository is at the latest version.
- A ***new meta repository*** (see "***Create/Setup Meta Repository***" on page 38) must be created for each data warehouse instance.

In This Chapter

Validate Meta Repository

This option validates the version of the meta tables installed in the specified SQL Server database, Oracle schema, DB2 schema or Teradata Database/User. If requested it goes on to validate the tables, sequences and procedures against the installation media. A validate can be run at any stage to confirm that all tables and procedures are up to date and that the procedures are in a compiled and valid state. This option can be used to recompile a procedure that has become invalid. This option should be used when installing a new version of Data Warehouse Designer and for SQL Server all procedures should be recompiled.

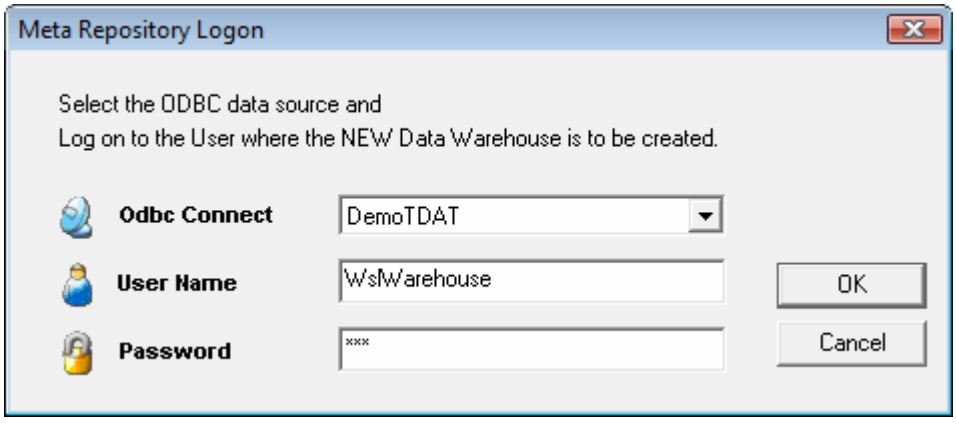
If a mismatch is found or a procedure is found to be invalid, the user is given the option of updating the metadata. If chosen the metadata will be upgraded to the latest version.

Note: Ensure the scheduler is not running and no users are accessing the metadata before doing an upgrade.

Create/Setup Meta Repository

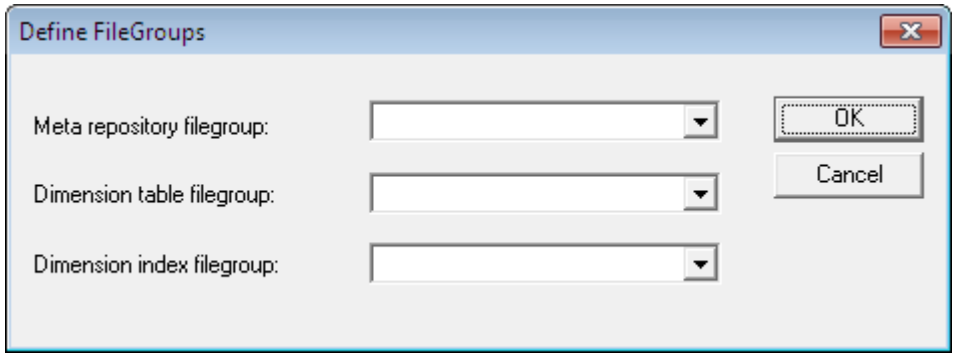
This option will create the QAD meta tables, sequences and procedures in the specified schema. It checks to ensure that the schema, database or user does not contain an existing metadata repository before starting the metadata install.

Click on Create Metadata Repository. Enter the ODBC Connect, User Name and Password and Click OK.



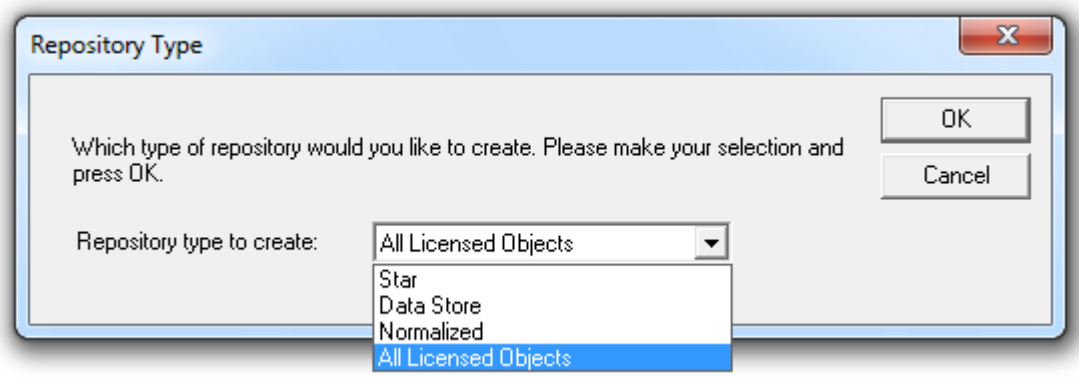
Meta Creation - SQL Server

When creating a SQL Server metadata repository, an additional screen is displayed requesting filegroup information:



Select the required filegroups and click OK. If filegroups are not being used, leave the file groups blank and click OK.

Depending on the type of license you have, the following dialog may be displayed:



If it is displayed, select the type of repository required and click OK.

Note: The metadata repository and data warehouse tables must both be created under the 'dbo' schema when using SQL Server as the data warehouse server. Ensure that the user used to create the meta repository has the required privileges to create objects in dbo.

Note: This is just setting the default objects visible in Data Warehouse Designer. They can later be changed under Tools/Options.

CHAPTER 7

Create/Validate Tutorial Tables

The tutorial tables component of the QAD 'Setup Administrator' attempts to validate, upgrade or perform an initial install of the QAD tutorial tables. The tutorial tables are only required if the tutorial is to be undertaken. They can be ignored if desired.

Click on the 'Create/Validate Tutorial tables' icon to start. Click on any of the options under this level to action them.

- The ***validate tutorial tables*** (on page 41) option will check for the existence of QAD Data Warehouse Designer tutorial tables.
- The ***create/setup tutorial tables*** (on page 42) option creates (or re-creates) the set of tables in the specified schema (user).

In This Chapter

Validate Tutorial Tables

This option validates that the tutorial tables exist in the given schema.

The tables are based around a simple sales model and are as follows:

Table	Contents
product	A list of products including a product code (sku), description, group and subgroup
prod_group	A description of each product group
prod_subgroup	A description of each product sub group
prod_line	A description of each product line
code_table	A table containing the codes and descriptions for product groups, sub groups and lines. A code type of 'pg','sg' and 'pl' is used to differentiate the different codes.
customer	A list of customers containing names and addresses.

order_line	The lines on a sale. Includes the specific product code, quantities and values
order_header	The header for a sale. Includes the customer code, order data and ship date
kpi	Used to generate a KPI dimension
budget	Budget data for sales of products to customers
forecast	Demand forecast data for products and customers

Create/Setup Tutorial Tables

This option will create, or re-create, the QAD tutorial tables within the schema (user) or database provided. The normal practice is to locate these tables in a different schema from the data warehouse user for Oracle and in a different database for SQL Server. The data warehouse (DSS) user will need select access to these tables to perform the tutorial. The tutorial documentation assumes these tutorial tables will be loaded under the user 'wtutorial' for Oracle or in the Tutorial database for SQL Server. Another user or database may be used if desired. The user chosen must have sufficient privileges to allow the creation of the tables.

When this option is chosen an ODBC connection dialog will appear, and if a valid user is entered the tables will be created and populated under the specified user.

Please ensure that no tables that match those shown in the Validate tutorial tables section exist under the specified user. If existing tables are found, a prompt will be issued, and if proceed is chosen then all data will be deleted from the tables and the tutorial data loaded.

Teradata

When creating tutorial tables for Teradata, an additional dialog is displayed for the user or database name that the tutorial data will be created in. Enter the name of the user or database in which the tutorial data will be created and click OK.

CHAPTER 8

Install/Maintain Scheduler

In This Chapter

Scheduler Introduction

The Install/Maintain scheduler component of the QAD 'Setup Administrator' provides assistance in the setup of the scheduler. The bulk of the work undertaken by the scheduler is performed within the database by the QAD procedures. There is however a component that must run on the data warehouse database host or another server. Unix and Windows schedulers exist for Oracle and DB2 data warehouses. A windows scheduler is used for SQL Server and Teradata data warehouses (and is also available for Oracle and DB2).

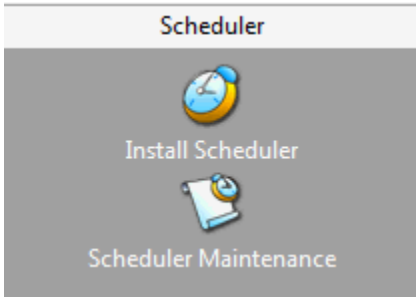
The windows scheduler can be used to action the bulk of the tasks on a Unix hosted data warehouse, but it is not capable of running any Unix file or script based loads. If ODBC based loading is to be used then a Windows scheduler will be required for any scheduled loads.

In an environment that runs the data warehouse on a Unix platform it would be normal practice to run the Unix scheduler.

It is valid to have both a Windows and Unix scheduler running at the same time, and even multiple copies of the scheduler running in each environment.

 TIP: A separate scheduler must be setup for each distinct meta repository (i.e. for each different database user which has a meta repository).

Click on the 'Install/Maintain Scheduler' icon to start. Click on any of the options under this level to action them. The Scheduler menu is as follows:

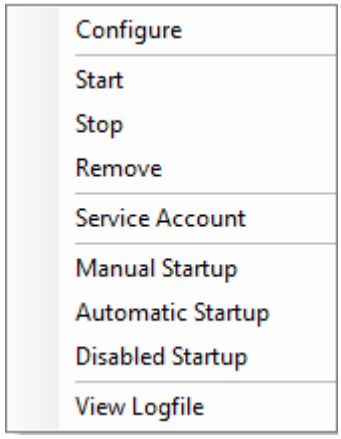


- The Windows scheduler runs as a system service. The ***Maintain scheduler*** (see "***Maintain Windows Scheduler***" on page 45) option lets you control a scheduler service.
- The ***Install Scheduler*** (see "***Install Windows Scheduler***" on page 47) option defines and installs the scheduler as a Windows service
- The ***Install Unix Scheduler*** (on page 67) provides a list of tasks to be undertaken in the setting up of a Unix scheduler

Note: A windows scheduler must be run as either the system service account or as a user with administrator privileges in windows.

Maintain Windows Scheduler

This option lists all currently active QAD schedulers. By using the right mouse menu when positioned over a scheduler name it is possible to start, stop, remove, change the settings, or view the log of a scheduler. The menu is as follows:



The options are:

Option	Description
Configure	Allows the alteration of the service configuration. You can change the level of logging. The default is 2. Level 0 provides no log information and level 9 provides full debug. See 'Scheduler creation' for more detail.
Start	Starts a stopped scheduler. This action requires service control manager (SCM) privileges on the PC.
Stop	Stop a running scheduler. SCM privilege required.
Remove	Removes and deletes the scheduler. SCM required.
Service Account	By default the scheduler will start in the system account. If you wish to change this you can specify an account and password. However, you may have to remove the service and enter the account information when creating a new scheduler.
Manual Startup	Changes the startup profile (on PC startup) of the scheduler. The scheduler will not be automatically started when the PC starts and must be started manually. SCM required.
Automatic Startup	Sets the scheduler to start automatically when the PC starts. SCM required.
Disabled Startup	Sets the scheduler startup profile to disabled. No startup will occur unless the profile is changed to either Manual or Automatic. SCM required.
View Logfile	Displays the scheduler service log file. The level of logging is defined in the configuration of the scheduler.

Note: If an error occurs while a scheduled job is running, you should check the following:

- the audit log of the failed task, for any messages of type E or F.

- the audit log of the failed job, for any messages of type E or F.

- the scheduler log file for any error messages.

- the windows event logs, specifically the Application and System event logs, for any Error or Warning messages at the time of the job failure.

How to view and manage event logs depends on which version of Windows you are using. If you need further assistance, please contact support at http://support.qad.com.

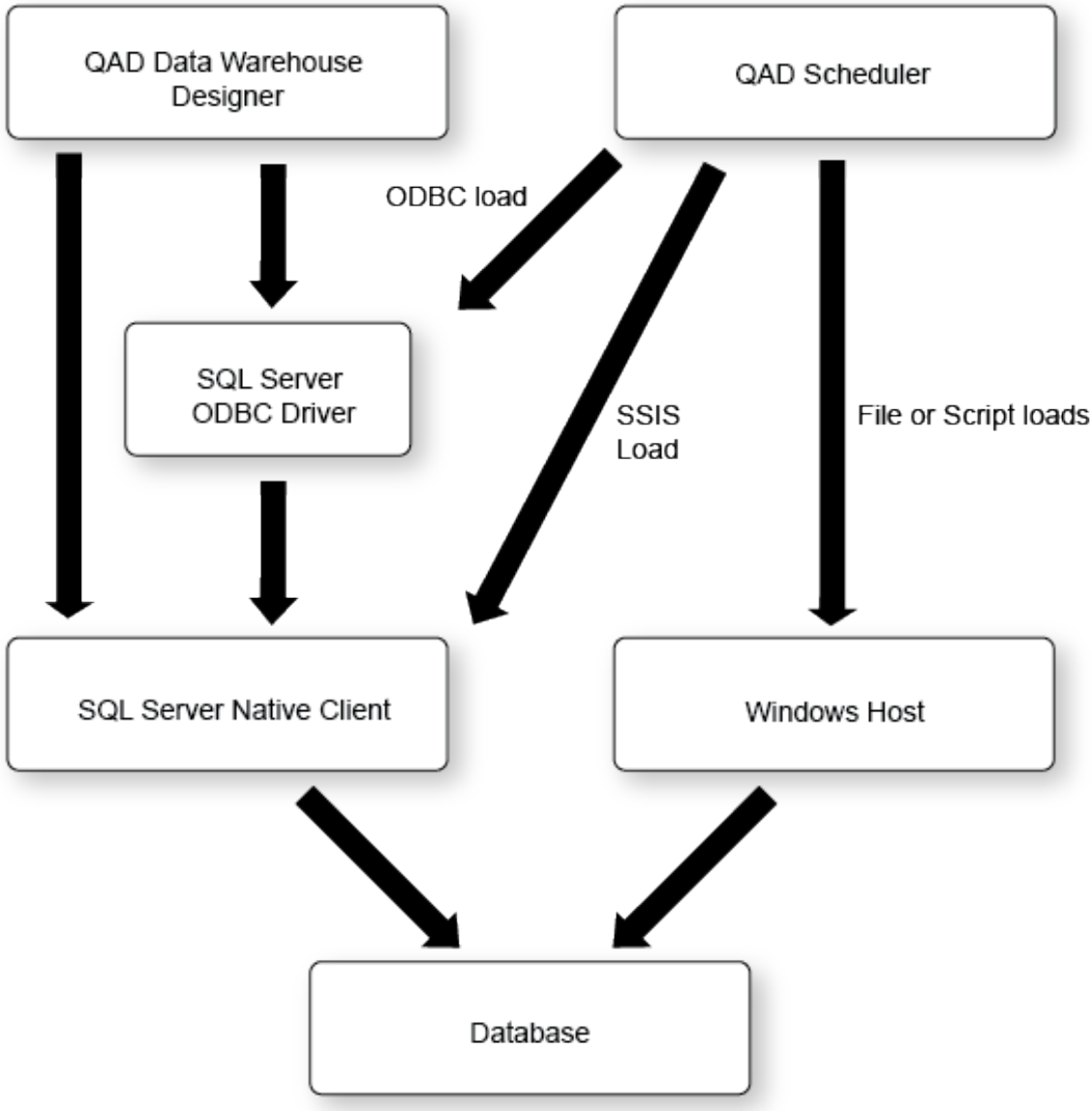
Install Windows Scheduler

This option deals with installing and starting a copy of a QAD Windows scheduler. We have divided this topic up into four sections; one for each of the four databases respectively:

- ***SQL Server Database*** (on page 48)
- ***Teradata Database*** (on page 52)
- ***Oracle Database*** (on page 56)
- ***DB2 Database*** (on page 62)

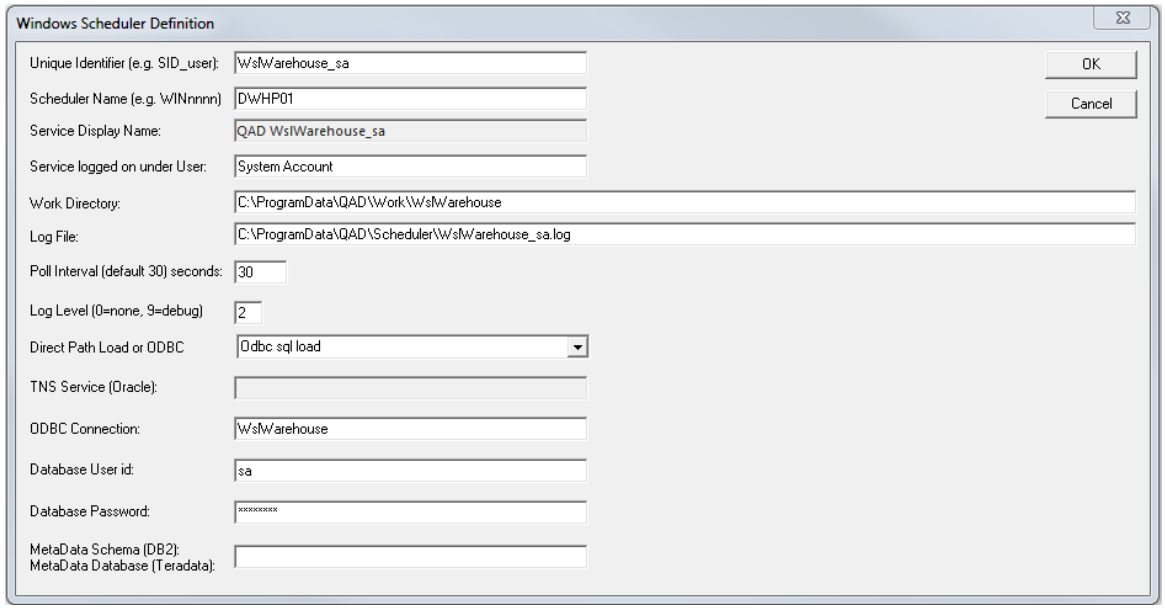
SQL Server Database

This option installs and starts a copy of the QAD Windows scheduler. The scheduler runs as a system service and can also be maintained through the normal Windows services functions. The following diagram provides an overview of how QAD Data Warehouse Designer and the scheduler access a SQL Server database.



The scheduler makes use of the ODBC layer to perform the routine tasks involved in scheduling and running jobs. QAD Data Warehouse Designer and the scheduler can also use SSIS to load data into SQL Server. For file or script based loads, the scheduler actions the load on the host system utilizing BULK INSERT.

On selection of this menu option the following dialog box will appear:



Unique Identifier

Each scheduler requires a unique name (possibly the database name and user e.g. WslWarehouse_sa). This unique name is the name by which the service is referenced.

Scheduler Name

When scheduling a job with QAD Data Warehouse Designer it is possible to assign the job to a specific scheduler. Each scheduler should therefore have a unique name. The default name is WIN0001. Most users change this to be the scheduler machine's name followed by a number, e.g.: DWHP01, where DWHP is the name of the windows scheduler server.

Service Display Name

This field displays the name that the scheduler will appear under when viewed through the Windows services. It is a combination of the string 'QAD' and the unique identifier.

Service Logged on Under User

This is the name of the Windows local user account that the scheduler service will use when logging on. The default setting is "System Account", also known as the "Local System" service account. To set the scheduler to use a Windows domain user account with a password, use the Service Account option from the right-click menu. If a domain user account is specified then the contents of this field (on the "Windows Scheduler Definition" dialog) are ignored.

Note: A domain user account will require the "log on as a service" privilege in order for it to work. This is assigned automatically when setting the log on account using the Services console in Windows.

Work Directory

The directory used to store some temporary files created by the scheduler.


TIP: Two schedulers should not use the same work directory

Log File

The full path where the scheduler service will write a log of the various actions undertaken. This file may get large if log level 9 is set.

Poll Interval

This defines the frequency of the scheduler polling the database, looking for work. The default is 30 seconds. If set to a very low interval such as 1 then the impact on the database may be noticeable.

Log Level

This defines the level of detail written to the log file. The following logging levels are available:

- 0 = will result in no information being written.
- 1 = fatal errors only
- 2 = major events
- 3 = major events and return stats to caller
- 9 = full debug


TIP: Setting Log Level 3 will return timing stats during ODBC based loads

Direct Path load or Odbc load

This should always be ODBC sql load for SQL Server schedulers.

TNS Service

This is not set for SQL Server schedulers.

ODBC Connection

The ODBC connection field requires the ODBC source name that will provide a connection to the data warehouse system. The scheduler uses ODBC to connect to the data warehouse.

Database User Id

For Windows Authentication, leave this field empty. For SQL Server Authentication, enter a database username.

Database Password

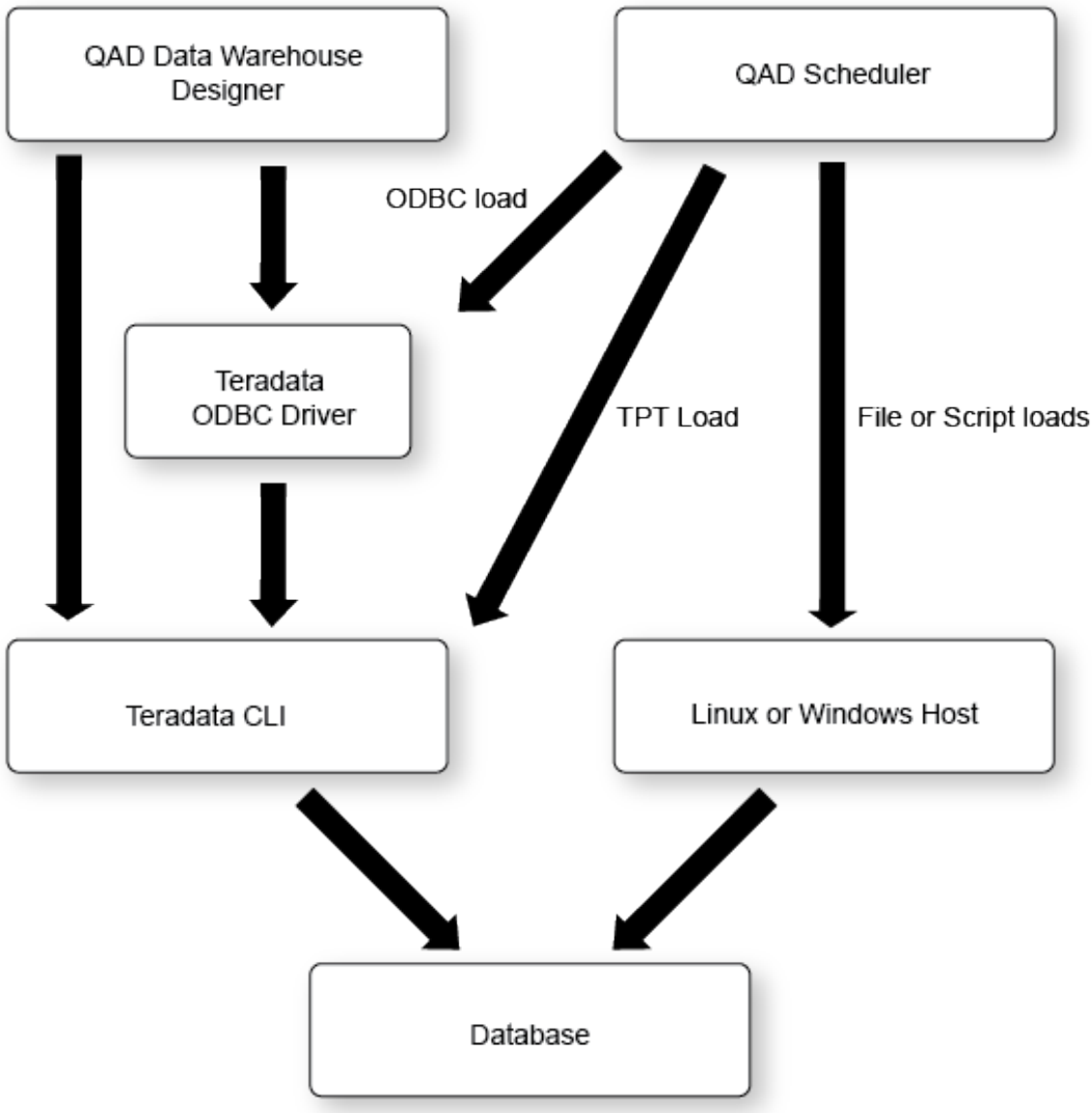
For Windows Authentication, leave this field empty. For SQL Server Authentication, enter the password for the database username entered in the Database User Id field.

Metadata Schema (DB2) / Metadata Database (Teradata)

This is not set for SQL Server schedulers.

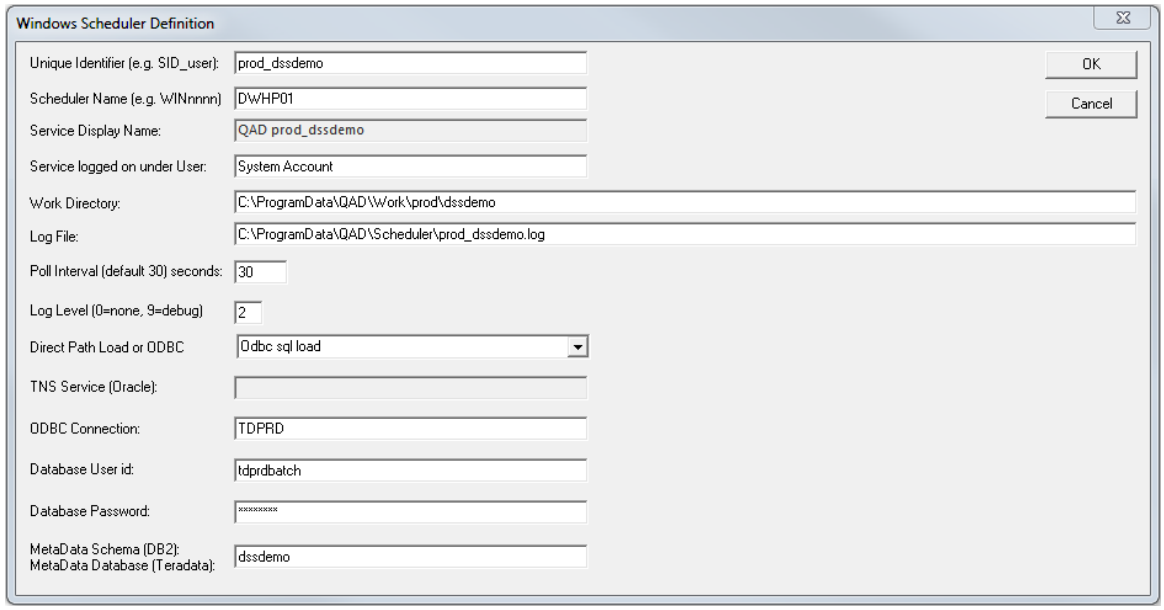
Teradata Database

This option installs and starts a copy of the QAD Windows scheduler. The scheduler runs as a system service and can also be maintained through the normal Windows services functions. The following diagram provides an overview of how QAD Data Warehouse Designer and the scheduler access a Teradata database.



The scheduler makes use of the ODBC layer to perform the routine tasks involoved in scheduling and running jobs. QAD Data Warehouse Designer and the scheduler can also use the TPT ODBC operator to load data into Teradata. For file or script based loads, the scheduler actions the load on the host system utilizing fastload, multiload and TPT (load and update operators).

On selection of this menu option the following dialog box will appear:



Unique Identifier

Each scheduler requires a unique name (possibly the environment name and metadata owner e.g. prod_dssdemo). This unique name is the name by which the service is referenced.

Scheduler Name

When scheduling a job with QAD Data Warehouse Designer it is possible to assign the job to a specific scheduler. Each scheduler should therefore have a unique name. The default name is WIN0001. Most users change this to be the scheduler machine's name followed by a number, e.g.: DWHP01, where DWHP is the name of the windows scheduler server.

Service Display Name

This field displays the name that the scheduler will appear under when viewed through the Windows services. It is a combination of the string 'QAD' and the unique identifier.

Service Logged on Under User

This is the name of the Windows local user account that the scheduler service will use when logging on. The default setting is "System Account", also known as the "Local System" service account. To set the scheduler to use a Windows domain user account with a password, use the Service Account option from the right-click menu. If a domain user account is specified then the contents of this field (on the "Windows Scheduler Definition" dialog) are ignored.

Note: A domain user account will require the "log on as a service" privilege in order for it to work. This is assigned automatically when setting the log on account using the Services console in Windows.

Work Directory

The directory used to store some temporary files created by the scheduler.


TIP: Two schedulers should not use the same work directory

Log File

The full path where the scheduler service will write a log of the various actions undertaken. This file may get large if log level 9 is set.

Poll Interval

This defines the frequency of the scheduler polling the database, looking for work. The default is 30 seconds. If set to a very low interval such as 1 then the impact on the database may be noticeable.

Log Level

This defines the level of detail written to the log file. The following logging levels are available:

- 0 = will result in no information being written.
- 1 = fatal errors only
- 2 = major events
- 3 = major events and return stats to caller
- 9 = full debug

 TIP: Setting Log Level 3 will return timing stats during ODBC based loads

 TIP: Setting Log Level 9 will create an additional detailed log file for each thread of each job that is run

Direct Path load or Odbc load

This should always be ODBC sql load for Teradata schedulers.

TNS Service

This is not set for Teradata schedulers.

ODBC Connection

The ODBC connection field requires the ODBC source name that will provide a connection to the data warehouse system. The scheduler uses ODBC to connect to the data warehouse.

Database User Id

The User name the Scheduler will use when connecting to the data warehouse Teradata database.

Database Password

The password for the User the Scheduler will use when connecting to the data warehouse Teradata database.

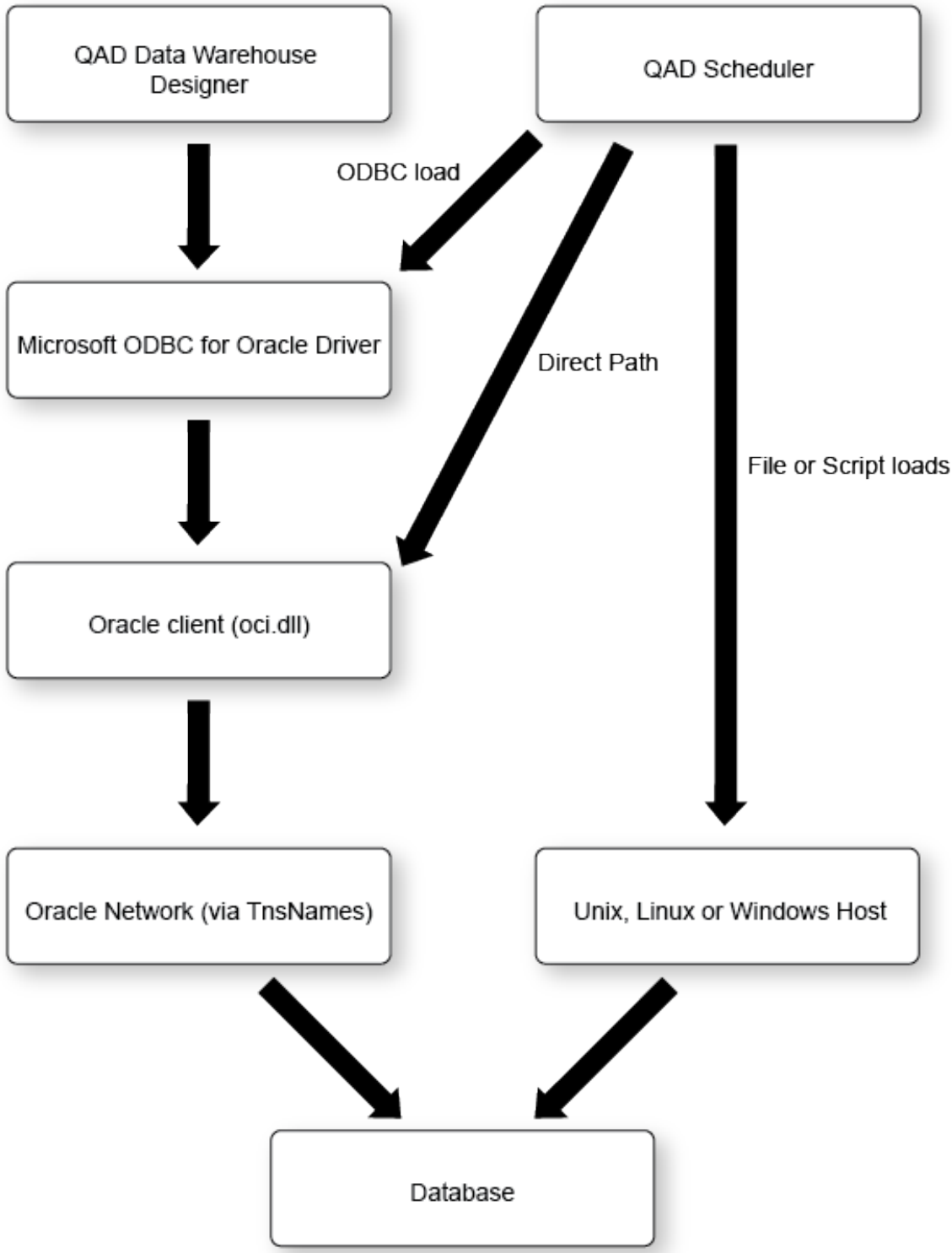
Metadata Schema (DB2) / Metadata Database (Teradata)

The Database Name or User Name where the metadata repository is stored. This must be entered for Teradata.

Oracle Database

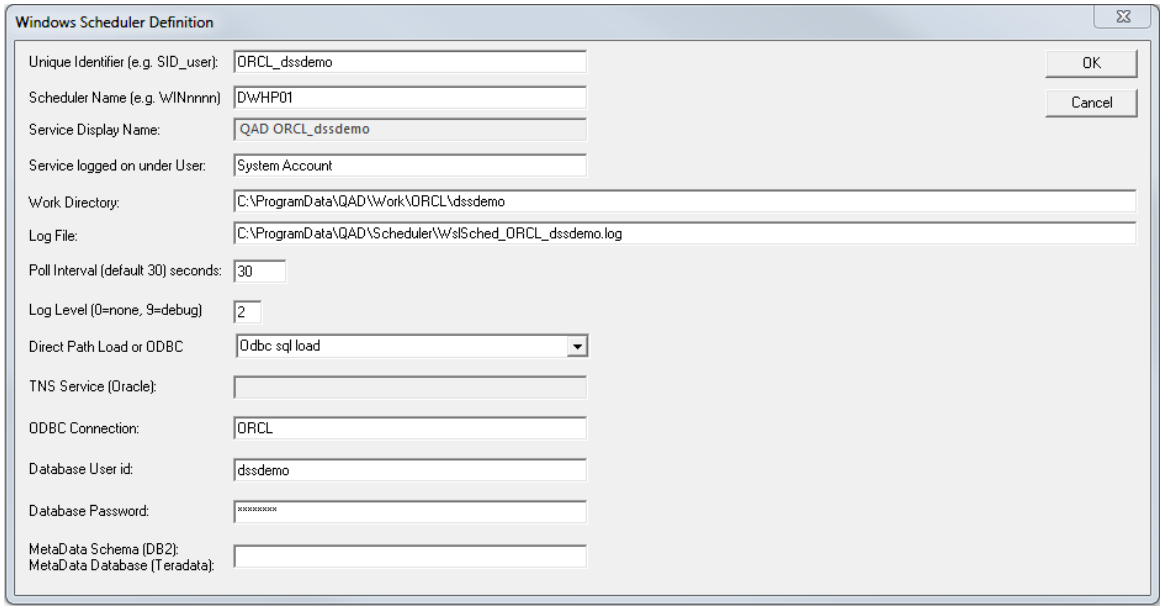
Note: The QAD windows scheduler for Oracle requires the Oracle client 8.1.7 or later software. The Oracle client software should be a close version match to the Oracle Database version, that is the same major release or one release either side.

This option installs and starts a copy of the QAD Windows scheduler. The scheduler runs as a system service and can also be maintained through the normal Windows services functions. The following diagram provides an overview of how QAD Data Warehouse Designer and the scheduler access an Oracle database.


QAD Data Warehouse Designer
QAD Scheduler
ODBC load
Microsoft ODBC for Oracle Driver
Direct Path
File or Script loads
Oracle client (oci.dll)
Oracle Network (via TnsNames)
Unix, Linux or Windows Host
Database

The scheduler makes use of the ODBC layer to perform the routine tasks involved in scheduling and running jobs. When loading data from an ODBC source the scheduler can either use the ODBC layer or the Oracle direct path load facility. For file or script based loads, the scheduler actions the load on the host system utilizing Oracle SQL* Loader.

On selection of this menu option the following dialog box will appear:



Unique Identifier

Each scheduler requires a unique name (possibly the SID and user e.g. ORCL_dssdemo). This unique name is the name by which the service is referenced.

Scheduler Name

When scheduling a job with QAD Data Warehouse Designer it is possible to assign the job to a specific scheduler. Each scheduler should therefore have a unique name. The default name is WIN0001. Most users change this to be the scheduler machine's name followed by a number, e.g.: DWHP01, where DWHP is the name of the windows scheduler server.

Service Display Name

This field displays the name that the scheduler will appear under when viewed through the Windows services. It is a combination of the string 'QAD' and the unique identifier.

Service Logged on Under User

This is the name of the Windows local user account that the scheduler service will use when logging on. The default setting is "System Account", also known as the "Local System" service account. To set the scheduler to use a Windows domain user account with a password, use the Service Account option from the right-click menu. If a domain user account is specified then the contents of this field (on the "Windows Scheduler Definition" dialog) are ignored.

Note: A domain user account will require the "log on as a service" privilege in order for it to work. This is assigned automatically when setting the log on account using the Services console in Windows.

Work Directory

The directory used to store some temporary files created by the scheduler.


TIP: Two schedulers should not use the same work directory

Log File

The full path where the scheduler service will write a log of the various actions undertaken. This file may get large if log level 9 is set.

Poll Interval

This defines the frequency of the scheduler polling the database, looking for work. The default is 30 seconds. If set to a very low interval such as 1 then the impact on the database may be noticeable.

Log Level

This defines the level of detail written to the log file. The following logging levels are available:

- 0 = will result in no information being written.
- 1 = fatal errors only
- 2 = major events
- 3 = major events and return stats to caller
- 9 = full debug


TIP: Setting Log Level 3 will return timing stats during ODBC based loads

A common Oracle error is NA000, object invalid which implies that the Ws_Job_Exec_44 procedure was not compiled successfully during the latest install or upgrade, or that the procedure has become invalid, due to a change in one of the QAD meta tables or procedures. Validate the metadata to ensure that this procedure has been compiled.

Direct Path load or Odbc load

This choice is only relevant for ODBC based data loads. When ODBC loads write to the data warehouse database two options are available. These are Oracle's direct path loading or an ODBC SQL statement. Direct path loads will be considerably faster and should be used in most situations. Normally if a direct path load can not be undertaken the scheduler will revert to an ODBC based load. This choice is primarily offered as a means of disabling direct path loads in the event of problems. Note that direct path loads have strict date format requirements. See the 'Loading Data' chapter in the QAD Data Warehouse Designer User Guide.

TNS Service

This is the database server as defined in the tnsnames table. It is used for direct path loads and must be a valid tnsnames entry for the data warehouse database. Normally this will be the same name as entered for the 'server' in an ODBC source definition.

ODBC Connection

The ODBC connection field requires the ODBC source name that will provide a connection to the data warehouse system. The scheduler uses ODBC to connect to the data warehouse.

Database User Id

The User Name the Scheduler will use when connecting to the data warehouse database.

Database Password

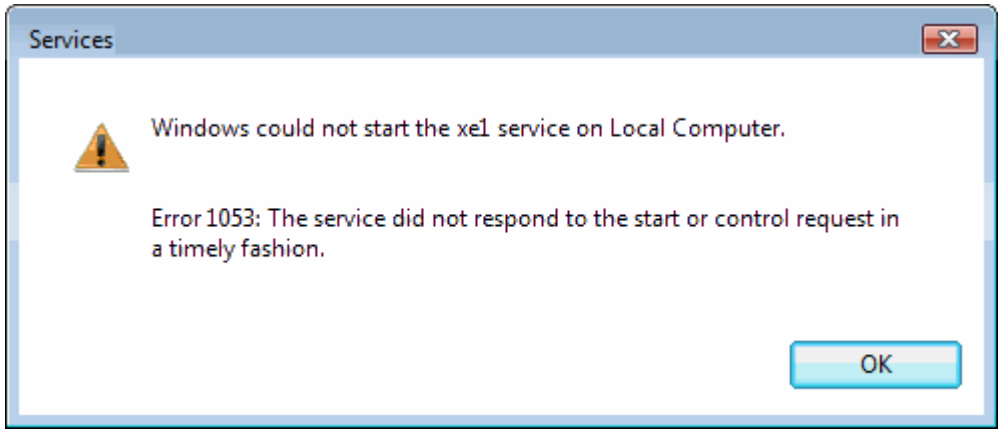
The password for the User the Scheduler will use when connecting to the data warehouse database.

Metadata Schema (DB2) / Metadata Database (Teradata)

This is not set for Oracle schedulers.

Oracle Error Messages:

The following error message may appear as the scheduler starts. In such an event the scheduler will not function until the problem is resolved.

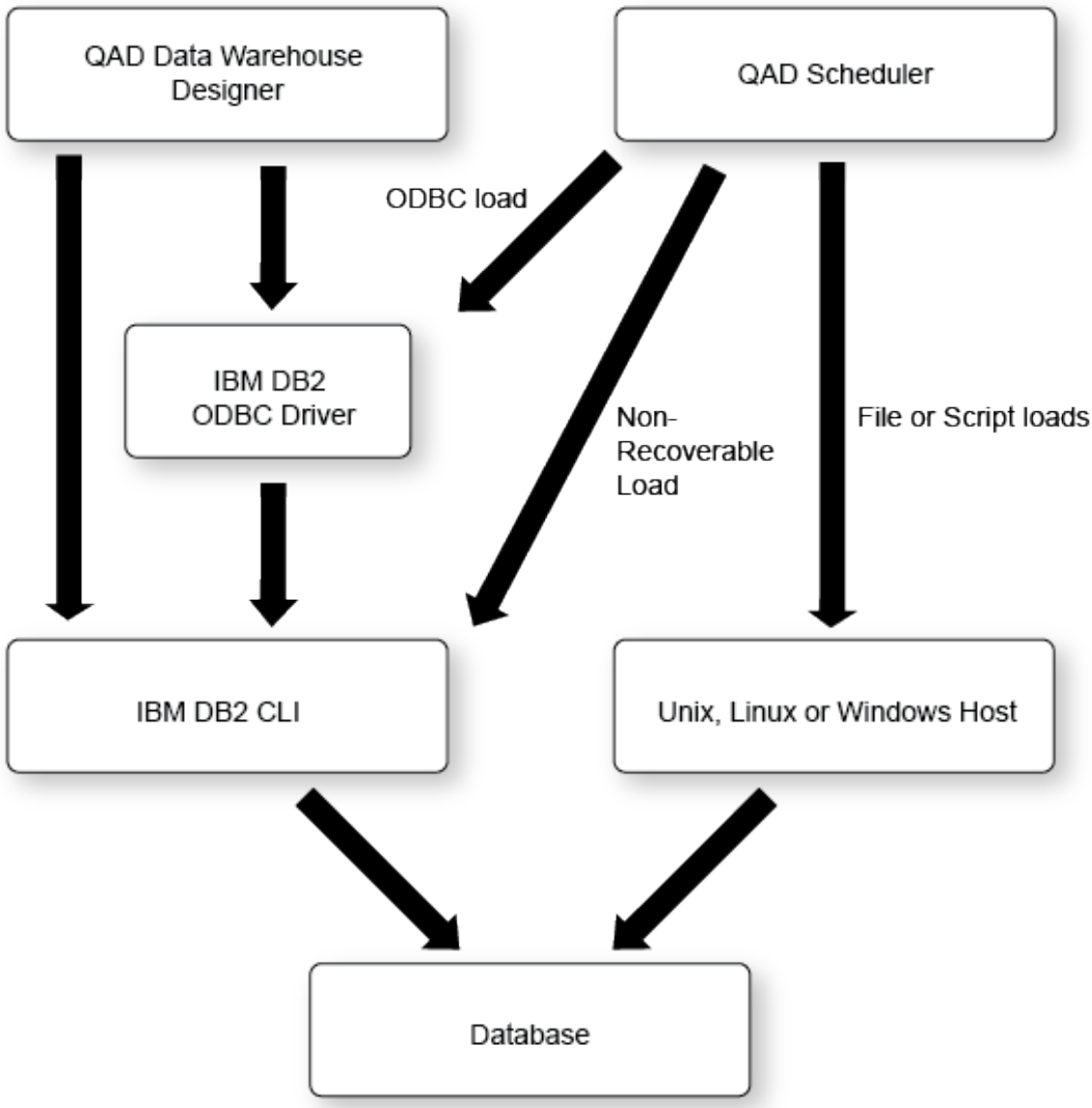


This error may be the result of an old version of the Oracle Call Interface (OCI) dynamic link library (dll) being in the path. The QAD scheduler requires Oracle client 8.1.7 or above. This situation can occur when an Oracle product such as Discoverer is installed. Discoverer may install an older version of the oci.dll file. If it does, its location will be put in the Windows path in front of the 8.1.7+ client version.

A work-around is to copy an 8.1.7 or later version of the oci.dll file into the QAD program directory. To see if an oci.dll is valid, right click on the file and select properties and then version. A valid oci.dll will have version 8.1.7 or later.

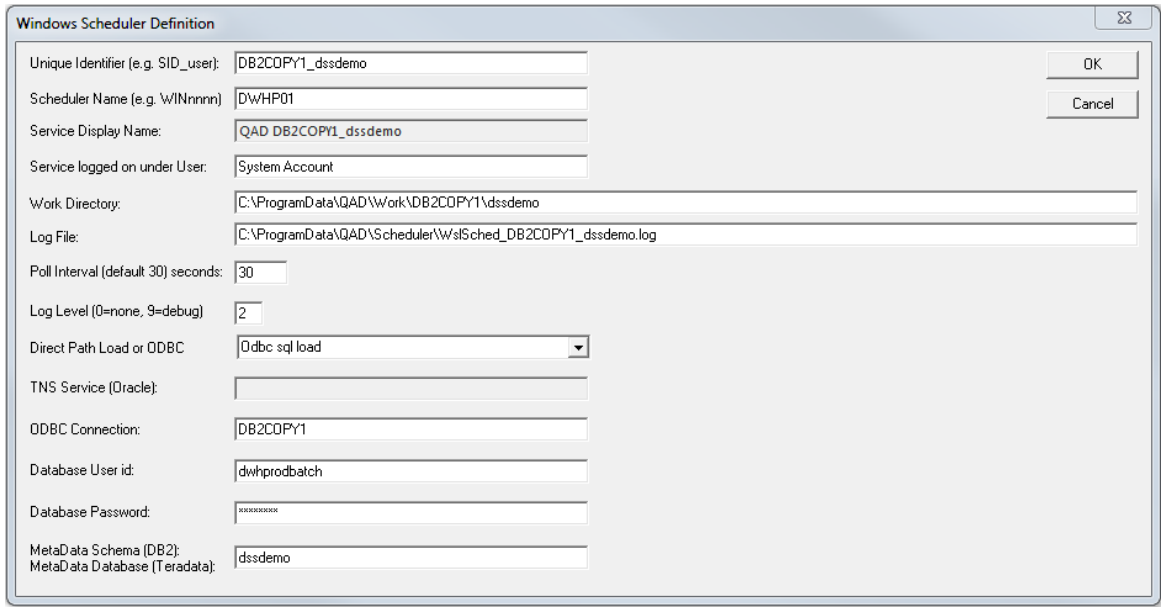
DB2 Database

This option installs and starts a copy of the QAD Windows scheduler. The scheduler runs as a system service and can also be maintained through the normal Windows services functions. The following diagram provides an overview of how QAD Data Warehouse Designer and the scheduler access a DB2 database.



The scheduler makes use of the ODBC layer to perform the routine tasks involved in scheduling and running jobs. QAD Data Warehouse Designer and the scheduler can also use CLI non-recoverable loads to load data into DB2. For file or script based loads, the scheduler actions the load on the host system utilizing the DB2 Load statement.

On selection of this menu option the following dialog box will appear:



Unique Identifier

Each scheduler requires a unique name (possibly the instance name and metadata schema e.g. DB2COPY1_dssdemo). This unique name is the name by which the service is referenced.

Scheduler Name

When scheduling a job with QAD Data Warehouse Designer it is possible to assign the job to a specific scheduler. Each scheduler should therefore have a unique name. The default name is WIN0001. Most users change this to be the scheduler machine's name followed by a number, e.g.: DWHP01, where DWHP is the name of the windows scheduler server.

Service Display Name

This field displays the name that the scheduler will appear under when viewed through the Windows services. It is a combination of the string 'QAD' and the unique identifier.

Service Logged on Under User

This is the name of the Windows local user account that the scheduler service will use when logging on. The default setting is "System Account", also known as the "Local System" service account. To set the scheduler to use a Windows domain user account with a password, use the Service Account option from the right-click menu. If a domain user account is specified then the contents of this field (on the "Windows Scheduler Definition" dialog) are ignored.

Note: A domain user account will require the "log on as a service" privilege in order for it to work. This is assigned automatically when setting the log on account using the Services console in Windows.

Work Directory

The directory used to store some temporary files created by the scheduler.


TIP: Two schedulers should not use the same work directory

Log File

The full path where the scheduler service will write a log of the various actions undertaken. This file may get large if log level 9 is set.

Poll Interval

This defines the frequency of the scheduler polling the database, looking for work. The default is 30 seconds. If set to a very low interval such as 1 then the impact on the database may be noticeable.

Log Level

This defines the level of detail written to the log file. The following logging levels are available:

- 0 = will result in no information being written.
- 1 = fatal errors only
- 2 = major events
- 3 = major events and return stats to caller
- 9 = full debug

TIP: Setting Log Level 3 will return timing stats during ODBC based loads

Direct Path load or Odbc load

This should always be ODBC sql load for DB2 schedulers.

TNS Service

This is not set for DB2 schedulers.

ODBC Connection

The ODBC connection field requires the ODBC source name that will provide a connection to the data warehouse system. The scheduler uses ODBC to connect to the data warehouse.

Database User Id

The User Id the Scheduler will use when connecting to the data warehouse database.

Database Password

The password for the User the Scheduler will use when connecting to the data warehouse database.

Metadata Schema (DB2) / Metadata Database (Teradata)

The Schema in DB2 where the metadata repository is stored. This must be entered for DB2.

DB2 Grants

From DB2 9.7, additional grants are required to run the QAD Data Warehouse Designer Windows Scheduler as the system service account in Windows. See Meta Creation - IBM DB2 for more details.

Install Unix Scheduler

Note: Oracle, IBM DB2 and Teradata data warehouse only. There is no Unix Scheduler for SQL Server.

The Unix scheduler must be manually installed and maintained.

The scheduler consists of a series of shell scripts that are controlled by the cron process. The bulk of the work undertaken by the QAD scheduler takes place in the QAD procedures within the database. To obviate the need to provide a port to each Unix platform this series of shell scripts are used. They should run with no modification on all Unix platforms and have been tested under Unix offerings from Sun, HP, IBM, Compaq, SGI and Linux.

Installation

The installation of the Unix scheduler needs to be undertaken by someone familiar with the Unix environment and the running of database processes from Unix. The tasks that need to be undertaken are:

1. Unix User and Shell

Acquire a Unix user name that has access to cron and the database data warehouse environment. A reasonable quantity of disk space is also required if the automated backups of the metadata repository are used. The user should log on to Unix without passing through a menu system if possible. Log on to the user.

The QAD scheduler consists of a number of shell scripts. By default some of these shell scripts will run in the Bourne shell and some will run in the Korn shell. If Linux is being used then the Bash shell is used exclusively.

Note: During setup it is recommended that the Korn shell be used as the default shell for the user installing the scheduler (or Bash if running Linux).

2. Create directories

Create the following directories (if not already present) to be used by the QAD scheduler. If the directory names are not appropriate and others must be used then it will be necessary to alter the QAD supplied scripts.

- wsl (top level directory for QAD directories)
- wsl/bin (used to store the QAD scripts)
- wsl/sched (used for the scheduler job and log files)
- wsl/sched/job (used for the scheduler job files)
- wsl/sched/joblog (used for the scheduler job logs)
- wsl/sched/log (used for the scheduler logs)

- wsl/export (used for the automated exports of the metadata)
- wsl/expback (used to hold compressed copies of the metadata exports)
- wsl/log (used for export and archive logs)
- wsl/mon (used for the monitoring process)
- wsl/mon/db (used for database monitoring scripts and control files)
- wsl/mon/job (used for job monitoring scripts)
- wsl/mon/log (used for monitoring log files)

3. Copy supplied Unix files

FTP the scheduler files shipped with QAD Data Warehouse Designer to the wsl/bin directory on the Unix system (use ASCII mode). All files should be copied over.

QAD Data Warehouse Designer ships with several different versions of the unix files for the scheduler. The appropriate version of the files should be copied to the server. The following lists the files shipped:

- ...\QAD\Oracle\Unix\Version_nnn\...
- ...\QAD\Oracle\Linux\Version_nnn\...
- ...\QAD\DB2\Unix\Version_nnn\...
- ...\QAD\DB2\Linux\Version_nnn\...
- ...\QAD\Teradata\Linux\Version_nnn\...

Note: There may be several different versions of the scheduler files for a given database (Oracle, DB2 or Teradata) and platform (Unix or Linux). These are identified by version numbers nnn. For example, there may be three different folders in ...\QAD\Oracle\Linux\: Version_010, Version_410 and Version_560. The highest version number script less than or equal to the version of Data Warehouse Designer in use should always be used. In the table that follows, the nnn in file names refers to the version number of the script. Some scripts will have 010, some will have 411 and other 560, etc.

A definition of the files follows:

File	Purpose
oraenv (for Oracle) dbenv (for DB2) tdenv (for Teradata)	Contains the template of a script for setting environmental variables. This needs to reside in the home directory of the Unix user and must be tailored for the database user and environment. It may be renamed to reflect the actual environment. (e.g. oraenv_SID_user)
crontab	Contains the template of the required cron commands. This file needs to be modified to include the HOME directory of the Unix user, and the name of the environment file. It is then used to update the crontab tables.
ws_start_nnn.sh	Script to start the scheduler. This script is normally called by the ws_sched_check_nnn.sh script and is passed the database environment file name as a parameter.

ws_sched_check_nnn.sh	Script to check that the scheduler is running. This script is normally executed by cron every 20 minutes. If the scheduler is found to be absent it attempts to start a new iteration. By default the scheduler will terminate if its database is shut down. This script will therefore start the scheduler after a system start-up or a database start-up.
ws_sched_nnn.sh	The actual scheduler script. This script looks for new scheduler jobs every x seconds. The default is every 30 seconds. This script must be modified to change the default behavior. It calls the ws_job_wait_010.sh script to perform the actual checking and initiating of a job. It terminates when ws_job_wait_010.sh returns an error, and relies upon cron for a restart.
ws_sched_status_nnn.sh	This script is called by ws_sched_nnn.sh to update the scheduler status in the scheduler metadata. It's a wrapper of the metadata database procedure ws_sched_status (i.e. it calls the database procedure ws_sched_status and returns it's results to the calling script).
ws_job_wait_nnn.sh	This script performs the bulk of the work. It is called by ws_sched_nnn.sh on a regular basis. It checks for any job waiting to run. If it locates a job that is ready to run it will create and start a script in the $HOME/wsl/sched/job directory for each thread of that job. These job threads in turn perform the actual processing and log their results to $HOME/wsl/sched/joblog. Under normal operation it is not necessary to check or inspect these files.
ws_job_check_nnn.sh	This script is called to check the status of running jobs. It is called by ws_sched_nnn.sh on a regular basis (the interval is defined in ws_sched_nnn.sh). It checks that any job running in the scheduler is actually running in the Unix environment. If a job is not found under Unix then it is aborted in the scheduler. This scenario can occur as the result of a database or machine crash. If a job abort is required the results are logged to $HOME/wsl/sched/log/job_check_'env file name'.log. Under normal operation it is not necessary to check or inspect this file.
meta_backup_nnn.sh	This script is normally called by the cron process once or maybe more times a day to perform an automated export backup of the metadata tables. It creates a number of command files and a database export in the $HOME/wsl/export directory named with the date/time of the export.
meta_archive_nnn.sh	This script is normally called by cron once a day to compress and tar the files in the $HOME/wsl/export directory and relocate them to the $HOME/wsl/expback directory. This action is performed purely to save on disk space usage.

meta_restore_nnn.sh	This script can be executed to restore from a meta repository backup. It will overwrite any existing metadata. It can be used to restore to a schema/database/user other than the one from which the backup was taken. It does not however restore the QAD procedures. If a full recovery or a new user create is required then the QAD metadata install would normally be performed before actioning this restore. If the restore is from an export that has been moved to the expback directory you will need to uncompress the export and move it and associated files back to the export directory.
cleanup_nnn.sh	This script is normally executed by cron and cleans up files created by the QAD scheduler. It deletes old versions of the jobs, job logs, and backups. It will also delete any temporary scripts created in the running of host scripts or file loads from within the scheduler. The location of these temporary scripts is defined within the QAD Data Warehouse Designer Unix Connection object. The default value in the cleanup script is /tmp and this must be alteData Warehouse Designer if a different temporary directory is used for a Connection.
ws_mon_nnn.sh	The main monitoring script. This script in turn calls the database and job monitoring scripts to perform the monitoring. By default the monitor process checks the jobs every 15 minutes and the database every 60 minutes. This script can be edited to change the frequency of the checks if required. Also a log level of 9 can be set to produce debug logging.
ws_mon_db_nnn.sh	This script checks the database status and if necessary calls the notify script to notify of a database down. Also checks to see if the criteria for database monitoring has changed and if changed calls the refresh script.
ws_mon_notify_nnn.sh	This script checks that notifications are valid for the day/time and if valid sends a notification.
ws_mon_refresh_nnn.sh	This script refreshes the scripts and command files stored in the mon/db directory. These scripts and command files are used to ascertain the action should the database be unavailable.
ws_mon_jobs_nnn.sh	This script performs the checking of jobs for which monitoring is defined. If a job has a notification required then this script will issue the notification.
ws_mon_start_nnn.sh	This script starts the monitor process
ws_mon_check_nnn.sh	This script checks that the monitor process is running. If not running the script will start the monitor process. This script is normally called via cron every 30 minutes.

4. Set permissions

Change the permissions on the copied scripts so that they are executable. (e.g. chmod 750 $HOME/wsl/bin/*.sh)

5. Create an environment file

Using the environment file (oraenv, dbenv or tdenv) as a template and adhering to the variable names within, create an environment file for the database environment and user under which the scheduler is to run. This file should be located in the $HOME directory. It may be called anything but a meaningful name may be oraenv_SID_user for a given Oracle scheduler. Check the users .profile and database environment file to see the normal settings. For oracle, check to see if the TNS_ADMIN variable is in use and if so, include it in the environment file.

For all environements make the following replacements in the environment file:

- Replace all occurances of *_HOME_* with the full pathname of your home directory
- Replace *_USER_* with the data warehouse database user name
- Replace *_PASSWORD_* with the data warehouse database user password.

In addition, for Oracle:

- Replace all occurances of *_SID_* with the sid of the data warehouse database
- Replace all occurances of *_ORAHOME_* with the full pathname of the Oracle home directory

In addition, for DB2:

- Replace all occurances of *_DBID_* with the database id of the data warehouse database
- Replace all occurances of *_SCHEMA_* with the schema where the metadata resides

In addition, for Teradata:

- Replace all occurances of *_DB_* with the host name of the data warehouse database server, used by the scheduler as the database server name for Teradata load and unload utilities
- Replace all occurances of *_BTEQDB_* with the host name or ip address of the data warehouse database server, used by the scheduler as the database server name/address for reading and writing QAD Data Warehouse Designer Metadata in Teradata (may be set to the same value as *_DB_*)
- Replace all occurances of *_METABASE_* with the user or database where the metadata resides

6. Setup crontab

Ascertain if there are any existing crontab entries, and if so save them to a file for re-inclusion later. (e.g. crontab –l crontab –l >/tmp/my_crontab)

Edit the crontab file and perform the following changes. Also check all entries and make sure you are happy with the times and actions:

- Change all occurrences of *_HOME_* with the full pathname of the Unix users home directory.

- Change all occurrences of the environment file name (oraenv, dbenv or tdenv) to the name of the file created in step (4).
- Include any existing crontab entries as recorded above.

7. Initialize crontab entries

Load the crontab file into cron (e.g. crontab crontab).

Perform a crontab list to ensure it all looks OK (e.g. crontab –l)

8. Start Scheduler

Manually start the scheduler by issuing the ws_sched_check_nnn.sh script with the environment file name from step (4) as the parameter (e.g. ws_sched_check_010.sh oraenv). If the scheduler is not running it will be started. Execute the script a few more times to ensure that only one copy of the scheduler is started. Check that the scheduler is running by executing a 'ps –ef' statement and checking that the ws_sched_560.sh process is present.

9. Start the Monitor process

Oracle only. Manually start the monitor by issuing the ws_mon_check_010.sh script with the environment file name from step (4) as the parameter (e.g. ws_mon_check_010.sh oraenv). If the monitor is not running it will be started. Execute the script a few more times to ensure that only one copy of the scheduler is started. Check that the monitor is running by executing a 'ps –ef' statement and checking that the ws_mon_010.sh process is present.

10. Test the Scheduler

Run a job from QAD Data Warehouse Designer and ensure that it works. In the event of the job not starting or going into and remaining in a pending state then the following investigations should be undertaken:

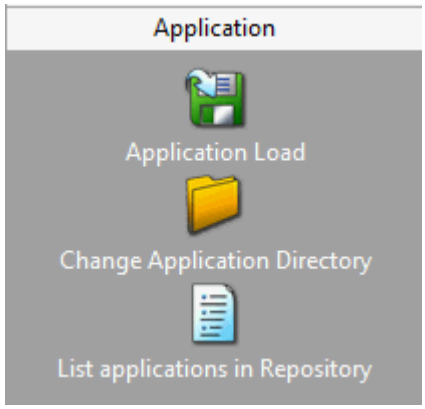
- Ensure all QAD procedures are compiled. Use the Validate metadata option in Setup Administrator to confirm that all procedures are valid.
- Review the job logs in $HOME/wsl/sched/joblog.
- Review the logs in $HOME/wsl/sched/log.
- Isolate the name of the job and locate the relevant files for this job in $HOME/wsl/sched/job and $HOME/wsl/sched/joblog if they exist. Mail these files together with the files in $HOME/wsl/sched/log to http://support.qad.com. It may be necessary to remove password references from the file(s) in $HOME/wsl/sched/job.

CHAPTER 9

List/Load Applications

The application component of the QAD 'Setup Administrator' attempts to list, upgrade or perform an initial install of a third-party supplied data warehouse application.

Click on the 'List/Load Application' icon to start. Click on any of the options under this level to action them. The application menu is as follows:



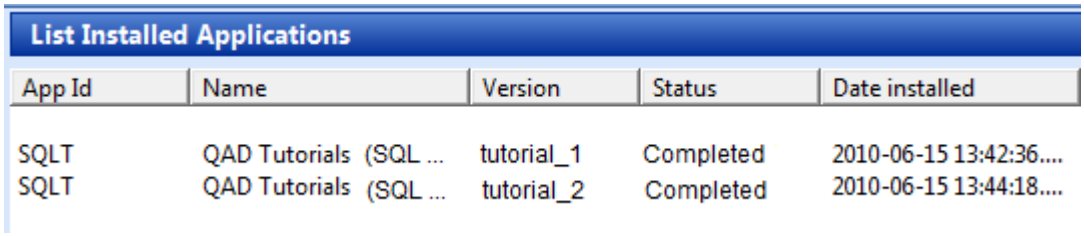
- The ***load application*** (on page 75) option loads the metadata, tables procedures, scripts etc. provided as part of the application.
- Change application directory options changes the directory of applications to be installed.
- The ***list applications*** (on page 74) option will list all applications that have been applied to the chosen metadata repository.

In This Chapter

List Applications

The list application menu option will prompt for a logon to the required metadata repository. Applications are applied to a specific metadata repository.

Once logged on a list of the applications that have been installed into that repository will be shown. If no applications have been loaded then the list will be empty. An example output is as follows:



List Installed Applications

App Id	Name	Version	Status	Date installed
SQLT	QAD Tutorials (SQL ...	tutorial_1	Completed	2010-06-15 13:42:36....
SQLT	QAD Tutorials (SQL ...	tutorial_2	Completed	2010-06-15 13:44:18....

This example shows that the tutorial_2 application has been loaded into the metadata.

Load Application

Loading an application inserts various objects into the chosen metadata repository. An application is best defined as a set of objects that are shipped to allow inclusion of those objects into a remote repository.

Note: An application can only be loaded into a metadata repository running on the same database type as that of the application creator. (e.g. An Oracle application cannot be loaded into a SQL Server metadata repository).

During the load of an application a number of cross environment mappings need to be resolved. These are:

Connections

Connections are normally unique for each metadata repository. They provide the path to the source data and this nearly always varies. Even though an application can ship a set of connections, these connections will nearly always need modification. Alternatively a connection shipped with an application may map to an existing connection, and this can be done as part of the load process. If you are unsure whether or not a connection is required, it can be added and later modified.

Tablespaces (Filegroups in SQL Server and Databases in Teradata)

If an application ships table or index objects then these objects will typically have a tablespace (filegroup in SQL Server and databases in Teradata) associated with them. During the load it will be necessary to map these tablespaces (filegroup in SQL Server and databases in Teradata) to those available for the metadata repository.

Object Changes

The application may ship objects that already exist in the metadata. In such a case the process during the load is to create a version of the existing object and then replace it with the new object. At the start of the load a check is made to see if any objects will be replaced. If the application is a patch or an upgrade of an existing application version then this will be normal. The load can be stopped at this point to allow for further investigation.

Table Changes

If the load replaces table objects, it may in turn need to alter the physical table in the database to match the new definition. Again, if this is a patch or an application upgrade, this may be desirable. If not, the proposed actions should be reviewed before proceeding.

Procedure Changes

New procedures, scripts, or modifications may be loaded as part of the application load. In the case of procedures these procedures will need to be recompiled. The load can perform this action, but any failures or issues will need to be resolved after the load has completed.

The following section covers the actual application load process.

Note: It is always a good idea to backup the QAD Data Warehouse Designer metadata before running an application load. See the section ***Backing Up the Metadata before Loading Applications*** for details.

Backing Up the Metadata before Loading Applications

It's always a good idea to backup the QAD Data Warehouse Designer metadata before running an application load.

A metadata backup can be used to restore the metadata to the state before the application was loaded.

There are three options for backing up the metadata:

1. In Data Warehouse Designer, use the Backup/Unload the metadata to disk option to unload the metadata to a QAD Proprietary backup. See the section Unloading Metadata in the QAD Data Warehouse Designer User Guide for more information.
2. In Data Warehouse Designer, use the Backup/Export the metadata option to backup the metadata using the data warehouse database's backup utility. See the section Backup using DB Routines in the QAD Data Warehouse Designer User Guide for more information.
3. In Data Warehouse Designer, use the Tools/Build application tables option to build an application of the objects being replaced. This is achieved by selecting the new application files you're about to load in the Previous application box of the Define and build an application distribution dialog. See the section Application Creation in the QAD Data Warehouse Designer User Guide for more information.

Note: If the table changes in the application are being applied to the data warehouse database, then it is also a good idea to backup the data warehouse tables being changed.

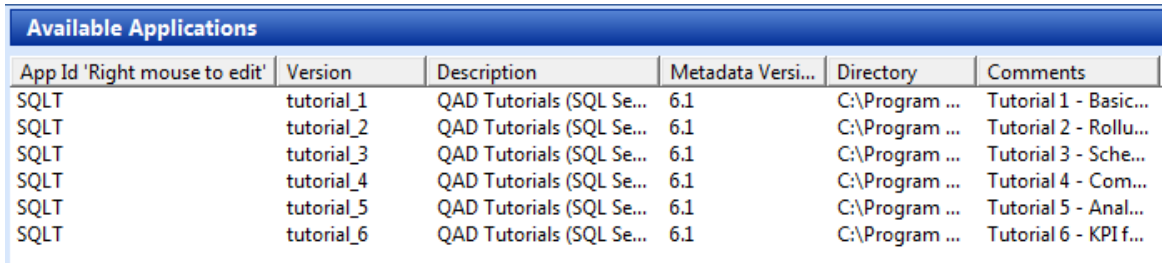
Application Load Process

An application consists of five files that are loaded from a directory. The default directory is the QAD program directory. Another directory may be specified by clicking on the 'Change Application Directory' button. The last selected application directory is the default directory for the next application load.

Once the directory has been selected, the application load is started by clicking on the 'Application Load' button. This displays the available applications in the right hand pane.

Note: If the 'Change Application Directory' button was used, then the list of available applications will already be displayed and the 'Application Load' button need not be clicked.

If no applications are present the list in the right hand pane will be empty and no application load is possible. The tutorial applications are nearly always present. An example application list follows:



Available Applications

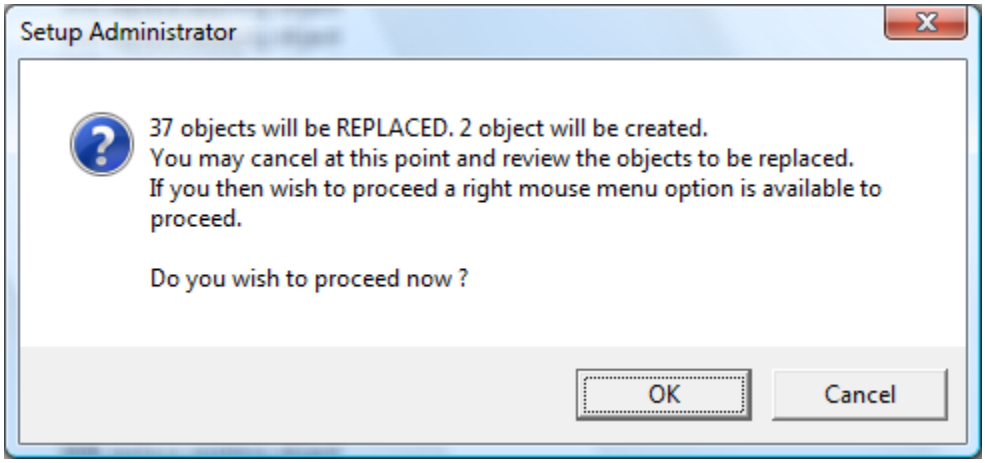
App Id 'Right mouse to edit'	Version	Description	Metadata Versi...	Directory	Comments
SQLT	tutorial_1	QAD Tutorials (SQL Se...	6.1	C:\Program ...	Tutorial 1 - Basic...
SQLT	tutorial_2	QAD Tutorials (SQL Se...	6.1	C:\Program ...	Tutorial 2 - Rollu...
SQLT	tutorial_3	QAD Tutorials (SQL Se...	6.1	C:\Program ...	Tutorial 3 - Sche...
SQLT	tutorial_4	QAD Tutorials (SQL Se...	6.1	C:\Program ...	Tutorial 4 - Com...
SQLT	tutorial_5	QAD Tutorials (SQL Se...	6.1	C:\Program ...	Tutorial 5 - Anal...
SQLT	tutorial_6	QAD Tutorials (SQL Se...	6.1	C:\Program ...	Tutorial 6 - KPI f...

To load an application use the right mouse when positioned on the Application identifier of the application and version you wish to load. A pop-up menu will appear with the option 'Install Application'.

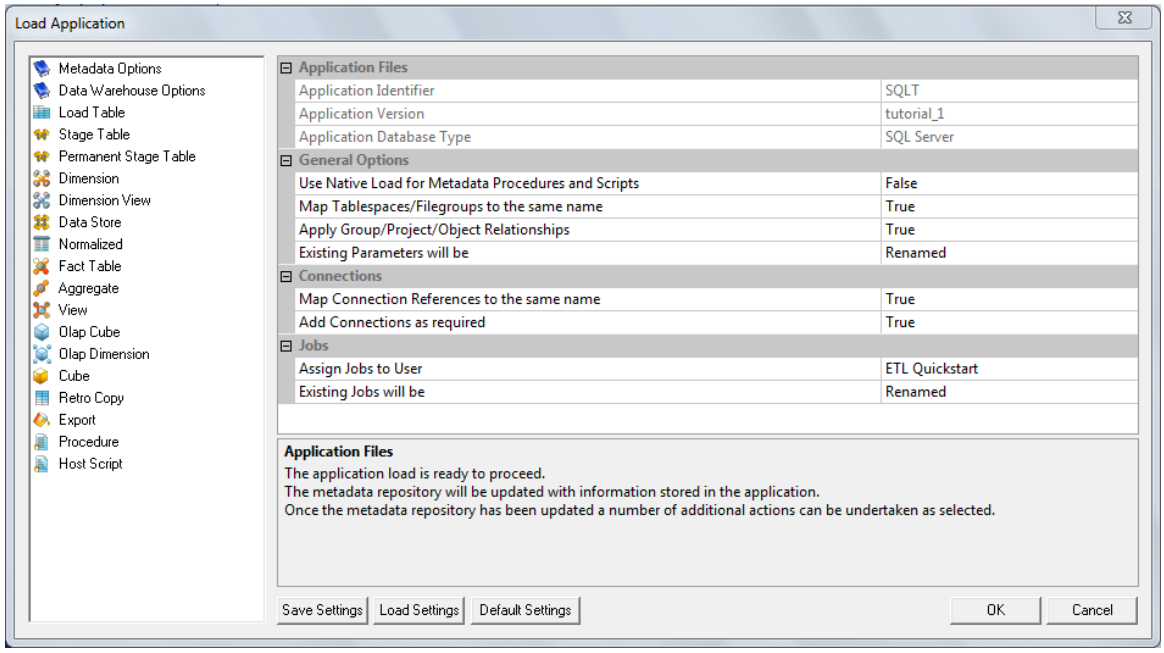
Once this menu option is chosen, enter the ODBC data source and logon to the meta repository and click OK.

Note: If the application is being loaded into a Teradata database, an additional dialog is displayed for selecting the metadata repository database. Select the database and click OK. If not loading into a Teradata database, this dialog is not displayed.

A list of all the objects in the application is displayed in the right hand pane. Each object will be identified as either new to the metadata, or as replacing an existing object. For example:



A dialog will appear after the list has been displayed advising of what is to occur. You may cancel the application load at this point and review the objects to be replaced. The load can then be continued by using the right mouse menu option when positioned over any object. Select the OK button to proceed with the load of the application. A new dialog will appear providing options for the application load:



If OK is selected the objects will be inserted into the metadata.

Metadata Options

Application Files	
Application Identifier	SQLT
Application Version	tutorial_1
Application Database Type	SQL Server
General Options	
Use Native Load for Metadata Procedures and Scripts	False
Map Tablespaces/Filegroups to the same name	True
Apply Group/Project/Object Relationships	True
Existing Parameters will be	Renamed
Connections	
Map Connection References to the same name	True
Add Connections as required	True
Jobs	
Assign Jobs to User	ETL Quickstart
Existing Jobs will be	Renamed

Use Native Load for Metadata Procedures and Scripts

This option causes the metadata for procedures and scripts to be loaded into the target metadata repository using the target repository's load utility instead of ODBC. This can have significant performance advantages when loading large applications, when loading applications into a slow target database or over a slow network. The loaders used are: SQL*Loader for Oracle, bcp for SQL Server and BTEQ for Teradata data warehouses.

Map Tablespace/Filegroups to the same name

Every table in Oracle and DB2 is located in a tablespace. SQL Server tables may be in filegroups. If this option is marked as true, the tablespaces and filegroups of objects will be set to the same name they had in the source repository if the tablespace/filegroup exists in the target data warehouse database. If the tablespace/filegroup does not exists in the the target data warehouse database, an alternative tablespace/filegroup must be selected using the Tablespace selection dialog. This dialog appear once for each tablespace/filegroup that cannot be matched.

Note: This option is not available for Teradata

Apply Group/Project/Object Relationships

If this option is marked as true, the project group relationships are updated in the target data warehouse database.

Existing Parameters will be

The options are

- Updated/Overwritten
- Retained
- Renamed

Map Connection References to the same name

Use this option to ensure that any objects with linked connections map to an existing connection using the same name as the connection that linked to in the metadata repository where the application was built. If a connection does not exist in the target metadata repository and the option *Add new connections as required by the application* is not selected, then an alternative connection must be selected using the Connection Build dialog. This dialog appears once for each tablespace/filegroup that cannot be matched.

Add Connections as required

If this option is marked as true, any connection objects included in the application that do not exist in the target metadata repository, that are used by other objects in the application, will be added to the target metadata repository.

Assign Jobs to User

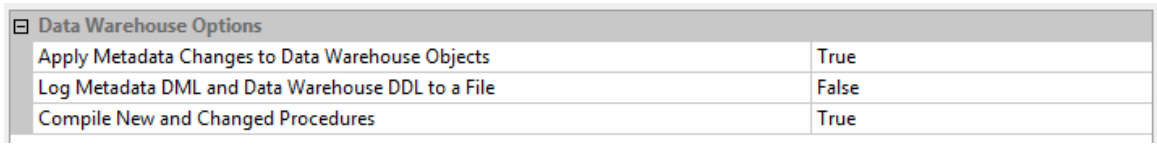
All jobs in the QAD scheduler belong to a user. If this option is marked as true, the user name specified in the *User name for any jobs added by the application* drop down is set as the user name of all jobs loaded by the application. If the user name specified does not exist, it's created.

Existing Jobs will be

The options are

- Updated/Overwritten
- Retained
- Renamed

Data Warehouse Options



Data Warehouse Options	
Apply Metadata Changes to Data Warehouse Objects	True
Log Metadata DML and Data Warehouse DDL to a File	False
Compile New and Changed Procedures	True

Apply Metadata Changes to Data Warehouse Objects

If this option is marked as true, the changes contained in the application are applied to the metadata according to the options selected for each object type.

Log Metadata DML and Data Warehouse DDL to a File

If this option is marked as true, the changes contained in the application are written to a file. If not then only the file is created.

Compile New and Changed Procedures

If this option is marked as true then all procedure objects loaded by the application load are compiled in the target data warehouse database.

Note: Depending on the type of database, invalid procedures may not compile.

Data Warehouse Table/View Options

Load Table	
New Load Tables will be	Created
New Load Tables will be Versioned	False
New Load Table Indexes will be	Created
Existing Load Tables will be	Recreated
Existing Load Tables will be Versioned	True
Existing Load Table Indexes will be	Recreated

For each table and view object loaded by the application load, it is possible to specify whether or not the metadata changes should be applied to the data warehouse table or view and to the indexes on these objects. The change action can be specified by object type. New objects can have a different action to existing objects of the same type. Possible actions are:

- Created - available for new tables and indexes of any object type
- Recreated - available for existing tables and indexes of any object type
- Altered - available for existing tables and indexes of most object types (not available for load tables or views)
- Nothing - available for all new and existing object types and indexes.

This dialog also enables you to set whether or not new or existing objects are to be versioned. Set to True to use versioning, else set to False.

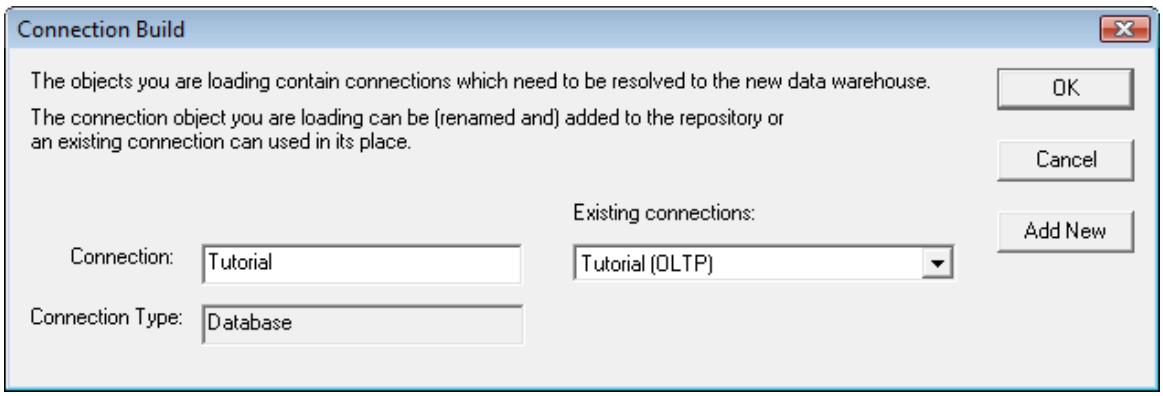
Stage Table	
New Stage Tables will be	Created
New Stage Tables will be Versioned	False
New Stage Table Indexes will be	Created
Existing Stage Tables will be	Recreated
Existing Stage Tables will be Versioned	True
Existing Stage Table Indexes will be	Recreated
Alter allows columns to be dropped	False

Another option enables you to set whether or not columns are allowed to be dropped by object type, for tables being altered. Set to True to allow columns that are not in the metadata, to be dropped; else set to False. This option applies to Stage, Permanent Stage, Dimension/Model, ODS, Normal, Fact and Aggregate Tables.

Object action settings may be saved and reloaded later for another application using the Save Settings and Load Settings buttons. Supplied default settings may also be loaded by clicking on the Default Settings button.

The Connection Build dialog

If connections are encountered and are not automatically mapped (see connection options above), a dialog is issued to ascertain whether the connection can be mapped to an existing connection or whether the connection needs to be added. The application vendor will need to advise of the action required. The dialog is as follows:



Either a connection is selected from the 'Existing connections' drop down list or the 'Add New' button is clicked. If a new connection is added then it will need to be modified to match the database and user names of the new environment.

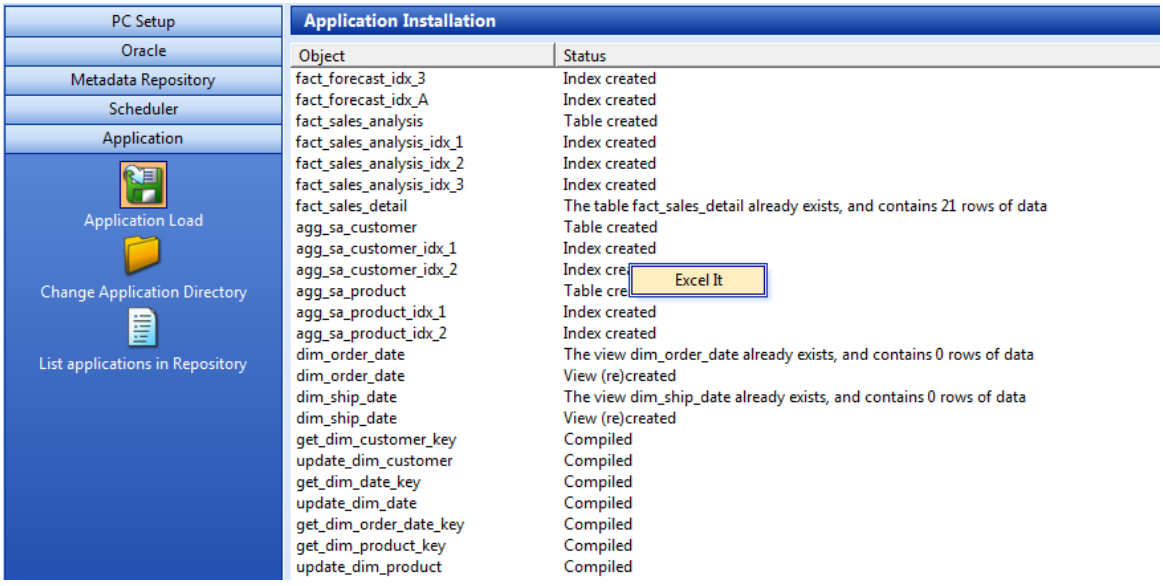
TIP: Added connections will need to be modified by selecting their properties in QAD Data Warehouse Designer.

The Tablespace Selection dialog

Tablespaces may need to be manually remapped to the corresponding tablespace available to the metadata repository. A dialog box as follows will appear for each tablespace encountered that is not automatically mapped (See tablespace options above).

The drop down list box 'New Tablespace' provides a list of all tablespaces visible to the repository.

A dialog box will appear when the application load has completed. The full results should be reviewed by scrolling back, or you can right click and select Excel It to send the results to Excel.



Testing Applications

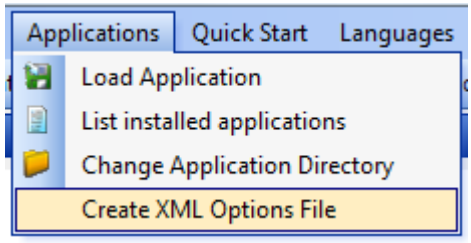
A testing application set consists of a Procedure and an XML script and provides the ability to define a series of tests against data warehouse objects; either comparing them to an expected value or to the results of a query.

The XML script contains the test definitions. Each test is a new XML node in the XML script and contains a name, a test query, an expected value, or a comparison query.

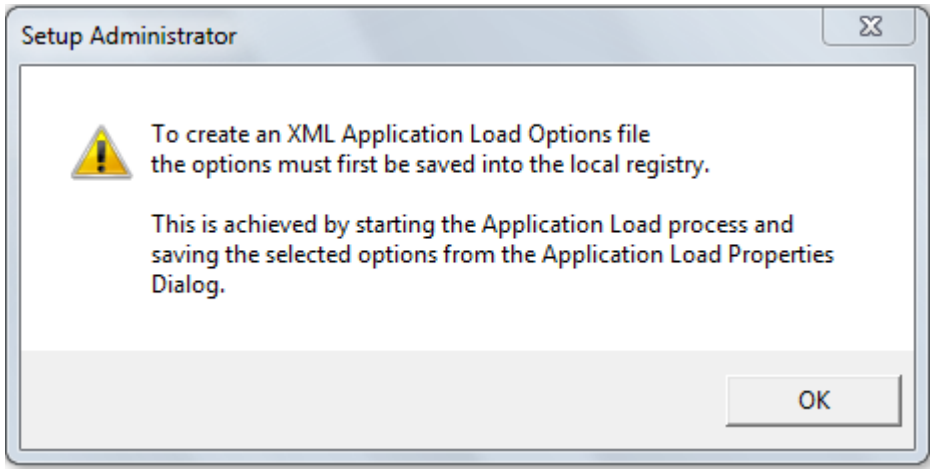
The procedure simply runs the tests and determines whether the tests are passed or not. This is most likely to be run as a scheduled job within QAD Data Warehouse Designer.

XML Files for Application Loads

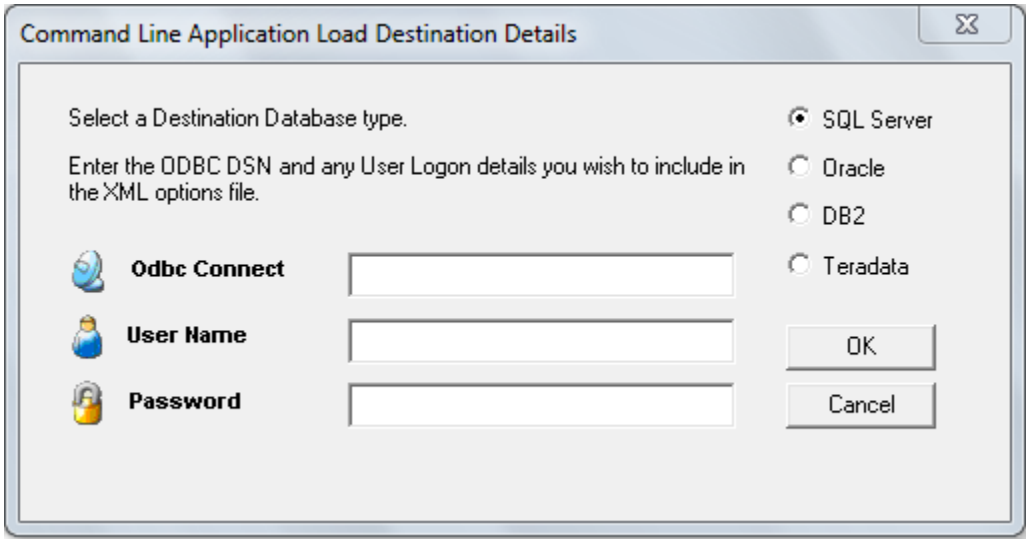
If applications are being loaded using the command line, the saved application load options in the local registry can be outputted to an xml file for use in the application load. To create the xml file, select Applications/Create XML Options File from the top toolbar.



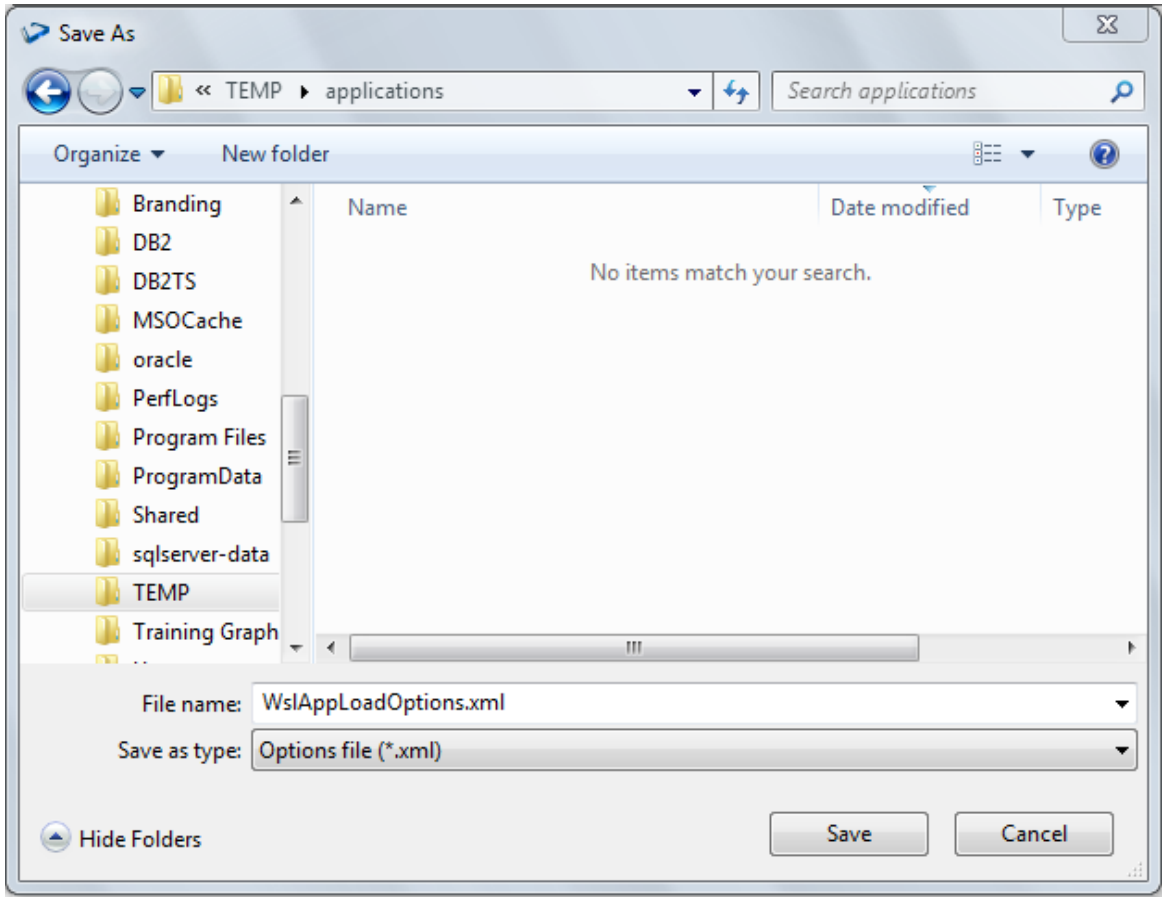
If the load options have not yet been saved into the local registry, the following message will appear; prompting you to select the application to be loaded and to saved the load settings for that application.



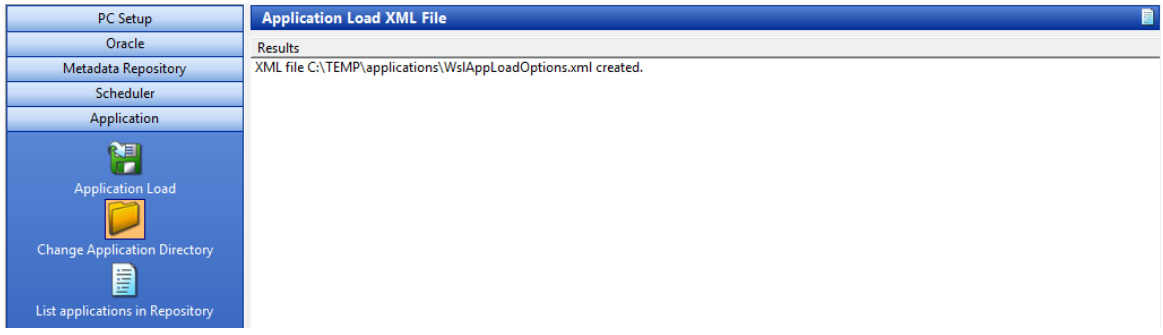
Enter in the Odbc Connection, User Name and Password; thus adding the destination details to the XML options file.



Choose the location; where the XML file is to be saved.



The results will confirm that the XML file has been created.



CHAPTER 10

Quick Start

The quick start component of the QAD 'Setup Administrator' attempts to do a rapid install of QAD Data Warehouse Designer onto a SQL Server database server.

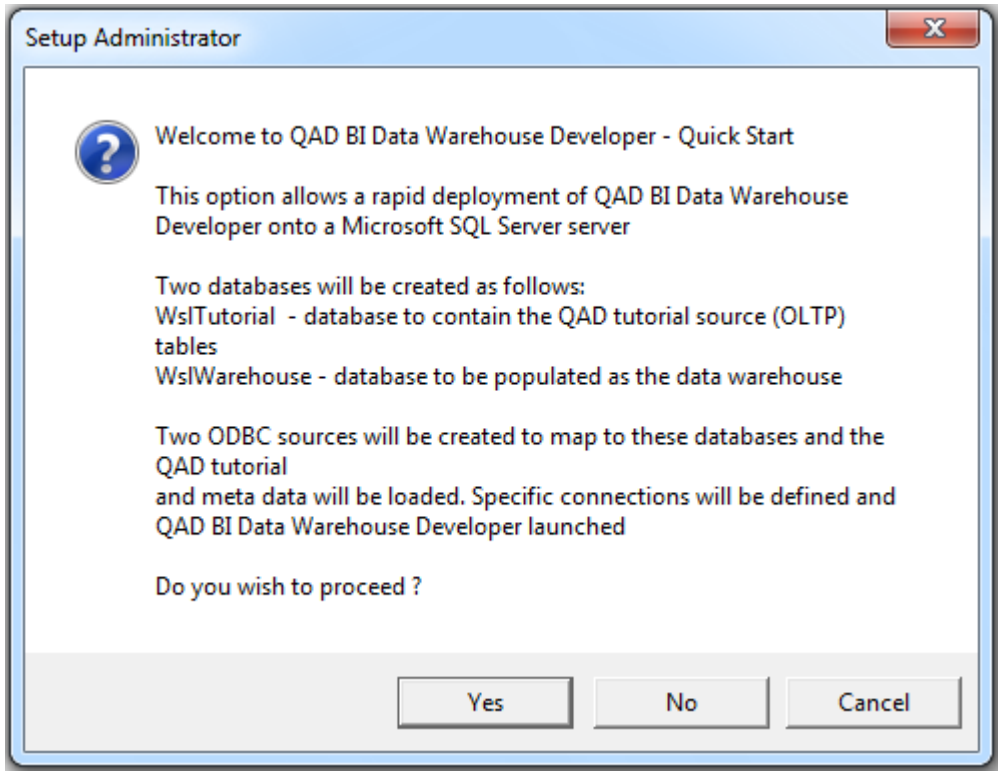
Click on the 'Quick Start' icon to start.

In This Chapter

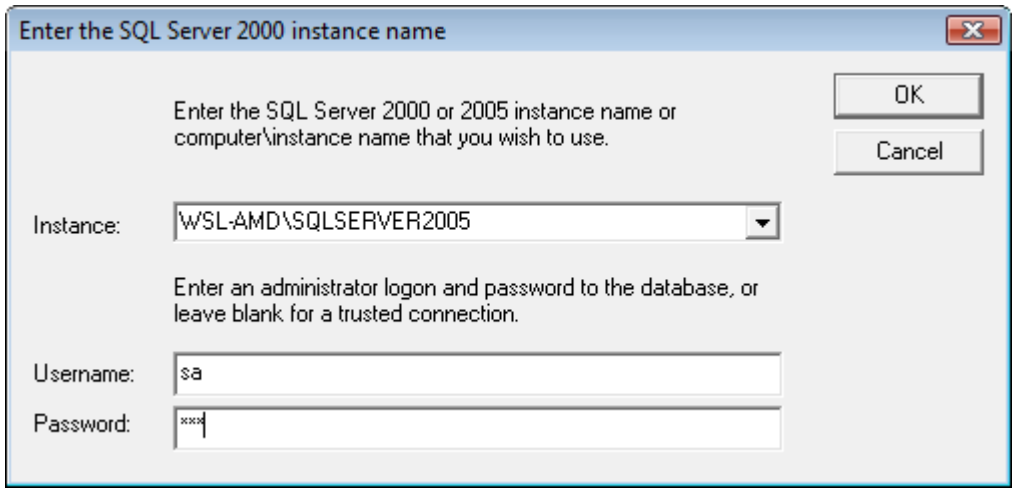
SQL Server Quick Start

The quick start option for SQL Server attempts to create two databases on a designated server. It then installs the Data Warehouse Designer software and the tutorial and finally invokes QAD Data Warehouse Designer and the on-line tutorial help.

When clicked the following dialog is displayed:



Push the Yes button to proceed with the creation of the databases. You will first be asked for the SQL Server instance in which to load the metadata repository and tutorial. The following dialog box is presented.



Data Warehouse Designer will attempt to locate all SQL Server servers available and list them in the drop down. This process can fail for a number of reasons, in which case the instance name will need to be ascertained and entered. Leave the username and password blank if using a trusted connection and no password is required.

Once the install has completed Data Warehouse Designer will start allowing you to logon to the newly created data warehouse,

CHAPTER 11

Quick Application

The quick application component of the QAD 'Setup Administrator' attempts to do a rapid install of QAD Data Warehouse Designer onto a SQL Server database server. It then attempts to install a specified application and optionally run a job within that application.

Click on the 'Quick Application' icon to start.

In This Chapter

SQL Server Quick Application

The quick application option for SQL Server attempts to create a Data Warehouse database, populate that database with the QAD metadata, load an application, install a scheduler and optionally run a batch job. It provides a means of quickly installed a Data Warehouse Designer application and getting the contents of the tables loaded and populated.

The quick application can be initiated from the Setup/Administrator utility menu or from the command line or from a Windows batch file.

Windows Command Line Invocation

The syntax for calling from the command line is as follows:

adm.exe -QA [-SWITCH "switch value"]

The switches that can be applied supply information to the Setup/Administrator (adm) utility. When all switches are provided the application install will occur automatically. Where some switches are left out a partially completed dialog will appear. The missing information can be provided and the install will then proceed.

Note: Where a parameter for a switch contains spaces the parameter must be enclosed in double quotes.

The switches are:

Switch	Parameter to be supplied
-QA	Indicates that this is a "Quick Application". No parameter is required.

-SS	The SQL Server instance name where the Data Warehouse database will reside
-SL	The administrator user name for the SQL Server instance defined with the -SS switch. If this switch is not provided a Trusted connection is assumed.
-SP	The administrator password for the SQL Server instance defined with the -SS switch.
-SN	The Data Warehouse database name. This database will be created in the instance defined by -SS. An ODBC Source DSN of the same name will also be created on the PC.
-AN	The Data Warehouse Designer application name. This is the name of the 'id' file and is typically of the structure app_id_XXXX_vvvvvv.wst where XXXX is the application identifier and vvvvvv is the application version. For example the tutorial application would be app_id_SQLT_tutorial_6.wst
-AD	The Data Warehouse Designer application directory. This is the directory in which the file defined by the -AN switch resides.
-LI	The SQL Server instance that contains the source database.
-LD	The ODBC source DSN name as defined in the Connection object in the data warehouse application. This ODBC source will be created.
-LN	The database name on the SQL Server instance as defined by the -LI switch. This is the database that contains the source data.
-LT	This switch does not have a parameter. If set it flags the ODBC source DSN as defined in the -LD switch as a trusted connection.
-JN	The name of a job in the data warehouse that is to be run once all other tasks are completed.
-CC	The name of the cube connection if appropriate. If set then all cubes will be updated to have this as the Analysis Services connection.
-DC	The name of the Data Warehouse connection if appropriate. If set then all cubes will be updated to have this as their Source connection.
-CD	The name of the cube database in Analysis Services if appropriate. If set then all cubes will be updated to have this database as their Analysis Services database.

For Example a batch file to call an application load of the tutorial may look as follows:

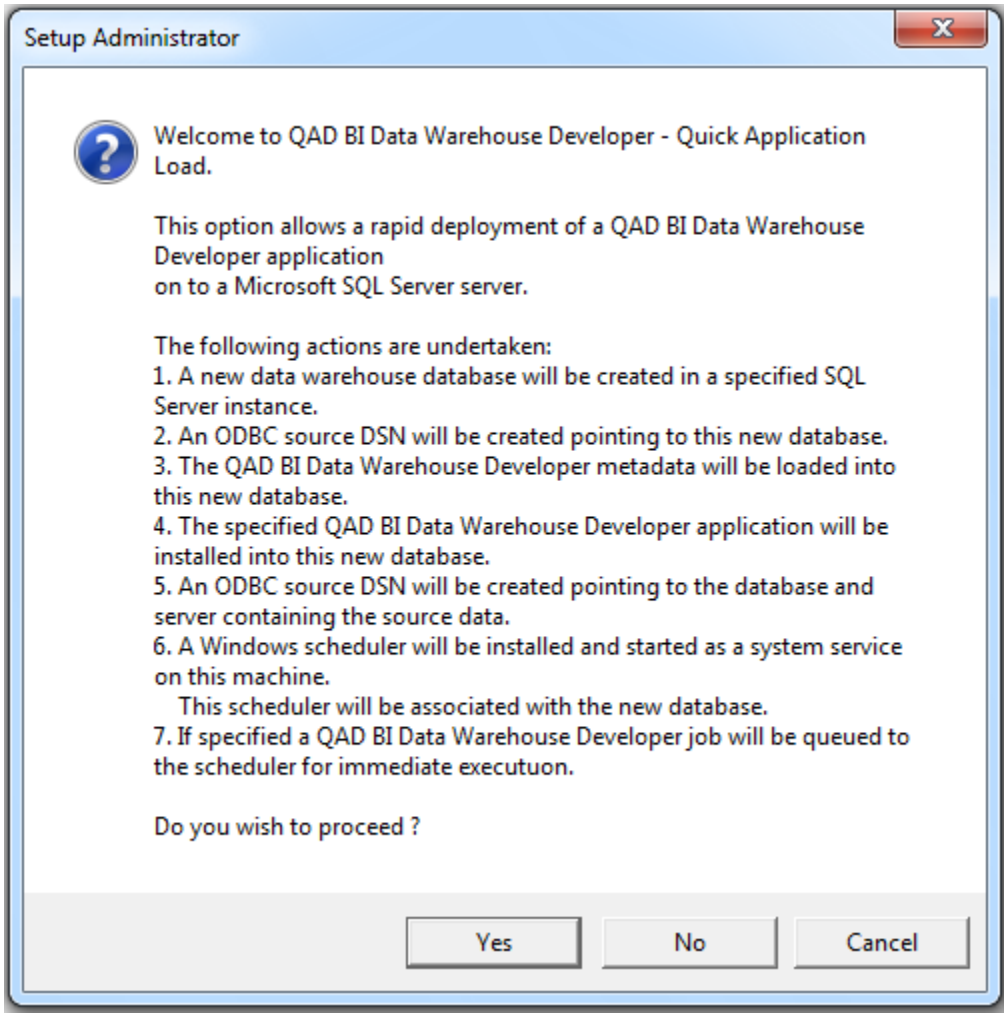
```
@echo off

c:\

cd \Program Files\QAD\
```

adm -QA -SS "WSL-TEST" -SN new_dw -LT -AN app_id_SQLT_tutorial_3.wst -AD "c:\Program Files\QAD\" -LI WSL-TEST -LD tutorial -LN tutorial -JN "Daily Update" -CC Cubes -CD My_database -DC DataWarehouse

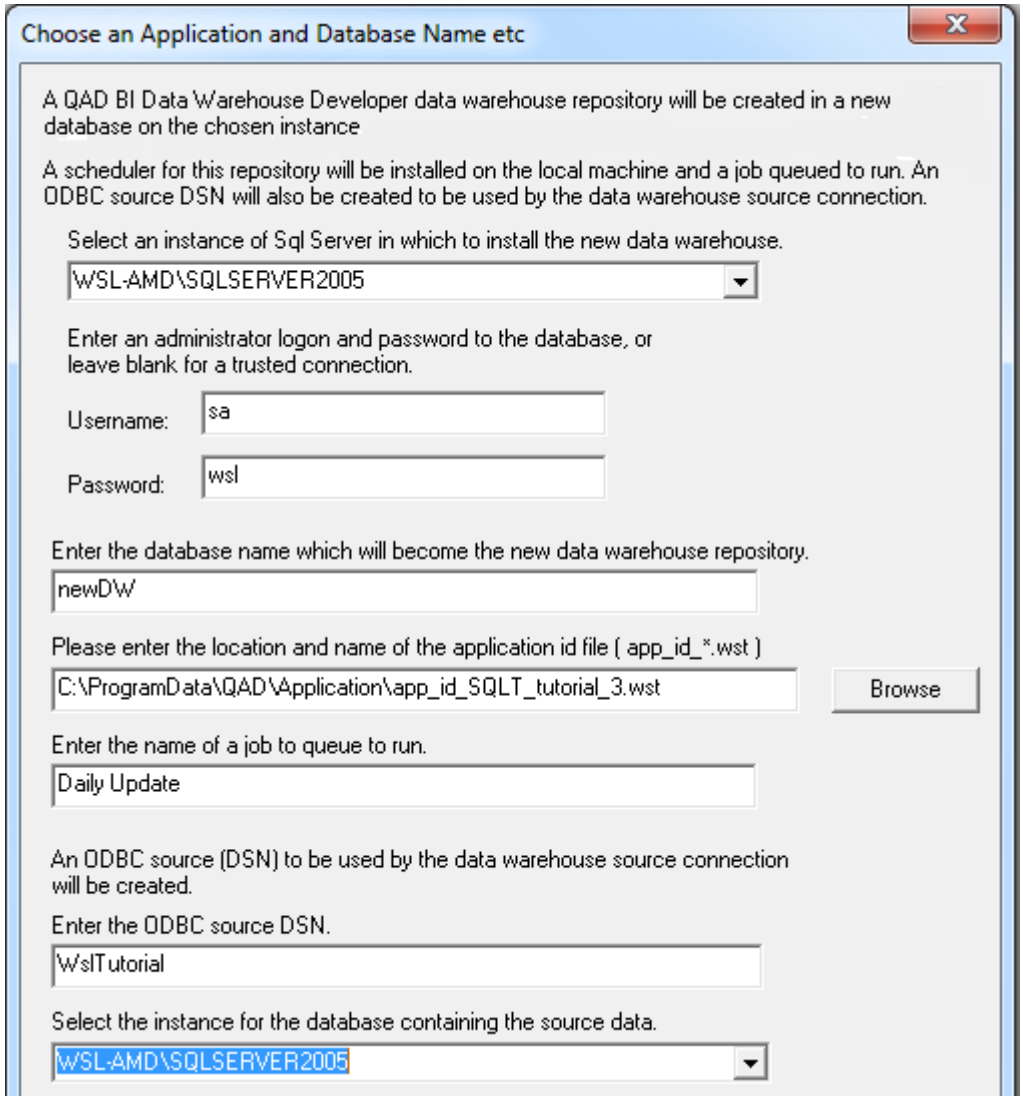
exit

Setup/Administrator Invocation

When clicked the following message displayed.



Click Yes to proceed with the Quick Application install. The following dialog will then be displayed.



The definition of the various fields can be found by examining the table in the section above. Once all values have been enteData Warehouse Designer click OK to proceed with the creation of the databases and the loading of the application etc.

CHAPTER 12

Languages

The Languages component of the QAD 'Setup Administrator' allows you to create a Language file or to load a Language File into Data Warehouse Designer.

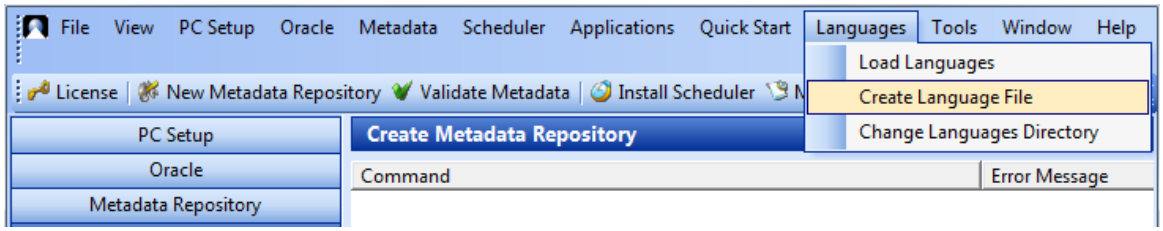
Click on the 'Languages' menu item in the command bar at the top of the screen. Click on any of the options under this level to action them. The Languages menu is as follows:

- The 'load languages' option will list the Language files that have been created. A file can then be selected and loaded into the target Metadata Repository.
- The 'create language file' option allows you to create a Language file for a particular Metadata Repository. A translator will then need to replace the English definitions in the file with the translations relevant to the chosen language.

In This Chapter

Create Language File

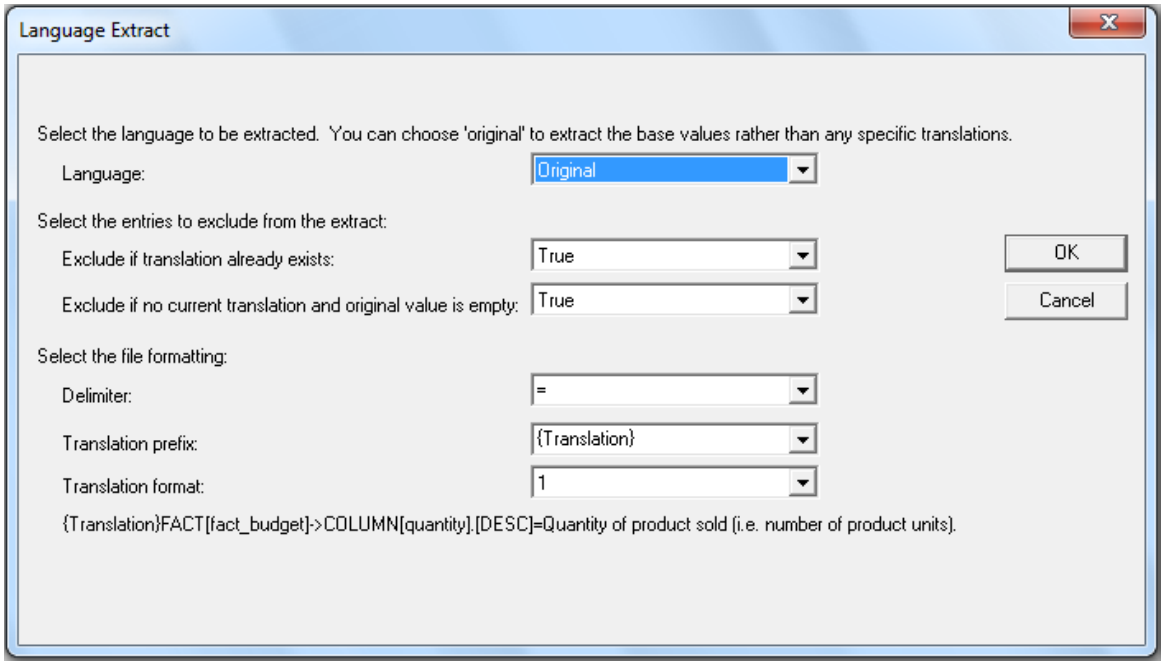
Click on the 'Languages' menu item in the command bar at the top of the screen. Select the 'Create Language File' option.



You will then be prompted to logon to the required metadata repository. Language Files are applied to a specific metadata repository.

Once logged on, you will be prompted for

- The Language - select from the drop-down list. Select a Language or choose 'Original' to create a Language file containing the original values.
- The entries to exclude from the extract.
- The file delimiter, translation prefix and translation format. The file delimiter provides an easy means for breaking up the file after extract. The translation prefix must be something that will NEVER appear in the data itself. This is used to identify the start/end of any given field translation.



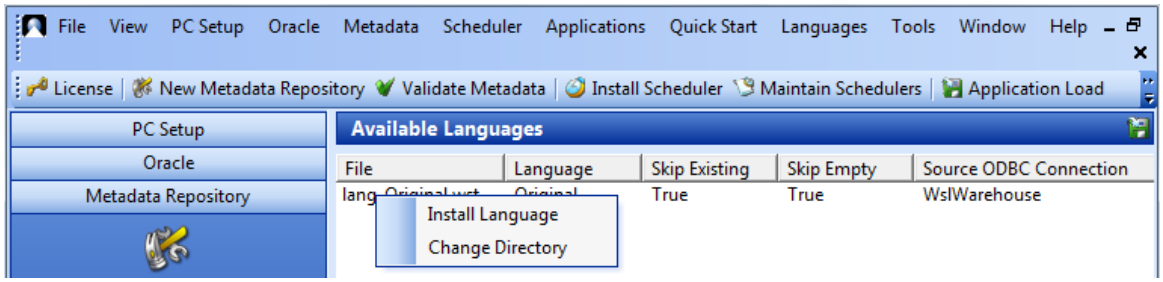
A Language extraction file will then be created and saved into C:\ProgramData\Data Warehouse Designer\Language\

A translator will need to replace the English definitions in the file with the translations relevant to the chosen language.

Load Languages

The 'Load Languages' menu option will list the Language Files available.

Right click on a particular file and select 'Install Language'.



You will then be prompted to logon to the required metadata repository and to select the Language to be updated in Data Warehouse Designer.

The results will be shown in the middle pane.

Index

X