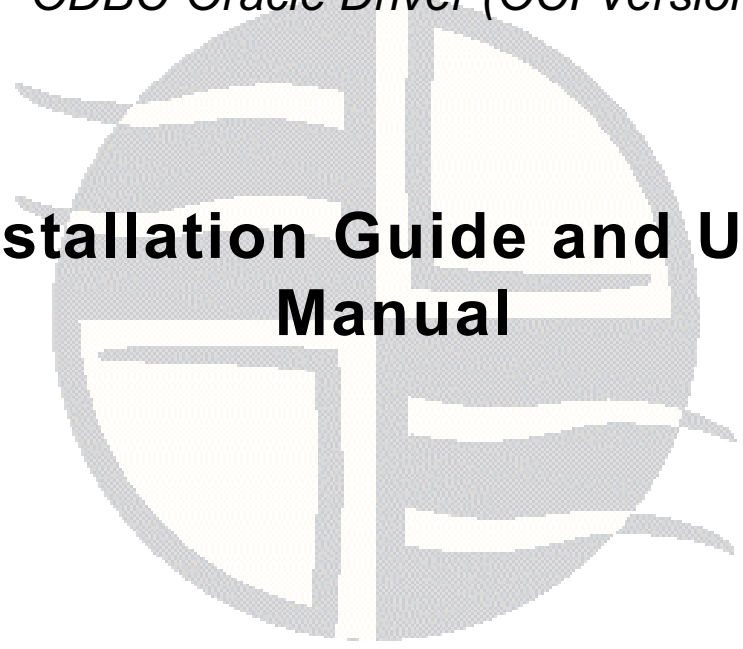
Easysoft® Data Access

ODBC-Oracle Driver (OCI Version)



Installation Guide and User Manual

Version 3.x.

This manual documents version 3.x of the Easysoft ODBC-Oracle Driver.

Publisher: Easysoft Limited

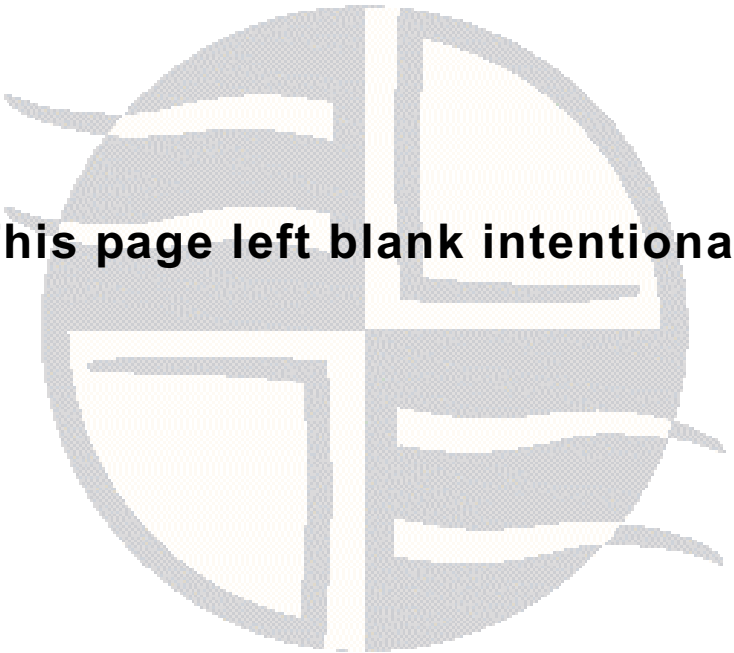
Thorp Arch Grange

Thorp Arch

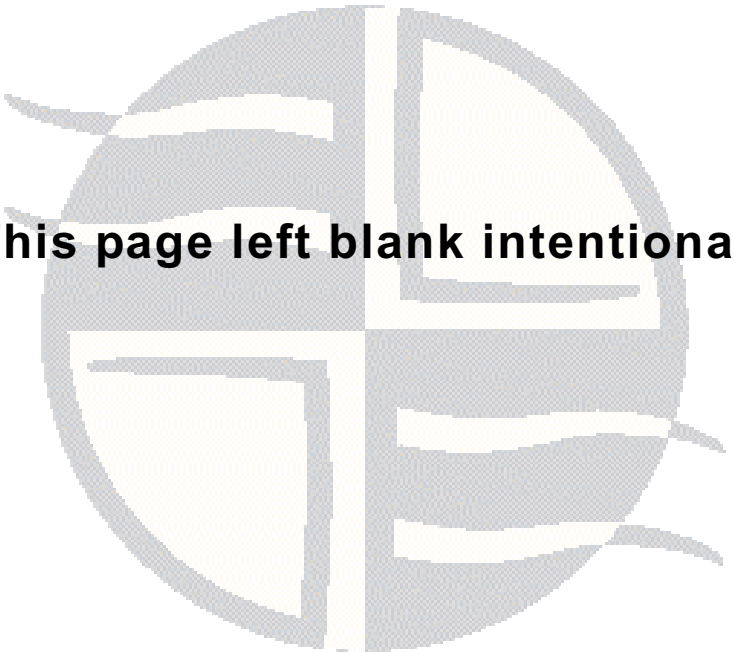
Wetherby

LS23 7BA

United Kingdom



This page left blank intentionally



This page left blank intentionally

PREFACE

About this manual

This manual is intended for use by anyone who wants to install the Easysoft ODBC-Oracle Driver, configure it, and then access Oracle data sources from an ODBC-compliant application.

Chapter Guide

Intended Audience

Sections written for the Microsoft Windows platforms require some familiarity with the use of buttons, menus, icons and text boxes, but should present no difficulties if you have any experience of Apple Macintosh computers, Microsoft Windows or the X Window System.

The Unix-based sections require experience of using Unix shell commands and performing basic functions like editing a file.

Displaying the Manual

This manual is available in the following formats:

- Portable Document Format (PDF), which can be displayed and printed using the Acrobat Reader, available free from Adobe at **http://www.adobe.com**.
- HTML (the format Easysoft recommend for viewing onscreen).

Notational Conventions

Across the range of Easysoft manuals you will encounter passages that are emphasized with a box and a label.

A *note box* provides additional information that may further your understanding of a particular procedure or piece of information relating to a particular section of this manual:

NB	Note boxes often highlight information that you may need to be aware of when using a particular feature.

A *reference box* refers to resources external to the manual, such as a useful website or suggested reading:

REF	For more manuals that use this convention, see the rest of the Easysoft documentation.

A *platform note* provides platform-specific information for a particular procedure step:

Unix	In Unix you must log on as the `root` user in order to make many important changes.

A *caution box* is used to provide important information that you should check and understand, prior to starting a particular procedure or reading a particular section of this manual:

Caution!	Be sure to pay attention to these paragraphs because Caution boxes are important!

Information has also been grouped within some chapters into two broad classes of operating system, Windows and Unix, for which side tabs are used to help you turn to the section relevant to you.

Typographical Conventions

To avoid ambiguity, typographic effects have been applied to certain types of reference:

- User interface components such as icon names, menu names, buttons and selections are presented in bold, for example:

  Click **Next** to continue.

  Where there is a chain of submenus, the following convention is used:

  Choose **Start > Programs > Command Prompt**.

- Commands to be typed are presented using a `monotype` font, for example:

  At the command prompt type `admin`.

- Keyboard Commands

  It is assumed that all typed commands will be committed by pressing the <*Enter*> key, and as such this will not normally be indicated in this manual. Other key presses are italicized and enclosed by angle brackets, for example:

  Press <*F1*> for help.

- File listings and system names (such as file names, directories and database fields) are presented using the `monotype` plain text style.

Contents

Trademarks

Throughout this manual, *Windows* refers generically to Microsoft Windows 98, 2000, NT, XP, ME, 2003 Server or Vista, which are trademarks of the Microsoft Corporation. The X Window system is specifically excluded from this and is referred to as *The X Window System* or just *X*.

Note also that although the name UNIX is a registered trademark of The Open Group, the term has come to encompass a whole range of UNIX-like operating systems, including the free, public Linux and even the proprietary Solaris. Easysoft use Unix (note the case) as a general term covering the wide range of Open and proprietary operating systems commonly understood to be Unix ‘flavors’.

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

Easysoft and Easysoft Data Access are trademarks of Easysoft Limited.

1 INTRODUCTION

Introducing the Easysoft ODBC-Oracle Driver

The Easysoft ODBC-Oracle Driver provides ODBC 3.5 access to Oracle 8.1.7+, 9*i*.x, 10*g* Release 1+, 11*g* Release 1+,12*c* Release 1+, 18*c* Release 1+ and Oracle Database XE databases from any supported Windows or Unix variant.

Although ODBC access from Windows client devices is common, this driver extends the same functionality to applications hosted on Linux and Unix systems, and may be extended to other platforms in the future.

Chapter Guide

Product Status

The Easysoft ODBC-Oracle Driver software is currently available on Windows, Unix and Linux platforms.

Software problems can be reported to **support@easysoft.com** by users who have either purchased support or registered via the website at **http://www.easysoft.com** and are evaluating products with a view to subsequent purchase.

Client tools tested include ApplixWare, StarOffice, Open Office, Perl DBI and PHP4. Notes associated with these products are located in **"Application Specific Issues" on page 124**.

Remote access to the Easysoft ODBC-Oracle Driver has also been tested via the Easysoft ODBC-ODBC Bridge. Applications tested by this route including Microsoft Access and Microsoft Query.

Product Dependencies

The Easysoft ODBC-Oracle Driver requires the user to obtain and install a copy of the Oracle Client Software (see **"Installing the Oracle Client" on page 22**).

Deployment

Several deployment options are available dependent upon the server platforms used and connectivity requirements.

The Easysoft ODBC-Oracle Driver can be located on any machine which has the Oracle client libraries. For platforms where the Oracle client libraries are not available or where you have many client machines, access may be provided with the addition of the Easysoft ODBC-ODBC Bridge. The addition of the Easysoft JDBC-ODBC Bridge also enables JDBC access from remote devices.

SCENARIO 1: LOCAL ACCESS TO A LOCAL ORACLE DATABASE

In this option Oracle client software will usually be already present on the host system.

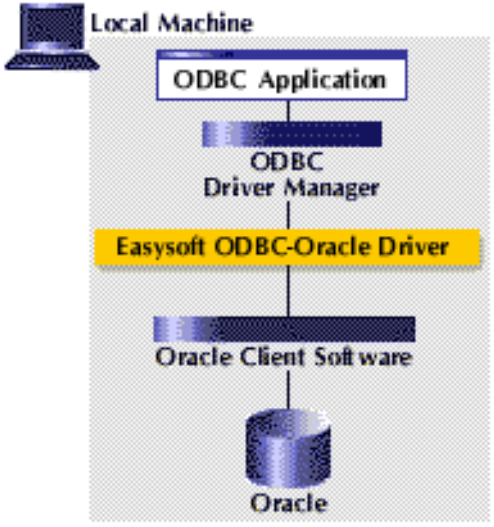


Figure 1: Local access to a local Oracle database

SCENARIO 2: SINGLE CLIENT ACCESS TO A REMOTE ORACLE DATABASE

This option uses Net8 as the communications protocol and therefore requires the installation of the Oracle client software on each client device.

If Oracle client software is not available for your platform you will need to use **"Scenario 3: Access to a remote Oracle database without Net8" on page 19**).

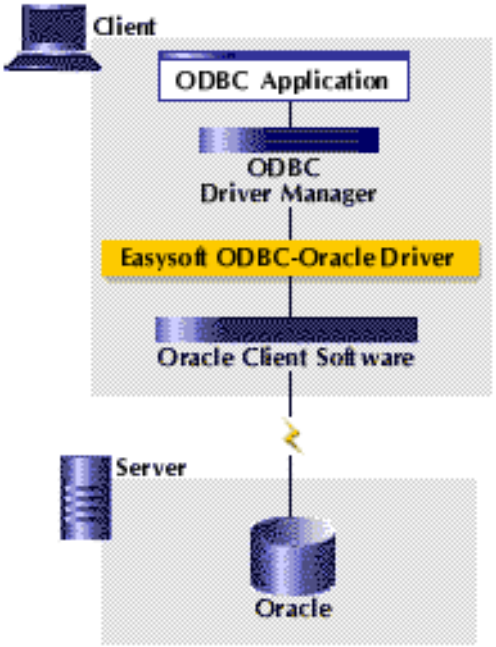


Figure 2: Single client access to a remote Oracle database

SCENARIO 3: ACCESS TO A REMOTE ORACLE DATABASE WITHOUT NET8

This option uses the Easysoft ODBC-ODBC Bridge and TCP/IP as the communications protocol.

This configuration does not require the installation of Oracle client software on each client device, making installation and management much simpler. This is appropriate if your client platform is supported by the Easysoft ODBC-ODBC Bridge, but Oracle client software is not available or you don't want to install the Oracle client on all client machines.

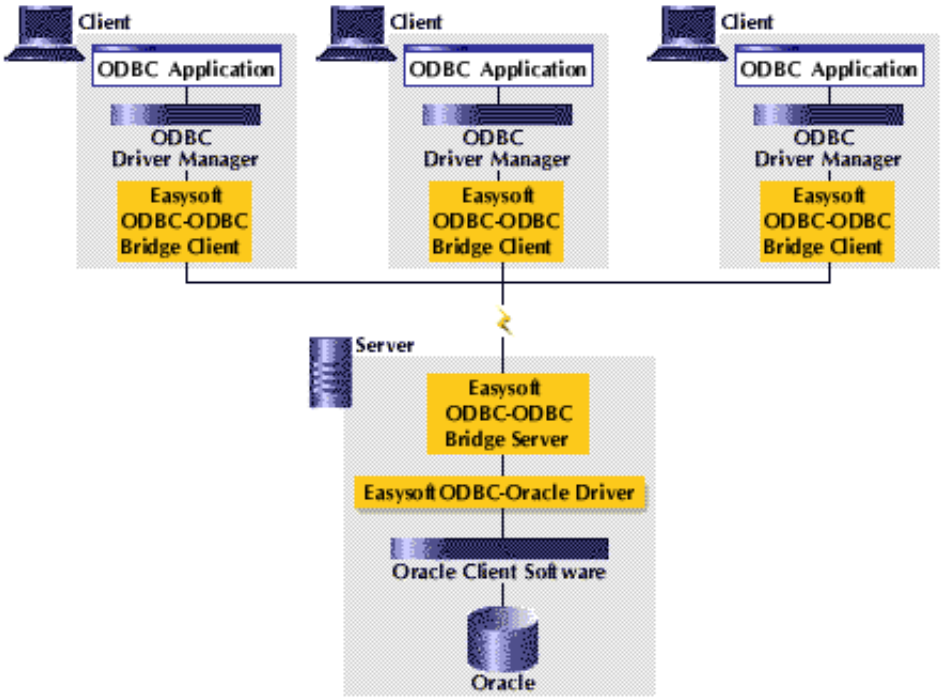
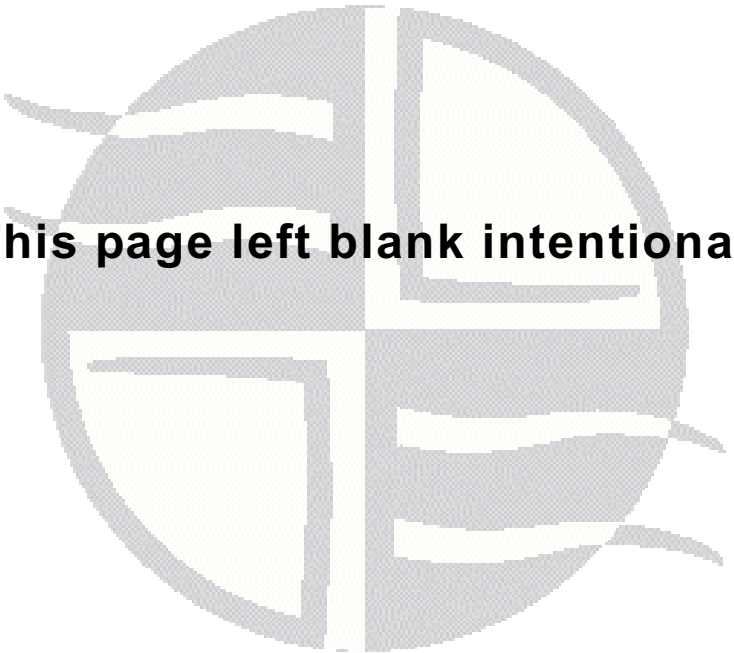


Figure 3: Access to a remote Oracle database without Net8



This page left blank intentionally

INSTALLATION

2

Installing the Easysoft ODBC-Oracle Driver

This section explains how to install, license and remove the Easysoft ODBC-Oracle Driver on supported Windows and Unix platforms.

The Windows installation can be carried out by anyone with local administrator privileges for the target machine.

The Unix installation assumes you are, or have available for consultation, a system administrator.

Chapter Guide

Installing the Oracle Client

The Easysoft ODBC-Oracle Driver uses the Oracle client software to access Oracle. Either the Instant Client or standard Oracle Client must be installed on the same machine as the Easysoft ODBC-Oracle Driver.

Although the Easysoft ODBC-Oracle Driver is compatible with both Oracle clients, Easysoft recommend that you use our driver with the Instant Client. This is because the Instant Client:

- Is quick to download and easy to install.
- Is available for more platforms than the standard client.
- Uses significantly less disk space than the standard client.

Is available for Oracle 10*g,* Oracle 11*g* and Oracle Database XE and backwards-compatible with Oracle 8, Oracle 9*i*.

64-bit Windows

The Easysoft installer program installs both a 32-bit and a 64-bit version of the Easysoft ODBC-Oracle Driver. If you want to use a 64-bit ODBC application, you need to use the 64-bit Easysoft ODBC-Oracle Driver. If you want to use a 32-bit ODBC application, you need to use the 32-bit Easysoft ODBC-Oracle Driver.

You need to use the 64-bit Easysoft ODBC-Oracle Driver with a 64-bit Oracle client (x64). You need to use the 32-bit Easysoft ODBC-Oracle Driver with a 32-bit Oracle client.

Installing and Testing the Instant Client

1. Log in at **http//www.oracle.com**.

   If you have not yet done so, you need to register first. You need to be a registered Oracle user to download the Instant Client.

2. Choose the Instant Client for your database and client platform at:

http://www.oracle.com/technology/tech/oci/instantclient/index.html

Download the Basic and SQL*Plus Instant Client Packages.

These packages contain the Instant Client and a version of SQL*Plus that is compatible with the Instant Client. SQL*Plus lets you test that you can access Oracle with the Instant Client.

Linux	**Instant Client 10.2**
	On Linux, the Instant Client 10.2 requires a minimum of glibc 2.3.3. To check the glibc version, on the Linux system, look at the first line of the output produced by running `/lib/libc.so.6` (on some platforms, `libc.so.6` is located in `/usr/lib`). For example:
	`/lib/libc.so.6`
	`GNU C Library stable release version 2.3.2, by Roland McGrath et al.`
	If you do not have the required glibc version, you need to download the Instant Client 10.1 packages. The Instant Client 10.1 is compatible with earlier versions of glibc.
	Instant Client 11.1
	On Linux, the Instant Client 11.1 requires glibc 2.3.3 or higher and the libaio package (version 0.3 or higher).

On the machine from which you want to access Oracle, extract the Instant Client and SQL*Plus files.

Unix

Add the Instant Client directory path to the `LD_LIBRARY_PATH` environment variable and export `LD_LIBRARY_PATH`.

Instant Client 11.1 Example

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib/instantclient_11_1
export LD_LIBRARY_PATH
```

Instant Client 10.2 Example

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib/instantclient_10_2
export LD_LIBRARY_PATH
```

Instant Client 10.1 Example

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib/instantclient10_1
export LD_LIBRARY_PATH
```

Replace `/usr/lib` with the directory where you extracted the Instant Client.

On AIX, replace `LD_LIBRARY_PATH` with `LIBPATH`.

Windows	Add the Instant Client directory path to the `PATH` environment variable. If **Control Panel** has a search facility in your version of Windows, search for "environment variables" to locate the dialog box where you can edit the `PATH`. Otherwise, in **Control Panel**, double-click **System**. In the **Advanced** tab, click **Environment Variables**. In the **System Variables** list, double-click **Path**. Use a semicolon (;) to separate the Instant Client directory from the existing entries.
	Instant Client 11.1 Example
	`%SystemRoot%;C:\Oracle\instantclient_11_1`
	Instant Client 10.2 Example
	`%SystemRoot%;C:\Oracle\instantclient_10_2`
	Instant Client 10.1 Example
	`%SystemRoot%;C:\Oracle\instantclient10_1`

3. Change to the directory where you extracted SQL*Plus. Then use SQL*Plus to test that you can access Oracle:

```
sqlplus username/password@//machine_name:port/database_name
```

where:

- *username* and *password* are the database username and password.
- *machine_name* is the host name or IP address of the Oracle database server.
- *port* is the Oracle listener port.
- *database_name* is the database you want to access.:

Note	If you are unable to connect to Oracle with SQL*Plus, contact your Oracle Database Administrator. If you cannot access your Oracle database with SQL*Plus, you will not be able to access the database with the Easysoft ODBC-Oracle Driver.

4. At the prompt, type a SELECT statement to test that you can retrieve some data:

```
select * from dual;
```

To exit SQL*Plus, type `exit`.

OBTAINING THE STANDARD CLIENT

For information about how to obtain the standard Oracle client, see:

http://www.easysoft.com/support/kb/kb00612.html

Obtaining the Easysoft ODBC-Oracle Driver

There are three ways to obtain the Easysoft ODBC-Oracle Driver:

- The Easysoft web site is available 24 hours a day at **http://www.easysoft.com** for downloads of definitive releases and documentation.

  Select **Download** from the Easysoft ODBC-Oracle Driver section of the website and then choose the platform release that you require.

  First time visitors must complete the new user form and click **Register**. Note that your personal Internet options may require you to login and click **Continue** if you have previously registered.

- The Easysoft FTP server is available 24 hours a day at **ftp://ftp.easysoft.com**, containing upgrades, documentation and beta releases of Easysoft products, as well as definitive releases.

  Change to the `pub/oracle` directory and then choose the platform release that you require.

- You can order Easysoft software on CD by email, telephone or post (see **Contact Details**).

What to install

The name of the Easysoft ODBC-Oracle Driver install file varies from platform to platform, but is of the form:

- `odbc-oracle-`*`x_y_z-platform`*`.exe` (Windows)

– OR –

- `odbc-oracle-`*`x.y.z-platform`*`.tar` (Unix)

where "*x*" is the major version number, "*y*" is the minor version number and "*z*" is the build index, which is incremented when fixes are applied to previous releases.

"*platform*" will vary depending on the operating system distribution you require and there are some variations of the form:

- `odbc-oracle-`*`x.y.z-platform-variation`*`.tar`

within specific Unix platforms, where "*variation*" refers to alternative versions available for a single platform.

NB	Select the highest release available for your platform within your licensed major.minor version number (installing software of a different major.minor version number requires a new Easysoft license).

Unix filenames may also be suffixed with `.gz` for a "`gzipped`" archive, `.bz2` for a "`bzip2ed`" archive, or `.Z` for a "`compressed`" archive.

NB	If you download a Unix file in Windows, the browser may alter the filename. For example, if you download a `.gz` file it may not be obvious that the file is "`gzipped`". Use "`file` *filename*" to find out the file type of the downloaded file.

You can now download a file and begin the installation process.

Caution!

On Unix, as long as you stop any software either from Easysoft or using Easysoft drivers, it is safe to reinstall or upgrade the Easysoft ODBC-Oracle Driver without uninstalling.

If you do uninstall, you should first back up any configuration data that you still need, as uninstalling some Easysoft products will result in this information being deleted (license details remain in place).

Refer to the section relevant to your platform to continue:

- **"Installing on Windows" on page 30**
- **"Uninstalling on Windows" on page 35**
- **"Installing on Unix" on page 37**
- **"Uninstalling on Unix" on page 55**

Installing on Windows

- Execute the file distribution that you downloaded in **"Obtaining the Easysoft ODBC-Oracle Driver" on page 27**.

  Follow the on screen instructions.

Updating Files That are in Use

To avoid rebooting your computer, the Easysoft ODBC-Oracle Driver installer prompts you when files that it needs to update are in use by another application or service. This frees the locked files and allows the installation to complete without a system restart.

On Windows Vista, the Easysoft ODBC-Oracle Driver installer uses the Restart Manager to locate the applications that are using files that need updating. These applications are displayed in the Files in Use dialog box. To avoid a system restart, choose **Automatically close applications and attempt to restart them after setup is complete**. The Easysoft ODBC-Oracle Driver installer then uses the Restart Manager to try to stop and restart each application or service in the list. If possible, the Restart Manager restores applications to the same state and with the same data that they were in before it shut them down.

On earlier versions of Windows, when the Files in Use dialog is displayed, manually shut down each application in the list and then click **Retry** to avoid a system restart.

Licensing on Windows

The install program starts the Easysoft License Manager (documented in the **Licensing Guide**).

The following types of license are available:

- a *free time-limited trial license* which gives you free and unrestricted use of the product for a limited period (usually 14 days).
- a *full license* if you have purchased the product. On purchasing the product you are given an authorization code, which you use to obtain a license.

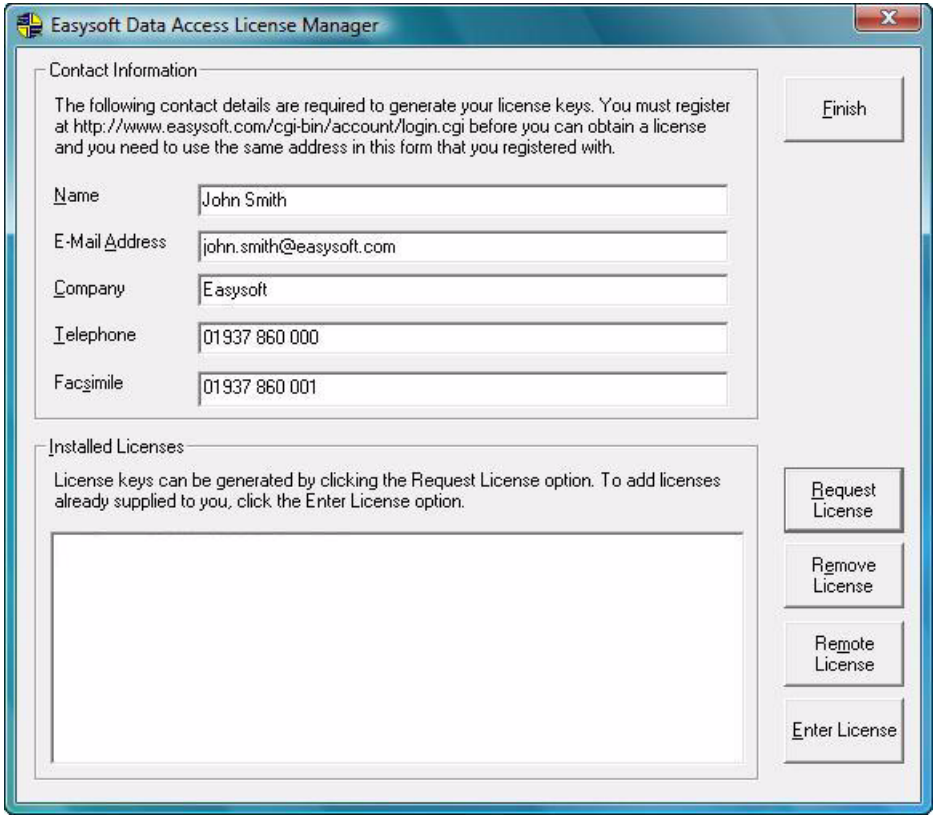


Figure 4: The License Manager dialog box

1. Enter your contact details.

   You **must** enter the **Name**, **E-Mail Address** and **Company** fields.

   The **Telephone** and **Facsimile** fields are important if you require Easysoft to contact you by those methods.

   The **E-Mail Address must** be the same as the address used to register and download from the Easysoft web site or you will be unable to obtain trial licenses.

2. Click **Request License**.

   You are asked for a license type:

   The next step depends on the type of license you want.

3. For a trial license click **Time Limited Trial** and then click **Next**.

   The License Manager asks what software you are licensing.

   Select the **Easysoft ODBC-Oracle Driver** from the drop-down list and then click **Next**.

   – OR –

   If you have obtained an authorization code for a purchased license, select **Non-expiring License** and then click **Next**. The License Manager requests your authorization code.

   Enter the authorization code and then click **Next**.

   The License Manager displays a summary of your details and allows you to choose the method of applying for your license:

4. Choose **On-line Request** if your machine is connected to the internet and can make outgoing connections to port 8884

The License Manager then sends a request to the Easysoft license server to activate your license key automatically. This is the quickest method and results in your details being entered immediately into our support database. The whole process is automatic so you can proceed to **step on page 34**.

NB	Only your license request identifier and contact details as they are displayed in the main License Manager screen are sent to Easysoft.

The remaining three options (**Email Request**, **Print Request** and **View Request**) are all methods to obtain a license if your machine is offline (i.e. does not have a connection to the internet).

Each of these methods involves providing Easysoft with information including your machine number (a number unique to your machine) and then waiting to receive your license key.

Instead of emailing, faxing or telephoning your details to Easysoft, you can enter them directly at the Easysoft web site and your license key will be emailed to you automatically.

To use this method, click **View Request**, and then visit:

- **http://www.easysoft.com/support/licensing/trial_license.html**

  (trial licenses)

- **http://www.easysoft.com/support/licensing/full_license.html**
  (purchased licenses)

In the Licensing page, enter your machine number (and authorization code for purchased license), click **Submit** and your license key will be emailed to you.

> **NB** You can copy your machine number from the **View Request** dialog box using `CTRL-C` and then paste it into the License Generator by using `CTRL-V`.

When you receive the license key, you can activate it either by double-clicking the email attachment or by clicking **Enter License** on the License Manager main screen and pasting the license key into the dialog box.

A message tells you how many licenses have been added..

> **NB** If you use the **Email Request** option, the license key is emailed to the email address as displayed on the License Manager screen, not the `from:` address of your email.

For more information about licensing refer to the **Licensing Guide**.

5. Click **Finish** in the License Manager.

   The installation is complete.

   You should have a new Easysoft ODBC-Oracle Driver program group with a link to this user guide.

REPAIRING THE EASYSOFT ODBC-ORACLE DRIVER INSTALLATION

The installer can repair a broken Easysoft ODBC-Oracle Driver installation. For example, you can use the installer to restore missing Easysoft ODBC-Oracle Driver files or registry keys.

1. Do one of the following:
   - In **Control Panel**, open **Programs and Features**.
   - In earlier versions of Windows, in **Control Panel** open **Add or Remove Programs**.
2. Do one of the following:
   - Right-click **Easysoft ODBC-Oracle Driver**, and then click **Repair**.
   - In previous of Windows, select **Easysoft ODBC-Oracle Driver** and click **Change/Remove**.

Uninstalling on Windows

Removing Easysoft ODBC-Oracle Driver Data Sources

Easysoft ODBC-Oracle Driver data sources are not removed when you uninstall. You therefore do not need to recreate your Easysoft ODBC-Oracle Driver data sources if you reinstall or upgrade. If you do not want to keep your Easysoft ODBC-Oracle Driver data sources, use ODBC Administrator to remove them, **before** uninstalling the Easysoft ODBC-Oracle Driver.

1. In **Control Panel**, double-click **Administrative Tools** and then **Data Sources (ODBC)**.

2. Select the data source in the **ODBC Administrator** and click the **Remove** button.

64-bit Windows

There is both a 32-bit and a 64-bit version of ODBC Administrator. The 64-bit ODBC Administrator is located in Control Panel under Administrative tools. To access the 32-bit ODBC Administrator, in the Windows Run dialog box, type:

```
%windir%\syswow64\odbcad32.exe
```

If you do not see the data source in the 64-bit ODBC Administrator, look for it in the 32-bit ODBC Administrator.

REMOVING THE EASYSOFT ODBC-ORACLE DRIVER

1. Do one of the following:
   - In **Control Panel**, open **Programs and Features**.
   - In earlier versions of Windows, in **Control Panel** open **Add or Remove Programs**.
2. Do one of the following:
   - Double-click **Easysoft ODBC-Oracle Driver**.
   - In earlier of Windows, select **Easysoft ODBC-Oracle Driver** and click **Change/Remove**.

Any licenses you obtained for the Easysoft ODBC-Oracle Driver and other Easysoft products are held in the Windows registry.

When you uninstall, your licenses are not removed so you do not need to relicense the product if you reinstall or upgrade.

Installing on Unix

These instructions show how to install the Easysoft ODBC-Oracle Driver on Unix and Linux platforms.

BEFORE YOU INSTALL

Requirements

The installation script has a minimal set of requirements:

- Bourne shell in `/bin/sh` (if your Bourne shell is not located there you may need to edit the first line of the install file).
- Various commonly used Unix commands such as:

  `grep, awk, test, cut, ps, sed, cat, wc, uname, tr, find, echo, sum, head, tee, id`

  If you are missing any of these commands, they can generally be obtained from the Free Software Foundation (**http://www.fsf.org**). As some machines have a broken `tee` command, the distribution comes with a `tee` replacement.
- Depending on the platform, you will need up to 10Mb of disk space free for the installed programs and up to 10Mb temporary space of the installation files themselves. If you install the unixODBC driver manager as well, these numbers increase by approximately 1.5Mb.
- For Easysoft Licensing to work you must do one of the following:

  a) Install in `/usr/local/easysoft`.

  b) Install elsewhere and symbolically link `/usr/local/easysoft` to wherever you chose to install the software.

The installation will do this automatically for you so long as you run the installation as someone with permission to create `/usr/local/easysoft`.

c) Install elsewhere and set the `EASYSOFT_ROOT` environment variable. See **"Post Installation" on page 52**.

- An ODBC driver manager. Easysoft ODBC-Oracle Driver distributions contain the unixODBC driver manager but you can use an already installed unixODBC if you prefer.
- You do not have to be the `root` user to install but you will need permission to create a directory in the chosen installation path. Also, if you are not the `root` user, it may not be possible for the installation to:

1. Register the Easysoft ODBC-Oracle Driver with unixODBC.
2. Create the example data source in the SYSTEM `odbc.ini` file.
3. Update the dynamic linker entries (only some platforms).

If you are not `root`, these tasks will have to be done manually later. Easysoft recommend you install all components as the `root` user.

What you can Install

This distribution contains:

- The Easysoft ODBC-Oracle Driver.
- The unixODBC driver manager.

You will need an ODBC driver manager to use the driver from your applications. This distribution contains the unixODBC driver manager (see **www.unixodbc.org**). Most (if not all) Unix applications and interfaces (for example, Perl DBD::ODBC, PHP and Python) support the unixODBC driver manager.

You do not have to install the unixODBC driver manager in this distribution as you can use an already installed unixODBC (whether that was installed with another Easysoft product, from your operating system vendor or even if you built it yourself). However, Easysoft ensure the unixODBC distributed with Easysoft ODBC drivers has been tested with our drivers so we recommend you use it.

If you choose to use an already installed unixODBC driver manager the installation script will attempt to locate it. The installation looks in the standard places but if you have installed it in a non-standard location you will need to provide that location to the installation script when it prompts you. The installation primarily needs unixODBC's `odbcinst` command to install drivers and any data sources.

Where to Install

This installation needs a location for the installed files. The default is `/usr/local`.

At the start of the installation, you will be prompted for an installation path. All files are installed in a subdirectory of your specified path called "`easysoft`" For example, if you pick the default of `/usr/local`, the product will be installed in `/usr/local/easysoft` and below.

If you choose an install path different from the default, the installation will try to symbolically link `/usr/local/easysoft` to the `easysoft` in your chosen path. This allows us to distribute binaries with built in dynamic linker run paths. If you are not `root` or the path `/usr/local/easysoft` already exists and is not a symbolic link this will fail (see later for how this may be corrected manually). You should note that you cannot license Easysoft products until either of the following is true:

- `/usr/local/easysoft` exists either as a symbolic link to your chosen install path or as the install path itself.
- You set `EASYSOFT_ROOT` to point to *`install_path`*`/easysoft`.

Changes Made to your System

This installation installs files in subdirectories of the path requested at the start of the installation and depending on what is installed a few changes may be made to your system as outlined below:

1. If you choose to install this driver into unixODBC, unixODBC's `odbcinst` command will be run to add an entry to your `odbcinst.ini` file. You can locate this file with `odbcinst -j` (`odbcinst` will be in *`install_path`*`/easysoft/unixODBC/bin`) if you are using the unixODBC in this distribution.

   The entry for this driver will look similar to this:

```
[ORACLE]
Description = Easysoft ODBC-Oracle Driver
Driver = /usr/local/easysoft/oracle/libesoracle.so
Setup =
/usr/local/easysoft/oracle/libesoraclesetup.so
FileUsage = 1
```

Obviously, the paths depend on where you installed this software.

See **"Uninstalling on Unix" on page 55** for ways to remove this.

2. The installation installs an example data source into unixODBC. This entry will be added to your SYSTEM `odbc.ini` file. You can locate your SYSTEM `odbc.ini` file using `odbcinst -j`. The entry will look similar to this:

```
[ORACLE]
Driver                  = ORACLE
Database = //my_database_host:1521/my_database_service_name
User                    = my_user
Password                = my_password
METADATA_ID             = 0
ENABLE_USER_CATALOG     = 1
ENABLE_SYNONYMS         = 1
```

If you are using the standard Oracle Client, the `Database` attribute value for this data source will look similar to this:

```
Database                = my_database
```

See **"Uninstalling on Unix" on page 55** for ways to remove this data source.

3. Dynamic Linker.

   On operating systems where the dynamic linker has a file specifying locations for shared objects (Linux, FreeBSD), the installation will attempt to add paths under the path you provided at the start of the install to the end of this list.

   On Linux, this is generally the file `/etc/ld.so.conf`.

   On FreeBSD, this is generally the file `/etc/defaults/rc.conf`.

Reinstalling or Installing when you Already have Other Easysoft Products Installed

Each Easysoft distribution contains common files shared between Easysoft products. These shared objects are placed in *`install_path`*`/easysoft/lib`. When you run an installation, the dates and versions of these files will be compared with the same files in the distribution and only updated if the files being installed are newer or have a later version number.

You should ensure that nothing on your system is using Easysoft software before starting an installation because on some platforms, files in use cannot be replaced. If a file cannot be updated, you will see a warning during the installation. You may review all warnings after the installation in the file called "`warnings`" in the directory you unpacked the distribution into.

If the installer detects you are upgrading a product the installer will suggest you delete the product directory to avoid having problems with files in use. An alternative is to rename the specified directory.

If you are upgrading you will need a new license from Easysoft to use the new driver.

Gathering Information Required during the Installation

During the installation you will be prompted for various information.

Before installing, you should determine:

- If you have unixODBC already installed and where it is installed. The install searches standard places like `/usr` and `/usr/local` but if you installed in a non-standard place and you don't install the included unixODBC you will need to know the location.

INSTALLATION

Unpacking the Distribution

The distribution for Unix platforms is distributed as a tar file. There are multiple copies of the same distribution with different levels of compression. You unpack the distribution as follows.

If the distribution file has been gzipped (i.e. the filename ends in `.gz`), then use:

```
gunzip odbc-oracle-x.y.z-platform.tar.gz
```

If the distribution file has been bzipped (i.e. the filename ends in `.bz2`), then use:

```
bunzip2 odbc-oracle-x.y.z-platform.tar.bz2
```

If the distribution file has been compressed (i.e. the filename ends in `.Z`), then use:

```
uncompress odbc-oracle-x.y.z-platform.tar.Z
```

You may have a distribution file that is not compressed at all (i.e. the filename ends in .tar).

To extract the installation files from the tar file use:

```
tar -xvf odbc-oracle-x.y.z-platform.tar
```

This will create a directory with the same name as the tar file (without the `.tar` postfix) containing further archives, checksum files, an install script and various other installation files.

Change directory into the directory created by unpacking the tar file.

License to Use

The license text can be found in the file `license.txt` and be sure to understand the terms before continuing, as you will be required to accept the license terms at the start of the installation.

Answering Questions during the Installation

Throughout the installation you will be asked to supply the answer to some questions. In each case the default will be displayed in square brackets and you need only press *<Enter>* to take the default. If there are alternative responses these will be shown in round brackets; to pick one of these type them and press *<Enter>*. For example:

```
Do you want to continue? (y/n) [n]:
```

The possible answers to this question are "`y`" or "`n`". The default when you enter nothing and press *<Enter>* is "`n`".

Running the Installer

Before you run the installer, make sure you have read **"Before you Install" on page 37**. If you are considering running the installation as a non `root` user, we suggest you review this carefully as you will have to get a `root` user to manually complete some parts of the installation afterwards. Easysoft recommend installing as the `root` user (if you are concerned about the changes that will be made to your system see **"Changes Made to your System" on page 40**).

To start the installation run:

```
./install
```

You will need to:

- Confirm your acceptance of the license agreement with "`yes`" or "`no`". See **"License to Use" on page 44**.
- Enter a location where the software is to be installed. Easysoft recommend taking the default here. See **"Where to Install" on page 39**.

NB	If you are upgrading, you will need a new license from Easysoft.

Locating or Installing unixODBC

Easysoft strongly recommend you use the unixODBC driver manager because:

- The installation is designed to work with unixODBC and can automatically add ODBC drivers and DSNs during the install.
- Most applications and interfaces that can use ODBC know about unixODBC and so any new ODBC drivers or data sources you add with this installation will automatically become available to your applications and interfaces.
- The unixODBC project is currently led by Easysoft developer Nick Gorham. This means there is much greater experience with unixODBC within Easysoft and we will be able to provide better support when running under unixODBC. It also means that if you find a problem in unixODBC it is much easier for us to facilitate a fix.

- The unixODBC package contains much more than a driver manager. The aim of the unixODBC project is to provide all the ODBC functionality available on Windows for Unix operating systems. The unixODBC package may be built with the QT libraries to allow GUI configuration of DSNs and drivers. It also contains the GUI DataManager program that may be used to explore your ODBC data. The Easysoft ODBC-Oracle Driver contains the code and shared object that is used by unixODBC's GUI `ODBCConfig` utility to add/delete and configure Easysoft ODBC-Oracle Driver DSNs.

The installation will start by searching for an installed unixODBC.

There are two possible outcomes here:

1. If unixODBC is located a message will be output saying:

   `Found unixODBC under /path_to_unixODBC and it is version n.n.n`

2. unixODBC is not found.

   If unixODBC is not found in the standard places, you will be asked whether you have it installed.

   If you have it installed, you need to provide the argument given to unixODBC's configure as `--prefix`. i.e. if you built unixODBC with "`configure --prefix=/usr/local/unixODBC`" you enter "`/usr/local/unixODBC`". Generally the path required is the directory above where `odbcinst` is installed i.e. if `odbcinst` is in `/opt/unixODBC/bin/odbcinst` the required path is `/opt/unixODBC`.

   If you have not got unixODBC installed, you should install the unixODBC included with this distribution.

If you already have unixODBC installed, you do not have to install the include unixODBC but you might consider doing so if your version is older than the one included.

The unixODBC in this distribution is not built with the default options in unixODBC's configure line:

- `--prefix=/etc`

  This means the default SYSTEM `odbc.ini` file where SYSTEM dsns are located will be `/etc/odbc.ini`.

- `--enable-drivers=no`

  This means other ODBC drivers that come with unixODBC are not installed.

- `--enable-iconv=no`

  This means unixODBC will not look for a `libiconv`. Warnings about not finding an iconv library were confusing our customers.

- `--enable-stats=no`

  Disables unixODBC statistics which uses system semaphores to keep track of used handles. Many machines do not have sufficient semaphore resources to keep track of statistics and they are only available in the GUI ODBC Administrator anyway.

- `--enable-readline=no`

  This disables `readline` support in `isql`. We disabled this because it ties `isql` to the version of `libreadline` on the machine we build on. We build on as old a version of the operating system we can for upwards compatibility. Many newer Linux machines no longer come with the older readline libraries and so enabling readline support renders `isql` unusable.

- `--prefix=/usr/local/easysoft/unixODBC`

  This installs unixODBC into `/usr/local/easysoft/unixODBC`.

Installing the Easysoft ODBC-Oracle Driver

The Easysoft ODBC-Oracle Driver installation comprises of:

- Installing the driver.
- Registering the driver with the unixODBC driver manager.

If unixODBC is now installed (either installed by this installation or an existing copy was found) the Easysoft ODBC-Oracle Driver will be registered as an ODBC driver with the unixODBC driver manager.

If you already have the Easysoft ODBC-Oracle Driver registered with unixODBC, you will see a warning that it is already registered and a list of the drivers unixODBC knows about. If you are installing the Easysoft ODBC-Oracle Driver into a different directory than it was installed before, you will need to edit your `odbcinst.ini` file after the installation and correct the Driver and Setup paths as unixODBC's `odbcinst` will not update them if a driver is already registered.

- Creating an Easysoft ODBC-Oracle Driver example data source in unixODBC.

  If unixODBC is installed and you registered the Easysoft ODBC-Oracle Driver with unixODBC, an example data source will be added to your odbc.ini file.

  If a data source called "ORACLE" already exists, the existing data source will be displayed and you have the option to replace it.

Licensing

The program *install_path*`/easysoft/license/licshell` is used to obtain or list licenses.

Licenses are stored in the file *install_path*`/easysoft/license/licenses`. After obtaining a license, you should take a copy of this file in case something happens to it.

When you install, the installation will ask you if you want to request an Easysoft ODBC-Oracle Driver license:

```
Would you like to request a Easysoft ODBC-Oracle
Driver license now (y/n) [y]:
```

You do not need to obtain a license during the installation, you can run `licshell` after the installation to obtain or view licenses.

If you answer yes to this, the installation will run the `licshell` script. The process of obtaining a license is best described in the Licensing Guide and on the Easysoft web site.

To obtain a license automatically, you will need to be connected to the Internet and allow outgoing connections to `license.easysoft.com` on port 8884. If you are not connected to the Internet or do not allow outgoing connections on port 8884 then the License Client can create a license request file which you can:

1. Enter in the License Request pages, accessible from the Licensing section of this web page: **http://www.easysoft.com/support**.
2. Mail, fax or telephone to Easysoft.

   Obviously, option 1 is quickest if you have a web browser and access to the Internet.

   Once the License Client has started you are presented with a menu of options which allow you to:

   ```
   [0] exit

   [1] view existing license

   [n] obtain a license for the desired product.
   ```

   Obviously, if you have not got any other Easysoft products licensed then option `[1]` will not show any existing licenses.

   To obtain a license, select one of the options from `[2]` onwards for the product you are installing. The License Client will then run a program that was installed for that product which generates a key that is used to identify the product and operating system (we need this key to license you).

Once you have picked the product to license (Easysoft ODBC-Oracle Driver), you need to supply:

1. Your full name.
2. Your company name.
3. An email contact address. This (currently) **must** be the email address you registered on the Easysoft web site.
4. Your telephone number (you need to specify this if you telephone the license request to us).
5. Your fax number (you need to specify this if you fax the license request to us).
6. A reference number. When applying for a trial license just press <*Enter*> on this field as this field is used to enter a reference number we will supply you for full (paid) licenses.

   You will then be asked for a method of obtaining the license where the choices are:

   [1] Automatically by contacting the Easysoft License Daemon (this requires connection to the Internet and the ability to support an outgoing TCPIP connection to `license.easysoft.com` on port 8884).

   [2] Write information to file so you can:

   a) Obtain your license from the License Request pages on the Easysoft web site. (Accessible from the Licensing section of this web page: **http://www.easysoft.com/support**.)

   b) fax, telephone it.

   The license request is output to `license_request.txt`.

   [3] Cancel request

If you choose to obtain the license automatically, the License Client will start a TCPIP connection to `license.easysoft.com` on port 8884 and send the details you entered at the prompts above and your machine number. No other data is sent. The data sent is transmitted as plain text so if you do not want this information possibly intercepted by someone else on the net you should choose `[2]` and telephone or fax the request to us. The License daemon will return the license key, print it to the screen and make it available to the installation script in the file licenses.out.

If you choose option `[2]` the license request is written to the file license_request.txt and you should exit the License Client via option `[0]` and complete the installation. Once you have mailed, faxed or telephoned the license request to us we will return a license key which should add to the end of the file *install_path*`/easysoft/license/licenses`.

If during this process any warnings or errors are output, please mail the output to **support@easysoft.com** and we will rectify the problem.

POST INSTALLATION

Supplied Documents and Examples

The last part of the installation runs a post install script that lists resources available to you.

Included documentation is installed in *install_path*`/easysoft/oracle/doc`.

- The Easysoft ODBC-Oracle Driver manual in PDF format.
- CHANGES.txt - a list of all the changes in each version.
- The Easysoft ODBC-Oracle Driver license.

There are many resources at the Easysoft web site **(http://www.easysoft.com)**.

Post Installation Steps for Non Root Installations

If you installed the Easysoft ODBC-Oracle Driver as a non `root` user (not recommended), there may be some manual steps you will need to perform:

1. If you attempt to install the Easysoft ODBC-Oracle Driver under the unixODBC driver manager and you do not have write permission to unixODBC's `odbcinst.ini` file, the driver cannot be added.

   You can manually install the driver under unixODBC by adding an entry to the `odbcinst.ini` file. Run `odbcinst -j` to ascertain the `DRIVERS` file then append the lines from the `drv_template` file (in the directory where the distribution was untarred to) to the `odbcinst.ini` file.

2. As in step 1, no example dsns can be added into unixODBC if you do not have write permission to the SYSTEM `odbc.ini` file. Run `odbcinst -j` to ascertain the name of the "SYSTEM DATA SOURCES" file then add your DSNs.

3. On machines where the dynamic linker has a configuration file, defining the locations where it looks for shared objects (Linux/FreeBSD) you will need to add:

   install_path`/easysoft/lib`

   install_path`/easysoft/unixODBC/lib`

   The latter one is only required if you installed the unixODBC included with this distribution. Sometimes after changing the dynamic linker configuration file, you need to run a program to update the dynamic linker cache (for example, `/sbin/ldconfig` on Linux).

4. If you did not install the Easysoft ODBC-Oracle Driver in the default location, you need to do one of the following:

   a) Link `/usr/local/easysoft` to the `easysoft` directory in your chosen install path. For example, if you installed in `/home/martin` the installation will create `/home/martin/easysoft` and you need to symbolically link `/usr/local/easysoft` to `/home/martin/easysoft`:

   ```
   ln -s /home/martin/easysoft /usr/local/easysoft
   ```

   b) Set and export the `EASYSOFT_ROOT` environment variable to *`install_path`*`/easysoft`

5. If your system does not have a dynamic linker configuration file, you need to add the paths listed in step 3 above to whatever environment path the dynamic linker uses to locate shared objects. You may want to amend this in a system file run whenever someone logs in like `/etc/profile`.

   The environment variable differs per dynamic linker. Consult your `ld` or `ld.so man` page. It is usually:

   `LD_LIBRARY_PATH`, `LIBPATH`, `LD_RUN_PATH` or `SHLIB_PATH`.

Uninstalling on Unix

To uninstall the Easysoft ODBC-Oracle Driver under Unix:

- If unixODBC is installed, the Easysoft ODBC-Oracle Driver driver must be removed from its database.
- If the system has a dynamic linker (such as `ld.so` on Linux), the Easysoft ODBC-Oracle Driver directories must be removed from the dynamic linker search path (this may require `root` access, depending on the mechanism used by the platform).
- The Easysoft ODBC-Oracle Driver install directory tree must be removed (this requires the same privileges as the user who performed the installation, which is normally `root`).

A step-by-step guide follows:

1. Log in as `root`.

Removing from unixODBC

2. Check whether the Easysoft ODBC-Oracle Driver is configured under unixODBC by typing:

   ```
   odbcinst -q -d
   ```

3. If "`ORACLE`" is returned in the output then remove the Easysoft ODBC-Oracle Driver entry by typing:

   ```
   odbcinst -u -d -n ORACLE
   ```

   If a message is displayed about a reduced usage count, repeat this step until `odbcinst` states that the Easysoft ODBC-Oracle Driver has been removed.

> **NB** If unixODBC is no longer required then the `odbcinst.ini` and `odbc.ini` files can be removed.

REMOVING FROM THE DYNAMIC LINKER

Notify the dynamic linker that the shared objects are no longer available.

> **NB** This information only applies to systems with the `ld.so` dynamic linker (normally only Linux).

4. If the file `/etc/ld.so.conf` file exists, make a backup copy and open it.
5. Remove the path to the Easysoft ODBC-Oracle Driver shared objects:

   <InstallDir>`/easysoft/oracle`
6. If no other Easysoft software is in use then remove the path to the common Easysoft shared objects:

   <InstallDir>`/easysoft/lib`
7. If unixODBC is no longer required then remove the reference:

   <InstallDir>`/easysoft/unixODBC`
8. Run `/sbin/ldconfig` so that the dynamic linker re-reads the file and will no longer search the removed paths.

DELETING THE SOFTWARE

Finally, remove the software from the system hard drive.

9. Change directory to:

   <InstallDir>`/easysoft/`

   `pwd`

   The system displays the current directory.

Double-check that this is the directory under which you installed the Easysoft ODBC-Oracle Driver.

Caution!

Be very careful issuing the `rm -r` command as `root`. `rmdir` will not remove directories that contain files, but `rm -r` will remove all subdirectories along with their contents. It is possible to destroy your system and/or lose all user files by removing the wrong directory.

10. Remove the Easysoft ODBC-Oracle Driver installation directory:

    ```
    ls
    ```

    Check that you are in the right directory.

    ```
    rm -r oracle
    ```

    The system may ask you to confirm deletion for some files. You can confirm these as long as you are sure you are in the correct directory.

    ```
    ls
    ```

11. If you have no other Easysoft products on your system and you are not using any copy of unixODBC that may be in this directory, then you can delete the `easysoft` directory too.

    ```
    cd ..
    ```

    ```
    rm -r easysoft
    ```

    – OR –

    If there are other files in the directory tree (i.e. you have other Easysoft products installed) then you must not remove the `easysoft` directory, because it will contain your license keys and other important files.

12. If you left the Easysoft ODBC-Oracle Driver distribution file on your system then you may wish to remove it at this point.

    The uninstall process is complete.

    Any licenses you obtain for the Easysoft ODBC-Oracle Driver and other Easysoft products are stored in the `<InstallDir>/easysoft/license/licenses` file.

    After uninstalling the Easysoft ODBC-Oracle Driver, unless you have deleted this file, you will not need to relicense the product when you reinstall or upgrade.

    However, for security purposes you may want to make a copy of `<InstallDir>/easysoft/license/licenses` before uninstalling.

CONFIGURATION

3

Configuring the Easysoft ODBC-Oracle Driver

The Easysoft ODBC-Oracle Driver is usually installed on the machines where your applications are running. ODBC applications access ODBC drivers through the ODBC driver manager and a data source which tells the driver manager which ODBC driver to load, which database to connect to and how to connect to it. This chapter describes how you create data sources, use DSN-less connections and configure the Easysoft ODBC-Oracle Driver.

Before setting up a data source, the Easysoft ODBC-Oracle Driver must have been successfully installed on the machines where your ODBC application is running.

Instructions for installing the Easysoft ODBC-Oracle Driver on Windows and Unix platforms are provided in **"Installation" on page 21**.

Chapter Guide

DSN-less connections

In addition to using a data source, you can also connect to a database by using a DSN-less connection string of the form:

```
SQLDriverConnect(..."DRIVER={ORACLE};DB=pubs;UID=demo;PWD=easysoft;"...)
```

where *pubs* is the database name, *demo* is the user name with which to connect to the database, and *easysoft* is the password for the demo user.

Further Easysoft ODBC-Oracle Driver attribute settings, as described in this section, can also be added to the connection string using the same "PARAMETER=value;" format.

Refer to the section relevant to your platform to continue:

- **"Windows Setup" on page 62**
- **"Unix Setup" on page 75**

OS Authentication

The Easysoft ODBC-Oracle Driver can be configured to connect to an Oracle database using OS Authentication. To do this, you must first ensure that the database and client are configured and working correctly. More details on the use of OS Authentication and its implications can be found in your Oracle documentation. To establish that the system is configured correctly, we recommend using a tool such as sqlplus, before attempting to configure your driver. Once the Oracle software is configured, the driver can be set up by leaving the username and password fields empty.

- **"Windows Setup" on page 62**
- **"Unix Setup" on page 75**

Windows Setup

To connect an ODBC application on a Windows machine to an Oracle database on a remote server:

1. In **Control Panel**, double-click **Administrative Tools** and then **Data Sources (ODBC)**.

9x	In **Control Panel**, double-click **ODBC Data Sources (32bit)**.
NT	In **Control Panel**, double-click **Data Sources (ODBC)**.

64-bit Windows

The Easysoft installer program installs both a 32-bit and a 64-bit version of the Easysoft ODBC-Oracle Driver. If you want to use a 64-bit ODBC application, you need to use the 64-bit Easysoft ODBC-Oracle Driver. If you want to use a 32-bit ODBC application, you need to use the 32-bit Easysoft ODBC-Oracle Driver.

There is both a 32-bit and a 64-bit version of ODBC Administrator. The 64-bit ODBC Administrator is located in Control Panel under Administrative tools. To access the 32-bit ODBC Administrator, in the Windows Run dialog box, type:

```
%windir%\syswow64\odbcad32.exe
```

Easysoft ODBC-Oracle Driver data sources created in the 64-bit ODBC Administrator will specify the 64-bit version of the Easysoft ODBC-Oracle Driver. Easysoft ODBC-Oracle Driver data sources created in the 32-bit ODBC Administrator will specify the 32-bit version of the Easysoft ODBC-Oracle Driver.

For more information about ODBC on 64-bit Windows platforms see:

http://www.easysoft.com/developer/interfaces/odbc/64-bit.html

The **ODBC Data Source Administrator** dialog box is displayed:

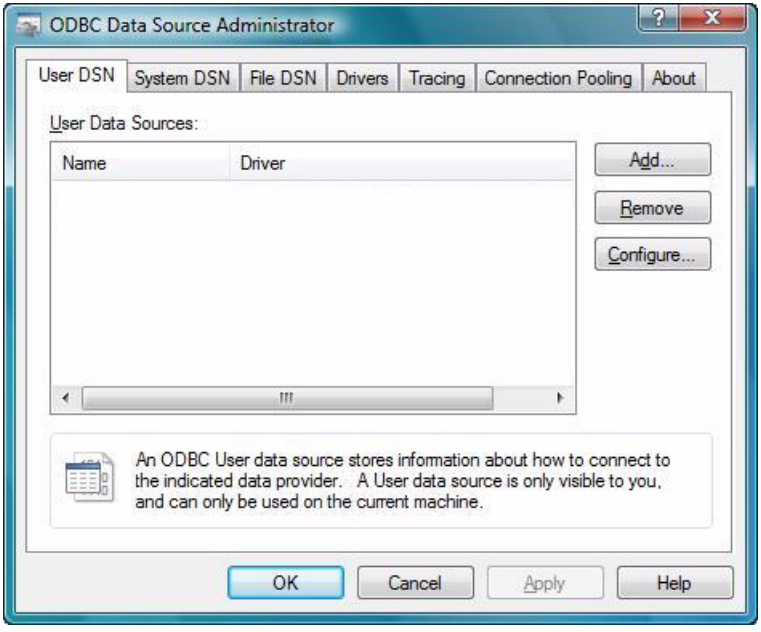


Figure 5: The ODBC Data Source Administrator

2. To create a data source that is only available to the user currently logged into this machine, select the **User DSN** tab.

   – OR –

   To create a data source that is available to any user who logs into this machine, select the **System DSN** tab.

3. Click **Add...** to add a new data source.

   The **Create New Data Source** dialog box displays a list of drivers:

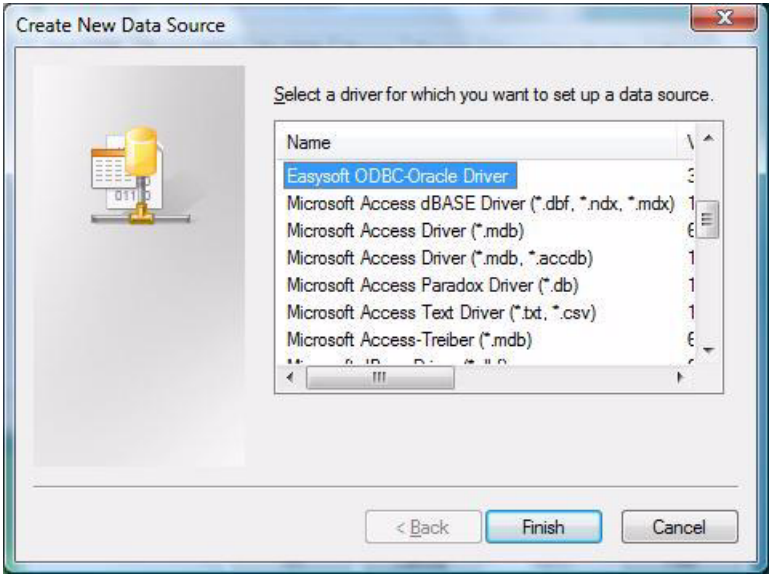


Figure 6: The Create New Data Source dialog box

4. Select `Easysoft ODBC-Oracle Driver` and click **Finish**.

The Easysoft ODBC-Oracle Driver DSN configuration dialog box is displayed.

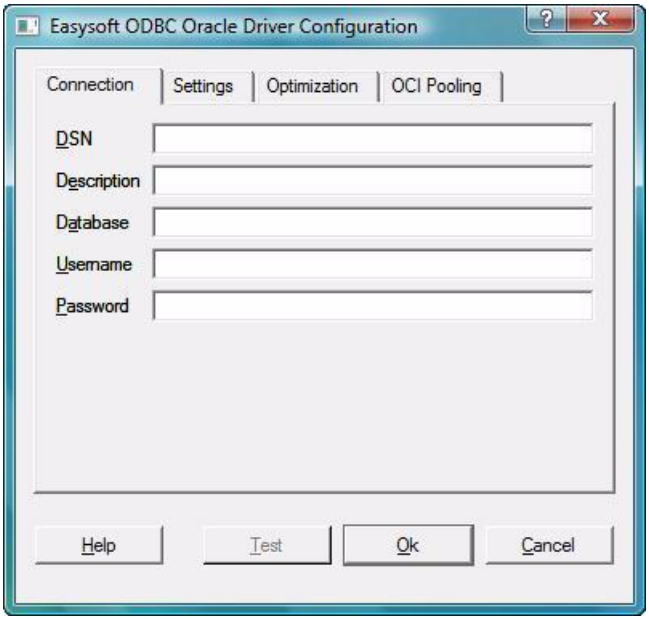


Figure 7: The DSN Setup dialog box - Connection tab

The attributes on this dialog box are split into four tabs, arranged by functionality, from left to right:

- **Connection**, how the data source appears to the driver manager and ODBC application
- **Settings**, ODBC driver specific features
- **Optimization**, ODBC driver performance settings
- **OCI Pooling**, OCI pooling options

Test allows you to check that the client is able to connect to the specified server data source.

CONNECTION

5. In the **DSN** box enter a name for this data source.

   Choose carefully because you will not be able to change this after pressing **OK**.

6. In the **Description** box enter something that would help a user faced with a choice of data sources.

7. Do one of the following:

   - If you are using the Instant Client, in the **Database** box, enter a SQL connect URL string. Use the following format:

     `//host:port/service_name`

     where *`host`* is the fully qualified domain name or IP address of the server on which the Oracle database is installed, *`port`* is the Oracle listener port or the alias name mapped to the port in the `/etc/services` file and *`service_name`* is the local net service name. For example, `//my_host:1521/my_database`.

   – OR –

   - If you are using the standard Oracle Client, in the **Database** box, the logical name used to identify the Oracle target database.

     This is the local net service name defined in your tnsnames.ora file.

8. In the **Username** and **Password** boxes enter the user name and password of a valid account in the Oracle target database.

SETTINGS

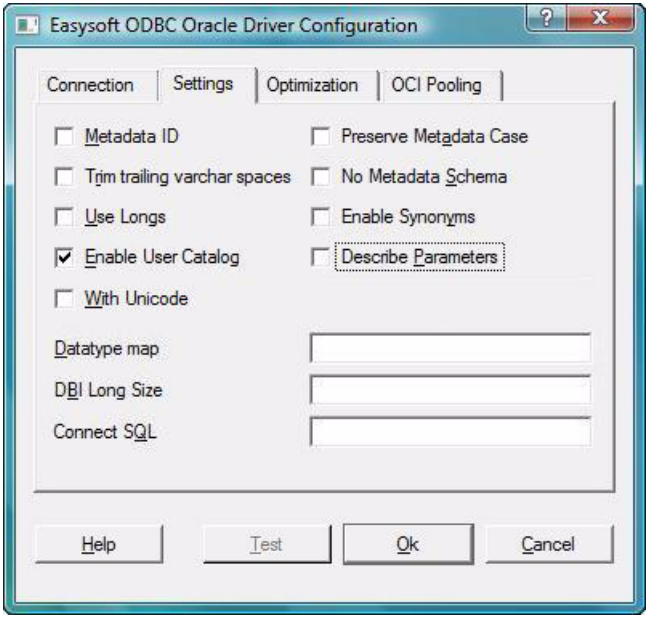


Figure 8: The DSN Setup dialog box - Settings tab

For details of the attributes that can be set on this dialog box, see **"Attribute Fields" on page 81**

OPTIMIZATION

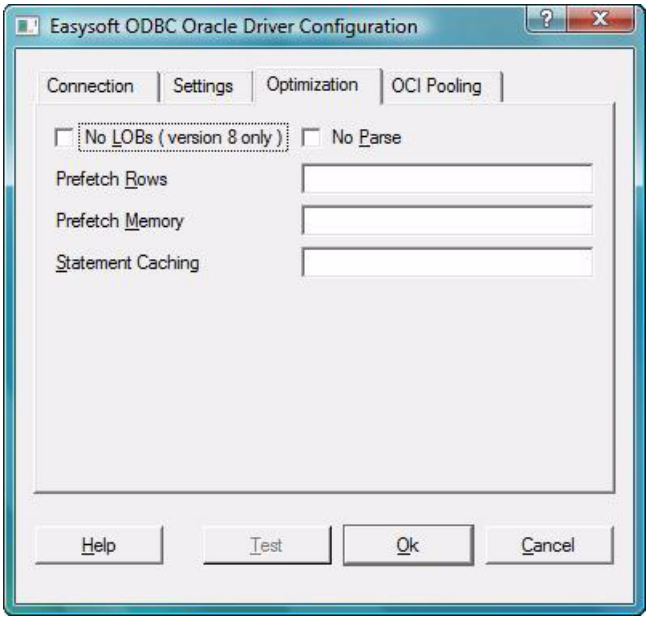


Figure 9: The DSN Setup dialog box - Optimization tab

For details of the attributes that can be set on this dialog box, see **"Attribute Fields" on page 81**

OCI POOLING

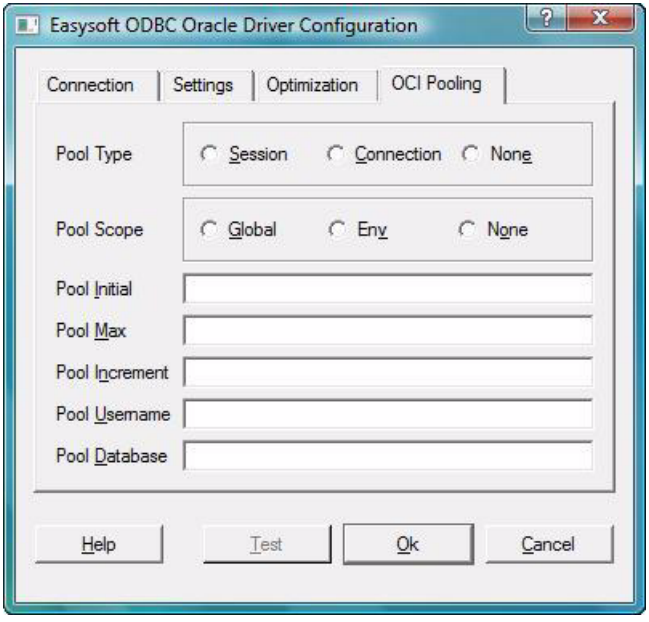


Figure 10: The DSN Setup dialog box - OCI Pooling tab

For details of the attributes that can be set on this dialog box, see **"Attribute Fields" on page 81**

9. Now click **Test**.

   The Easysoft ODBC-Oracle Driver attempts to connect to the database and send an ODBC request, displaying the results in a window.

10. If an error message is displayed then check the entries in the DSN setup fields.

If the target database can be successfully accessed by running SQL*Plus, the problem can be assumed to be on the client machine, rather than on the server machine.

– OR –

An information message will be displayed if you have successfully connected to the server:

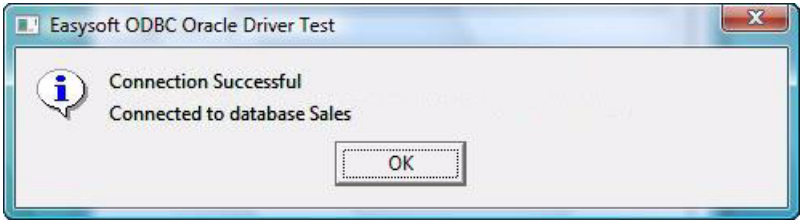


Figure 11: The Easysoft Setup Test DSN dialog box

11. Click **OK** in the **Easysoft Setup Test** dialog box and **OK** in the **Easysoft ODBC-Oracle Driver DSN Setup** dialog box.

    The connection has been made.

CONNECTING A CLIENT APPLICATION

There is now a data source on the Windows machine that connects via the Easysoft ODBC-Oracle Driver to an Oracle database on a remote server machine.

An example ODBC application can now be connected to the local data source to demonstrate that the Easysoft ODBC-Oracle Driver is functioning correctly.

NB	This section requires access to Microsoft Access.

1. Start Microsoft Access (for example) and create a blank database.
2. Do one of the following:
   - In Access 2007, on the **External Data** tab, in the **Import** group, click **More**. Click **ODBC Database**. 
   - In earlier versions of Access, select **File > Get External Data > Link Tables**.

     The **Link** dialog box displays the existing databases on your system.
3. Do one of the following:
   - In Access 2007, click **Link to the data source by creating a linked table**, and then click **OK**.
   - In earlier versions of Access, from the **Files of type** drop-down list, choose **ODBC Databases**.

The **Select Data Source** dialog box displays the existing data sources on your system:

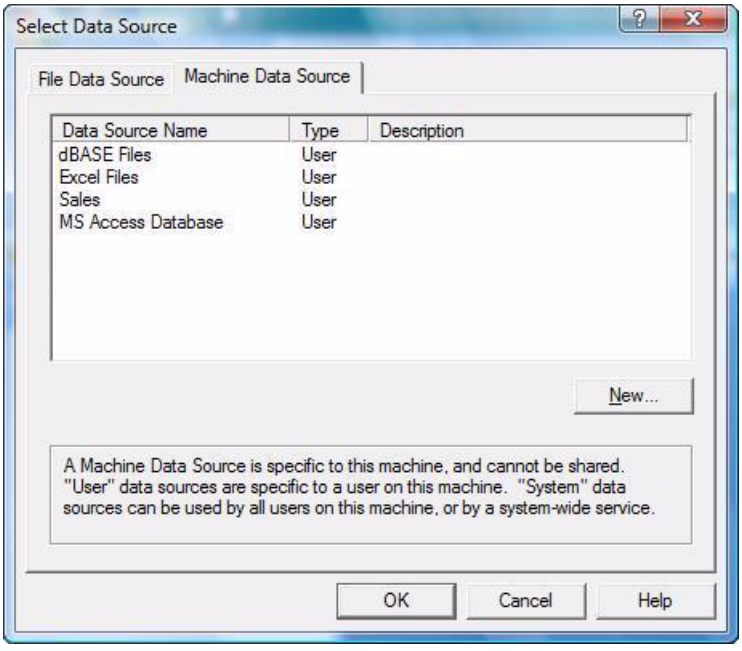


Figure 12: The Select Data Source dialog box Machine Data Source tab

4. Click the **Machine Data Source** tab and find the local data source you created in the list (note that your description of the data source is displayed beside it).
5. Select your data source and click **OK**.

   Microsoft Access connects (through the ODBC Driver Manager) to the Easysoft ODBC-Oracle driver and retrieves a list of available tables.

The **Link Tables** window is displayed, showing a list of available datasets:

6. Click on a table and then click **OK**.

   After a short wait, you are returned to the **Database** window.

7. Double-click on one of the tables to open and browse it.

Unix Setup

Data Source Attributes

There are two options when setting up a data source to your Oracle data:

- Create a system data source (which is available to anyone who logs onto this Unix machine)

– OR –

- Create a user data source (which is only available to the user who is currently logged into this Unix machine)

This is exactly the same mechanism as is used on the ODBC Data Source Administrator on Windows platforms (see **"Windows Setup" on page 62**).

NB	If you have user and system data sources with the same name the driver manager will use the user data source.

The default Easysoft ODBC-Oracle Driver installation will create a system DSN named `[ORACLE]`. If you are using unixODBC that came with this distribution the system odbc.ini file will be `/etc`.

If you have built unixODBC yourself, or installed it from some other source then system data sources are stored in the path specified with the configure option `--sysconfdir`=*directory.* If `sysconfdir` was not specified when unixODBC was configured and built it defaults to `/usr/local/etc`).

User data sources are stored in the file `.odbc.ini` in the current user's home directory (e.g. `$HOME/.odbc.ini`). .

NB	By default, you must be logged in as `root` to edit a system data source defined in `/etc/odbc.ini`.

NB	`odbcinst -j` will show you where the driver manager stores system and user data sources.

You may either change some of the initial configuration parameter values for the sample data source or add extra data sources.

Each section of the `odbc.ini` file starts with a data source name in square brackets [ ] followed by a number of *attribute=value* pairs.

NB	Attribute names in `odbc.ini` are not case sensitive.

For the addition of a data source, all the settings in the following example must be appended to the relevant `odbc.ini` file. The Driver attribute names the ODBC driver in the `odbcinst.ini` file to use for this data source. When the Easysoft ODBC-Oracle Driver is installed into unixODBC it places an ORACLE entry into the `odbcinst.ini` file so you should always have `Driver = ORACLE` in your Easysoft ODBC-Oracle Driver data sources.

NB	More details of configuring data sources with unixODBC can be found at **www.unixodbc.org**.

To configure your Oracle DSN in your `odbc.ini` file, you will need to edit:

- For the Instant Client, the SQL connect URL string (`Database`).

– OR –

- For the standard Oracle Client, the tnsnames.ora service name for the database (`Database`).
- The Oracle database user name (`User`).
- The Oracle database password (`Password`).

Instant Client Example

```
[ORACLE]
Driver = ORACLE
Database = //testhost:1521/testdb
User = system
Password = manager
```

Standard Client Example

```
[ORACLE]
Driver = ORACLE
Database = testdb
User = system
Password = manager
```

Other optional attribute values may be set in the `odbc.ini` file, and are described in **"Attribute Fields" on page 81**.

DRIVER ATTRIBUTES

There are a number of driver attributes that are stored in the `odbcinst.ini` file. By default this file is installed in `/etc`. If you have built unixODBC yourself, then it will be whatever path you specified in the `sysconfdir=`*directory* configure option (if `sysconfdir` has not been specified then the path will default to `/usr/local/etc`).

> **NB** By default, you must be logged in as `root` to edit a system data source defined in `/etc/odbcinst.ini`,.

- **Threaded**

  If the applications using the driver does not use threads, you can set `Threads` to be 0. This will provide a performance enhancement, by disabling thread safety in the driver.

ENVIRONMENT

For the Easysoft ODBC-Oracle Driver to function correctly the following environment variables **must** be created and given values specific to the directory structure on the target machine.

Instant Client

- `LD_LIBRARY_PATH`

  This **must** be the Instant Client software directory. For example:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib/instantclient_10_2
export LD_LIBRARY_PATH
```

Note that on some platforms you need to use `SHLIB_PATH` or `LIBPATH` rather than `LD_LIBRARY_PATH`.

Standard Client

- `ORACLE_HOME`

  This **must** be the Oracle Client software directory. For example:

  ```
  ORACLE_HOME=/home/oracle/OraHome1

  export ORACLE_HOME
  ```

- `LD_LIBRARY_PATH`

  This **must** always be set to the directory `$ORACLE_HOME/lib`. For example:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/oracle/OraHome1/lib

export LD_LIBRARY_PATH
```

  Note that on some platforms you need to use `SHLIB_PATH` or `LIBPATH` rather than `LD_LIBRARY_PATH`.

Easysoft ODBC-Oracle Driver

The Easysoft ODBC-Oracle Driver must also be able to find the following shared objects which it installs:

- `libodbcinst.so`

  By default, this will always be located in `/usr/local/easysoft/unixODBC/lib/`:

- `libeslicshr_r.so or libeslicshr.so`

  By default, this will always be located in `/usr/local/easysoft/lib/`

- `libessupp_r.so or libessupp.so`

  By default, this will always be located in `/usr/local/easysoft/lib`

It may be necessary to set up and export `LD_LIBRARY_PATH`, `SHLIB_PATH` or `LIBPATH` (depending on the operating system and run-time linker) to include the directories where `libodbcinst.so`, `libeslicshr.so` and `libeslicshr.so` are located.

NB	The shared object file extension (.so) varies across operating systems (.so, .a, or .sl).

ESTABLISHING A TEST CONNECTION

Run the `isql` query tool to prove that the Easysoft ODBC-Oracle Driver is working:

1. Change directory into

   `/usr/local/easysoft/unixODBC/bin`

2. Type `./isql -v` *`DSN`*

   where *`DSN`* is the name of the target data source

3. Enter SQL commands to query the database, such as:

   ```
   SQL> select table_name from user_tables
   ```

4. or enter 'help' to return a list of tables

   ```
   SQL> help
   ```

TROUBLE-SHOOTING A TEST CONNECTION

Some Easysoft ODBC-Oracle distributions contain our `checksys` diagnostic tool which can report configuration and environment problems and suggest corrective action. It can be found in the `/usr/local/easysoft/oracle` directory. To run the tool:

```
cd /usr/local/easysoft/oracle
./checksys -d DSN
```

Attribute Fields

This section lists the attributes which can be set for the Easysoft ODBC-Oracle Driver in a table showing:

- The label of the attribute on the Windows dialog box.
- The entry required when editing the Unix `odbc.ini` file.
- The string to be used in a call to `SQLDriverConnect` or in a connect string for ADO type use.

Attributes which are text fields are displayed as "`value`".

Attributes which are logical fields can contain either 0 (to set to off) or 1 (to set to on) and are displayed as "`0|1`".

If an attribute can contain one of several specific values then each possible entry is displayed and separated by a pipe symbol.

For example, in the statement:

```
Data_Type_Map=0|1|2
```

the value entered may be "0", "1" or "2".

DSN

The name of the User or System data source to be created, as used by the application when calling the `SQLConnect` or `SQLDriverConnect` functions.

Interface	Value
DSN Dialog Box (Windows)	DSN
odbc.ini file (Unix)	[value]
Connect String	DSN=value

DESCRIPTION

Descriptive text which may be retrieved by certain applications to describe the data source.

Interface	Value
DSN Dialog Box (Windows)	Description
odbc.ini file (Unix)	Description=value
Connect String	Not Used

DATABASE

If you are using the Instant Client, a SQL connect URL string. Use the following format:

`//host:port/service_name`

where *host* is the fully qualified domain name or IP address of the server on which the Oracle database is installed, *port* is the Oracle listener port or the alias name mapped to the port in the `/etc/services file` and *service_name* is the local net service name. For example, `//my_host:1521/my_database`.

If you are using the standard Oracle Client (or the Instant Client with a `tnsnames.ora` file by setting `TNS_ADMIN`), the logical name used to identify the Oracle target database. This is the local net service name defined in your `tnsnames.ora` file. For example, `my_database`.

Interface	Value
DSN Dialog Box (Windows)	Database
odbc.ini file (Unix)	Database=value
Connect String	DATABASE=value

USERNAME

The name of the user that will be supplied to Oracle to authenticate the connection.

If a value is specified in the connect string then any entry in the DSN will be ignored.

Interface	Value
DSN Dialog Box (Windows)	Username
odbc.ini file (Unix)	User=value
Connect String	UID=value

PASSWORD

The password supplied to Oracle to authenticate the connection

Interface	Value
DSN Dialog Box (Windows)	Password
odbc.ini file (Unix)	Password=value
Connect String	PWD=value

Note that passwords are case sensitive for new or modified accounts in Oracle 11*g*.

If a value is specified in the connect string then any entry in the DSN will be ignored.

METADATA ID

When ON (i.e. set to 1 or checked), the default value of the Connection Attribute `SQL_ATTR_METADATA_ID` is set to `SQL_TRUE` (see **"StarOffice 5.2" on page 124**).

If `SQL_TRUE`, the string arguments of catalog functions are treated as identifiers. The case is not significant. For non-delimited strings, the driver removes any trailing spaces, and the string is folded to uppercase. For delimited strings, the driver removes leading and trailing spaces, and takes literally whatever is between the delimiters.

> **NB** Setting this can cause failures in applications which expect the default to be `SQL_FALSE` and do not check the attribute.

The default for `METADATA_ID` is off (0)

Interface	Value
DSN Dialog Box (Windows)	MetaData ID
odbc.ini file (Unix)	MetaData_ID=0|1
Connect String	METADATA_ID=0|1

PRESERVE METADATA CASE

When ON (i.e. set to 1 or checked), the case of the parameter values passed to metadata calls will not change.

The default for `METADATA_DONT_CHANGE_CASE` is off (0).

Interface	Value
DSN Dialog Box (Windows)	Preserve MetaData Case
odbc.ini file (Unix)	MetaData_Dont_Change_Case=0|1
Connect String	METADATA_DONT_CHANGE_CASE=0|1

TRIM TRAILING VARCHAR SPACES

When ON (i.e. set to 1 or checked), the driver trims trailing spaces from varchar types when passed as bound parameters. If `VarcharTrimTrailingSpaces` is set to 1 then trailing spaces are removed from the end of the data.

The default behaviour is to not trim spaces.

Interface	Value
DSN Dialog Box (Windows)	Trim Trailing Varchar Spaces
odbc.ini file (Unix)	VarcharTrimTrailingSpaces=0|1
Connect String	VARCHARTRIMTRAILINGSPACES=0|1

NO METADATA SCHEMA

When ON (i.e. set to 1 or checked), schema names are not returned by metadata calls. This works around some problem applications which do not handle SCHEMA names properly (see **"OpenOffice.org 1.0" on page 124**)..

Interface	Value
DSN Dialog Box (Windows)	No Metadata Schema
odbc.ini file (Unix)	Metadata_Dont_Do_Schema=0|1
Connect String	METADATA_DONT_DO_SCHEMA=0|1

USE LONGS

When ON (i.e. set to 1 or checked), information on `LONG` data types will be returned in the result set from the `SQLGetTypeInfo` function call.

Restrictions with `LONG` data types in Oracle databases (such as only permitting one column per table to be defined) often cause errors to occur, and this attribute can be used to include `LONG` within the list of valid data types which can be used by an application.

The default for `USE_LONGS` is off (0).

Interface	Value
DSN Dialog Box (Windows)	Use Longs
odbc.ini file (Unix)	Use_Longs=0|1
Connect String	USE_LONGS=0|1

ENABLE SYNONYMS

When ON (i.e. set to 1 or checked), table name synonyms are returned in metadata result sets.

By default synonyms are not returned. If you do not need to see synonyms, leave ENABLE_SYNONYMS set to its default value. Enabling synonyms in metadata calls may greatly increase the size of metadata result sets for ODBC API calls such as `SQLTables`.

Interface	Value
DSN Dialog Box (Windows)	Enable Synonyms
odbc.ini file (Unix)	Enable_Synonyms=0|1
Connect String	ENABLE_SYNONYMS=0|1

ENABLE USER CATALOG

When ON (i.e. set to 1 or checked), this limits the driver to returning metadata (e.g. tables) for the current Oracle user.

The driver default is to see metadata for all users (OFF). Many ODBC applications will never need this amount of catalog data. Setting ENABLE_USER_CATALOG to ON reduces the number of rows returned by SQLTables calls.

Note that the default for data sources configured in the Windows Easysoft ODBC-Oracle Driver dialog box is ON.

Interface	Value
DSN Dialog Box (Windows)	Enable User Catalog
odbc.ini file (Unix)	Enable_User_Catalog=0|1
Connect String	ENABLE_USER_CATALOG=0|1

DESCRIBE PARAMETERS

Oracle does not support the describing of parameters, so the driver does not support the `SQLDescribeParam` ODBC call. However, if this ini setting is set to 1 (or checked) then the driver will describe any parameters as varchar.

The default for `DESCRIBE_PARAM_AS_STRINGS` is off (0).

Interface	Value
DSN Dialog Box (Windows)	Enable User Catalog
odbc.ini file (Unix)	Describe_Param_As_Strings=0|1
Connect String	DESCRIBE_PARAM_AS_STRINGS=0|1

DATATYPE MAP

The following values can be set in order to map Oracle data types onto ODBC data types:

Value	Oracle Data Type	ODBC Data Type
0	NUMBER <= 4 digits	SQL_SMALLINT
	NUMBER <= 9 digits	SQL_INTEGER
	NUMBER = n digits	SQL_NUMERIC
	NUMBER = n,m digits	SQL_NUMERIC
1	NUMBER <= 4 digits	SQL_SMALLINT
	NUMBER <= 9 digits	SQL_INTEGER
	NUMBER = n digits	SQL_DOUBLE
	NUMBER = n,m digits	SQL_DOUBLE

Value	Oracle Data Type	ODBC Data Type
2	NUMBER <= 4 digits	SQL_SMALLINT
	NUMBER = n digits	SQL_INTEGER
	NUMBER <= 19 digits	SQL_BIGINT
	NUMBER n,m digits	SQL_DOUBLE
3	NUMBER <= 4 digits	SQL_SMALLINT
	NUMBER <= 9 digits	SQL_INTEGER
	NUMBER <= 19 digits	SQL_BIGINT
	NUMBER >9 digits	SQL_NUMERIC

The default for `Data_Type_Map` is 0.

Interface	Value
DSN Dialog Box (Windows)	Datatype Map
odbc.ini file (Unix)	Data_Type_Map=0|1|2
Connect String	DATA_TYPE_MAP=0|1|2

DBI Long Size

Any value specified will override the maximum size of a `LONG` column (in bytes).

Perl DBI tries to allocate a buffer the size of a `LONG` column and, as this is rather large, it can cause problems which setting `DBI_LONG_SIZE` can resolve (see **"Perl DBI" on page 124**).

Interface	Value
DSN Dialog Box (Windows)	DBI Long Size
odbc.ini file (Unix)	DBI_Long_Size=value
Connect String	DBI_LONG_SIZE=value

CONNECT SQL

This setting defines an Oracle SQL statement that will be run immediately after the driver has established a connection to the database.

Interface	Value
DSN Dialog Box (Windows)	Connect SQL
odbc.ini file (Unix)	Connect_SQL=value
Connect String	CONNECT_SQL=value

NO LOBS

When ON (i.e. set to 1 or checked), increases the performance of the Easysoft ODBC-Oracle Driver if there are no `CLOB` or `BLOB` data types in use. This is only applicable to Oracle version 8.1.7.

The default is off (0).

Interface	Value
DSN Dialog Box (Windows)	No Lobs
odbc.ini file (Unix)	No_LOBS=0|1
Connect String	NO_LOBS=0|1

NO PARSE

When ON (i.e. set to 1 or checked) this stops the Easysoft ODBC-Oracle Driver from preparsing the SQL (passed to `SQLPrepare` and `SQLExecDirect`) to convert ODBC escapes and parameter markers. Setting this option will provide a small speed increase but will prevent your application from using ODBC escapes sequences and parameter markers.

Interface	Value
DSN Dialog Box (Windows)	No Parse
odbc.ini file (Unix)	No_Parse=0|1
Connect String	NO_PARSE=0|1

PREFETCH ROWS

Indicates the number of rows returned from a single "`fetch`" call made to the server.

For example, if the value of `OCI_ATTR_PREFETCH_ROWS` attribute is set to ten, then ten rows will be fetched from the database server and the next call to `SQLFetch` will not need to make a call to the server as the required row will be held by the client already. The default value is 10. Increasing this value can reduce the number of round trip network calls to the server needed to return result sets from the server at the expense of greater memory use.

Interface	Value
DSN Dialog Box (Windows)	Prefetch Rows
odbc.ini file (Unix)	OCI_Attr_Prefetch_Rows=value
Connect String	OCI_ATTR_PREFETCH_ROWS=value

PREFETCH MEMORY

Indicates the number of bytes of memory used on the client to store records returned from a single `SQLFetch` call made to a driver.

This controls the number of records returned, which will be the total required in order to fill the allocated memory area.

For example, if the available memory can store two rows then the next call to `SQLFetch` will not need to make a call to the server, as the required row will be held by the client already.

Interface	Value
DSN Dialog Box (Windows)	Prefetch Memory
odbc.ini file (Unix)	OCI_Attr_Prefetch_Memory=value
Connect String	OCI_ATTR_PREFETCH_MEMORY=value

`OCI_ATTR_PREFETCH_ROWS` and `OCI_ATTR_PREFETCH_MEMORY` update the Oracle Statement attributes of the same name:

e.g.

```
OCI_ATTR_PREFETCH_ROWS     = 10
```

– OR –

```
OCI_ATTR_PREFETCH_MEMORY   = 100
```

These two settings allow a user to specify as an alternative either the number of rows to be fetched or the amount of memory to be used when fetching data.

NB	These attributes are different methods of specifying the quantity of data returned in one fetch. You should only set one but if you set both attributes OCI_ATTR_PREFETCH_ROWS will take precedence.

Increasing this value can reduce the number of round trip network calls to the server needed to return result sets from the server at the expense of greater memory use.

STATEMENT CACHING

This attribute enables Oracle Statement caching. Oracle statement caching establishes and manages a cache of statements within a session. It improves performance by efficiently using prepared cursors on the Oracle server and eliminating repetitive statement parsing. To enable caching, set this attribute to the size of the required cache, i.e. the attribute value should specify the number of statements to cache. Setting the attribute to 0 will switch statement caching off. For more details on Oracle Statement Caching, please consult you Oracle documentation. The default is no statement caching.

Interface	Value
DSN Dialog Box (Windows)	Statement Caching
odbc.ini file (Unix)	Stmt_Caching=value
Connect String	STMT_CACHING=value

FAKE_CLOB_LENGTH

When connecting to Oracle 10*g* or later from a Unix platform, the Easysoft ODBC-Oracle Driver reports the length of `BLOB`, `BFILE`, `CLOB` data types as "0". The driver does this because for these versions of Oracle, the maximum LOB size is 128 terabytes, which is too large a length for the ODBC API to handle.

To change this default behaviour, set `FAKE_CLOB_LENGTH` to 1. When ON (i.e. set to 1), the Easysoft ODBC-Oracle Driver sets the length to the largest value that the integer used to report the length is capable of holding. (Note that this is the default behaviour for the Easysoft ODBC-Oracle Driver on Windows, which is not affected by `FAKE_CLOB_LENGTH`.)

The default for `FAKE_CLOB_LENGTH` is OFF (0).

Interface	Value
DSN Dialog Box (Windows)	Not available.
odbc.ini file (Unix)	Fake_Clob_Length=0|1
Connect String	FAKE_CLOB_LENGTH=0|1

OCI_UTF_FLAG

When ON (i.e. set to 1), the Easysoft ODBC-Oracle Driver does additional conversion when reading LOB data. The Easysoft ODBC-Oracle Driver does this to compensate for non-conformant `OCILobRead` behaviour when reading multibyte character data. When OFF (i.e. set to 0), the Easysoft ODBC-Oracle Driver assumes that the `OCILobRead` behaviour conforms to the Oracle documentation.

Setting `OCI_UTF_FLAG` to 1 may provide a workaround if you experience problems when reading UTF-8 LOB data in parts (i.e. the buffer size passed to `SQLGetData` is not large enough to hold the entire LOB) and you are using the Instant Client 11.1.

The default for `OCI_UTF_FLAG` is OFF (0).

Interface	Value
DSN Dialog Box (Windows)	Not available.
odbc.ini file (Unix)	OCI_UTF_Flag=0|1
Connect String	OCI_UTF_FLAG=0|1

WITH_UNICODE

When ON (i.e. set to 1), the Easysoft ODBC-Oracle Driver attempts to detect whether the national character set for the current environment is AL16UTF16. If this is the case, the Easysoft ODBC-Oracle Driver:

- Describes `NCHAR`, `NVARCHAR2` or `NCLOB` columns as `SQL_WCHAR`, `SQL_WVARCHAR` and `SQL_WLONGVARCHAR`, when describing columns in a result set.
- Transfers data as 16-bit Unicode when binding parameters, if the SQL data type is `SQL_WCHAR`, `SQL_WVARCHAR` or `SQL_WLONGVARCHAR`.
- If the column type is one of the above, the column is bound to the implementation row descriptor (IRD) expecting the length returned and bound size to be in units of twin bytes.

To check what national character set the Easysoft ODBC-Oracle Driver has detected, set `With_Unicode` to 1, enable Easysoft ODBC-Oracle Driver logging by adding the entry `LOG = /tmp/oracle.log` to your data source, and then run a query against a table containing a `NCHAR`, `NVARCHAR2` or `NCLOB` column. Look in the log file for text similar to:

```
Looking at column of type 1 with charset_id of 2000
against al16utf16_csid = 2000
```

If the `charset_id` and `al16utf16_csid` values do not match, setting `With_Unicode` will have no effect.

The default for `With_Unicode` is OFF (0).

Interface	Value
DSN Dialog Box (Windows)	With Unicode
odbc.ini file (Unix)	With_Unicode=0|1
Connect String	WITH_UNICODE=0|1

POOL TYPE

This is used to define the type of pooling required. This can be SESSION or CONNECTION.

Interface	Value
DSN Dialog Box (Windows)	Pool Type
odbc.ini file (Unix)	Pool_Type=Session|Connection
Connect String	POOL_TYPE=SESSION|CONNECTION

POOL SCOPE

This can be GLOBAL or ENV. This is used to associate the pool with the ODBC environment or make it a global resource.

Interface	Value
DSN Dialog Box (Windows)	Pool Scope
odbc.ini file (Unix)	Pool_Scope=Global|Env
Connect String	POOL_SCOPE=GLOBAL|ENV

POOL INITIAL

This is used to define the number of sessions or connections that are created when the pool is created.

Interface	Value
DSN Dialog Box (Windows)	Pool Initial
odbc.ini file (Unix)	Pool_Initial=value
Connect String	POOL_INITIAL=value

POOL MAX

This is used to specify the maximum number of sessions or connections that the pool can contain.

Interface	Value
DSN Dialog Box (Windows)	Pool Max
odbc.ini file (Unix)	Pool_Max=value
Connect String	POOL_MAX=value

POOL INCREMENT

This is the number that the session or connection count is incremented by when required.

Interface	Value
DSN Dialog Box (Windows)	Pool Increment
odbc.ini file (Unix)	Pool_Increment=value
Connect String	POOL_INCREMENT=value

POOL USERNAME

This is the database username that is to be used to authenticate the sessions or connections.

Interface	Value
DSN Dialog Box (Windows)	Pool Username
odbc.ini file (Unix)	Pool_Username=value
Connect String	POOL_USERNAME=value

POOL PASSWORD

This is the database password that is to be used to authenticate the sessions or connections.

Interface	Value
DSN Dialog Box (Windows)	Pool Password
odbc.ini file (Unix)	Pool_Password=value
Connect String	POOL_PASSWORD=value

POOL DATABASE

This is the name of the database against which the pools are to be created.

Interface	Value
DSN Dialog Box (Windows)	Pool Database
odbc.ini file (Unix)	Pool_DB=value
Connect String	POOL_DB=value

POOL_CONNECTION_CLASS

Database Resident Connection Pooling (DRCP) guarantees that pooled servers are never shared across different users. Setting `Pool_Connection_Class` allows for further separation between the sessions of a given user by defining a connection class. A connection class lets different applications (connecting as the same database user) identify their sessions using a logical name that corresponds to the application. OCI then ensures that sessions belonging to a particular connection class are not shared outside of the connection class.

OCI supports a maximum connection class length of 1024 bytes. The asterisk character (*) is a special character and is not allowed in the connection class name.

Interface	Value
DSN Dialog Box (Windows)	Not available
odbc.ini file (Unix)	Pool_Connection_Class=value
Connect String	POOL_CONNECTION_CLASS=value

POOL_PURITY

Whether the application requests a brand new session or reuses a session from the DRCP pool.

To request a new session, set `POOL_PURITY` to `NEW`.

If you connect to a DRCP-enabled Oracle database server without setting `POOL_PURITY`, sessions are reused. When reusing a session from the pool, the NLS attributes of the server take precedence over that of the client.

Interface	Value
DSN Dialog Box (Windows)	Not available
odbc.ini file (Unix)	Pool_Purity=SELF | NEW
Connect String	POOL_PURITY=SELF | NEW

XA_CONNECTION_STRING

The name of the database specified with the DB field in the `xa_open` string. For example, you specify a database named "payroll" with the following `xa_open` string clause:

```
DB=payroll
```

You also need to specify "payroll" as the value for the `XA_Connection_String` attribute field:

```
XA_Connection_String=payroll
```

`XA_Connection_String` is only necessary if you are using the Easysoft ODBC-Oracle Driver to connect to Oracle in the context of an XA transaction and the Transaction Manager specifies a named database in the `xa_open` string.

Interface	Value
DSN Dialog Box (Windows)	Not available
odbc.ini file (Unix)	XA_Connection_String=value
Connect String	XA_CONNECTION_STRING=value

INTERFACING

4

Using the Easysoft ODBC-Oracle Driver with other software

This chapter lists some third-party programming languages and applications that are commonly used with the Easysoft ODBC-Oracle Driver. Easysoft tutorials are available for the applications and programming languages listed here. Look at the section for your application for a link to the tutorial on the Easysoft web site.

Chapter Guide

Apache/PHP

PHP's ODBC support enables PHP scripts running under Apache (or standalone) to access ODBC data sources. The Easysoft ODBC-Oracle Driver lets Apache/PHP access Oracle databases. For example, use our driver with Apache/PHP to dynamically generate web pages from data stored in Oracle 8, 9*i*, 10*g*, 11*g* and XE databases.

TUTORIAL

- **http://www.easysoft.com/developer/languages/php/apache_odbc.html**

SEE ALSO

- **"PHP/PEAR DB" on page 109**
- **"PHP4" on page 124**

C

The Easysoft ODBC-Oracle Driver lets you develop C applications that access and manipulate data stored in Oracle databases.

TUTORIAL

- **http://www.easysoft.com/developer/languages/c/odbc_tutorial.html**

FreeRADIUS

FreeRADIUS is an open source RADIUS (Remote Authentication Dial-In User Service) server that is used for authentication and accounting various types of network access. FreeRADIUS can use information stored in ODBC databases to authenticate users. The Easysoft ODBC-Oracle Driver enables FreeRADIUS to use Oracle as an authorization database.

TUTORIAL

- **http://www.easysoft.com/applications/freeradius/odbc-authorization.html**

Lotus Notes/Domino

Domino Enterprise Connection Services (DECS) lets Lotus Domino applications access data in non-Domino databases. The external data is then available to Lotus Notes client applications. By using DECS with the Easysoft ODBC-Oracle Driver, your Notes/Domino applications can access data in Oracle databases.

TUTORIAL

- **http://www.easysoft.com/applications/lotus_notes_domino/odbc.html**

Micro Focus COBOL

The Micro Focus OpenESQL interface translates embedded SQL statements into ODBC API calls, making it easy to develop COBOL applications that access different database systems. On Unix platforms, OpenESQL is provided with Micro Focus Server Express.

Use the Server Express OpenESQL Interface with the Easysoft ODBC-Oracle Driver to access Oracle databases from your COBOL applications.

TUTORIAL

- **http://www.easysoft.com/applications/micro-focus-cobol/server-express-openesql-odbc.html**

mxODBC

mxODBC is a database API for the Python scripting language that provides an interface to ODBC data sources. Use mxODBC with the Easysoft ODBC-Oracle Driver to access Oracle databases from your Python scripts.

TUTORIAL

- **http://www.easysoft.com/developer/languages/python/mxodbc.html**

OpenLDAP

OpenLDAP is an open source implementation of the Lightweight Directory Access Protocol (LDAP). LDAP is an open-standard protocol for accessing data stored in an information directory. It lets LDAP-aware client applications search for, add, modify and delete directory entries. For example, a user could use an LDAP client to query a directory server for information about specific users, computers, departments or any other information stored in the directory.

By using the Easysoft ODBC-Oracle Driver with OpenLDAP and the SQL backend, back-sql, you can expose data in Oracle databases to LDAP client applications.

TUTORIAL

- **http://www.easysoft.com/applications/openldap/back-sql-odbc.html**

OpenOffice.org

OpenOffice.org is the open source project through which Sun Microsystems has released the technology for the StarOffice Productivity Suite. OpenOffice.org can use ODBC data sources in its various applications to link to external data.

The Easysoft ODBC-Oracle Driver lets OpenOffice.org applications such as Base, Calc and Writer link to external data in Oracle databases.

TUTORIAL

- **http://www.easysoft.com/applications/openoffice_org/odbc.html**

SEE ALSO

- **"OpenOffice.org 1.0" on page 124**

Perl DBI DBD::ODBC

DBI is the database interface module for Perl. DBD::ODBC is the ODBC database driver for Perl DBI. When DBD::ODBC is built with an ODBC driver manager, it enables access to ODBC drivers. You can then use DBI/DBD::ODBC with the Easysoft ODBC-Oracle Driver to access Oracle databases from your Perl scripts.

TUTORIAL

- **http://www.easysoft.com/developer/languages/perl/dbi_dbd_odbc.html**

SEE ALSO

- **"Perl DBI" on page 124**

PHP/PEAR DB

PEAR (PHP Extension and Application Repository) is a framework and distribution system for reusable PHP classes, libraries, and modules. The PEAR DB package is a database abstraction layer that enables PHP programs written for one database to work with other databases.

The PEAR DB module's ODBC backend lets you access databases for which an ODBC driver is available. By using PEAR DB's ODBC backend with the Easysoft ODBC-Oracle Driver you can access Oracle databases from DB on Unix and Linux.

TUTORIAL

- **http://www.easysoft.com/developer/languages/php/pear-db-odbc.html**

SEE ALSO

- **"PHP/PEAR DB" on page 109**

QT

QT is a multiplatform C++ GUI toolkit. Building QT with ODBC support enables you to access ODBC data sources from QT. The Easysoft ODBC-Oracle Driver lets you access data stored in Oracle databases from your QT programs.

TUTORIAL

- **http://www.easysoft.com/developer/libraries/qt/odbc.html**

Rexx/SQL

Rexx/SQL provides Rexx programmers with a consistent, simple and powerful interface to SQL databases.

Use Rexx/SQL with the Easysoft ODBC-Oracle Driver to access Oracle databases from Rexx scripts.

TUTORIAL

- **http://www.easysoft.com/developer/languages/rexx/rexx_sql_odbc.html**

StarOffice

StarOffice is a multi-platform office productivity suite. StarOffice can use ODBC data sources in its various applications to link to external data.

The Easysoft ODBC-Oracle Driver lets StarOffice applications such as Base, Calc and Writer link to external data in Oracle databases.

TUTORIAL

- **http://www.easysoft.com/applications/staroffice/odbc.html**

SEE ALSO

- **"StarOffice 5.2" on page 124**

TECHNICAL REFERENCE

A

Technical Reference for the Easysoft ODBC-Oracle Driver

This section contains extra information relating to the deployment of the Easysoft ODBC-Oracle Driver.

Appendix Guide

ODBC Conformance

The Easysoft ODBC-Oracle Driver is an ODBC 3.5 driver and uses the driver manager to provide the deprecated and old ODBC 2 functions:

- The driver conforms to all the Core Level requirements.
- The driver conforms to 101, 103, 104, 105, 108 and 109 of Level 1 requirements
- The driver conforms to 202 (See **"Describe Parameters" on page 88**), 203, 205 and 211 of Level 2 requirements
- The driver implements all ODBC 3 functions with the exception of:

  `SQLDescribeParam` (not supported by default but see **"Describe Parameters" on page 88**)

- The driver partially supports `SQLSetPos`. An application can use the driver to specify a cursor position by calling `SQLSetPos` with the SQL_POSITION argument.

Cursor support in the Easysoft ODBC-Oracle Driver is restricted to FORWARD_ONLY and STATIC.

SCALAR FUNCTIONS

The Easysoft ODBC-Oracle Driver supports the following scalar functions:

- BIT_LENGTH
- CHAR
- CHAR_LENGTH
- CHARACTER_LENGTH
- LCASE
- OCTET_LENGTH
- UCASE
- CEILING

Functions need to be called by using the ODBC escape sequence {fn *scalar-function*}. For example:

```
SELECT {fn UCASE(Name)} FROM Customers
```

Oracle 8, 9, 10*g*, 11*g* and XE

The Easysoft ODBC-Oracle Driver will work with Oracle 8.1.7+, 9*i*.x 10*g* Release 1+, 11*g* Release 1+ and Oracle Database XE.

No changes to the driver setup are required if upgrading the Oracle Client software.

Restrictions

The Easysoft ODBC-Oracle Driver will allow access to Oracle databases on version 8.1.7 and upwards.

Supported Data Types

The following Oracle data types are supported by the Easysoft ODBC-Oracle Driver:

- BLOB
- BFILE
- RAW
- CLOB
- CHAR
- NUMBER

> **NB** To control how NUMBER data types are mapped onto ODBC data types, use the DATA_TYPE_MAP parameter. For more information, see **"Datatype Map" on page 88**.

- BINARY_FLOAT
- BINARY_DOUBLE

> **NB** BINARY_FLOAT and BINARY_DOUBLE data types are only available if you are using the Oracle 10*g* client or later to connect to Oracle 10*g* or later.

- DECIMAL
- INTEGER
- FLOAT
- DOUBLE PRECISION
- VARCHAR2
- TIMESTAMP
- TIMESTAMP WITH TIME ZONE

- `TIMESTAMP WITH LOCAL TIME ZONE`

> **NB** The `TIMESTAMP` data types are not supported by Oracle 8 and the retrieval of `TIMESTAMP` data requires the installation of both Client and Server of version 9+.

- `DATE`

Advanced Security

The Oracle Advanced Security option is an Oracle client or server add-on that combines network encryption, database encryption and strong authentication to protect sensitive data stored in Oracle databases. The Advanced Security option:

- Guarantees data integrity by detecting whether it has been modified during transmission.
- Encrypts data using encryption standards such as RSA or DES to ensure data privacy.
- Supports third party authentication services such as Kerberos and RADIUS.

Oracle Advanced Security requires Net8 or Oracle Net to transmit data securely. The Easysoft ODBC-Oracle Driver uses the Oracle client, which uses these protocols to communicate with the Oracle database server over the network. Applications that access Oracle by using the Easysoft ODBC-Oracle Driver can therefore take advantage of the Oracle Advanced Security option.

For information about configuring the Oracle client and server for use with the Oracle Advanced Security option, see the Oracle Advanced Security Administrator's Guide. No additional Easysoft ODBC-Oracle Driver configuration is necessary.

Oracle Real Application Clusters (RAC)

Real Application Clusters (RAC) is an Oracle High Availability feature that enables an Oracle Database Server Grid by providing a single database that spans multiple low-cost servers yet appears to the application as a single, unified database system. RAC combines the processing power of these multiple interconnected computers to provide system redundancy, scalability, and high availability. Application scale in a RAC environment to meet increasing data processing demands without changing the application code.

To increase the performance of a RAC database, you can add cluster nodes. Each additional node can help speed up application processing.

If a clustered server fails, the Oracle database will continue running on the surviving servers. If more processing power is needed, another server can be added without interrupting user's access to data.

The Easysoft ODBC-Oracle Driver will allow access to Oracle RAC environments. The Oracle client must be configured correctly to take full advantage of the RAC features. Please consult you Oracle documentation for more details.

Transparent Application Failover (TAF)

Transparent Application Failover (TAF) is a mechanism that automatically reconnects client applications to a node of a RAC database cluster following a connection failure. If a failure occurs, the Oracle client intercepts the resultant error message and starts the transparent failover process. The Oracle client requests another connection from the Oracle listener, which then connects the client to a surviving node of the RAC database cluster.

There may be a delay associated with failing over to another node. To keep users informed, it is possible for an OCI application to register a callback function that is invoked in the event of a connection loss and during the course of the failover. The callback function enables the OCI application to advise users that a failover is in progress and to wait while the failover completes. This ensures users do not attempt to restart their applications, because they perceive this delay as a application failure.

The Easysoft ODBC-Oracle Driver enables ODBC applications to register a failover callback function. To do this, the ODBC application must:

1. Define a callback function that takes the form:

```
int TAF_callback_fn( SQLHANDLE connection,
    int type,
    int event );
```

where:

- `connection` is the ODBC connection handle.
- `type` is the Oracle failover type, which tells the callback function what type of failover the client has requested. The failover types are `OCI_FO_NONE`, `OCI_FO_SESSION`, `OCI_FO_SELECT` and `OCI_FO_TXNAL`.
- `event` is the type of Oracle failover event that took place, which tells the callback function why it was called. The failover events are `OCI_FO_END`, `OCI_FO_ABORT`, `OCI_FO_REAUTH`, `OCI_FO_BEGIN` and `OCI_FO_ERROR`.

The function can trigger a new failover attempt by returning `OCI_FO_RETRY`.

All the `OCI_*` constants listed here are defined in the OCI header file oci.h. Consult the Oracle Call Interface Programmer's Guide for the meanings of these constants.

2. Register and establish a context for the callback function by calling `SQLSetConnectAttr` with the attributes `SQL_ATTR_REGISTER_TAF_HANDLE` and `SQL_ATTR_REGISTER_TAF_CALLBACK`.

   The definitions for these attributes are:

   ```
   #define SQL_ATTR_REGISTER_TAF_CALLBACK      1280
   #define SQL_ATTR_REGISTER_TAF_HANDLE        1281
   ```

   The value `SQL_ATTR_REGISTER_TAF_CALLBACK` is a pointer to the callback function. The value for `SQL_ATTR_REGISTER_TAF_HANDLE` is a pointer to the connection handle used to establish a context for the callback function. For example:

   ```
   SQLSetConnectAttr(dbc,
       1280 /* SQL_ATTR_REGISTER_TAF_CALLBACK */,
       &TAF_callback_fn,
       SQL_IS_POINTER);

   SQLSetConnectAttr(dbc,
       1281 /*SQL_ATTR_REGISTER_TAF_HANDLE*/,
       &dbc,
       SQL_IS_POINTER);
   ```

   Version 1.39 of the Perl DBD::ODBC module (which in conjunction with Perl DBI provides an interface to ODBC databases for Perl) has been used to test the Easysoft ODBC-Oracle Driver's TAF support.

An example Perl script is provided that shows how to use the failover types and events to:

- Keep the user informed throughout the duration of the failover.
- Abort the failover.

The example Perl script is available at:

http://search.cpan.org/~mjevans/DBD-ODBC-1.39/ODBC.pm#odbc_taf_callback

Database Resident Connection Pooling (DRCP)

Database Resident Connection Pooling (DRCP) is a scalability feature introduced in Oracle 11*g* Release 1, which uses a combination of dedicated server and connection broker to handle short, transient sessions coming from Web applications.

DRCP is especially relevant for architectures with multi-process, single-threaded application servers, such as PHP and Apache, that cannot do middle tier connection pooling.

The OCI client libraries enable applications to configure the behaviour of DRCP. Applications can:

- Request a brand new session if they cannot reuse a session from the pool.
- Specify a connection class that indicates that the application is willing to reuse a pooled server, which was used by other applications using the same connection class.

For example, applications in an HR suite may be willing to share pooled servers among themselves but not among other applications.

Because the Easysoft ODBC-Oracle Driver uses Oracle client software, it makes it possible for ODBC applications and interfaces to configure behaviour normally controlled from within the OCI layer. For example, the Easysoft ODBC-Oracle Driver enables PHP applications using the Unified ODBC interface to configure the DRCP pool purity or specify a connection class.

To configure the DRCP pool purity or specify a connection class from an ODBC application, set the `Pool_Purity` or `Pool_Connection_Class` attributes. For more information on these attributes see **"Attribute Fields" on page 81**. (Note that to set these attributes, you need to be using version 11.1+ of the Oracle client software.)

BACKGROUND

A connection pool is a cache of database connection objects. The objects represent physical database connections that can be used by an application to connect to a database.

Connection pools promote the reuse of connection objects and reduce the number of times that connection objects are created. Connection pools significantly improve performance for database intensive applications because creating connection objects is costly both in terms of time and resources.

The connection pool is normally configured with a shared pool of physical connections, translating to a back-end server pool containing an identical number of dedicated server processes.

Applications that can use connection pooling include middle tier applications for Web application servers and e-mail servers. (Web applications introduced the three-tier model in which the browser is the client tier, the database is the backend tier, and the web server and its extensions are the middle tier.)

Connection pooling is beneficial only if the middle tier itself is multithreaded, because it takes advantage of the ability of multiple threads in one application process to share resources. (Threads are lightweight processes that exist within a larger process.)

DRCP is an alternative connection pooling mechanism that enables multi-process applications to share connections to the database. (In multi-process applications, unlike in multi-threaded applications, processes are insulated from each other.)

Network Protocols

The Easysoft ODBC-Oracle Driver supports both IPC and TCP/IP network protocols.

Materialized Views

The Easysoft ODBC-Oracle Driver supports materialized views. A materialized view is a database object that contains the results of a query. Materialized views stored in the same database as their base tables can improve query performance through query rewrites.

The query rewrite mechanism reduces response time for returning results from the query. It does this by automatically rewriting the SQL query to use the materialized view instead of accessing the original tables. Query rewrites are particularly useful in a data warehouse environment.

For more information about materialized views, see your Oracle documentation.

XA Support

The Easysoft ODBC-Oracle Driver can be configured to take part in a distributed XA transaction. To do this, add a data source that contains entries that correspond with the `xa_open` string used by the XA Transaction Manager to connect to the Oracle database server.

For example, the following sample `xa_open` string opens an XA connection to an Oracle database named "payroll." It also specifies the Net8 link and the username and password used to log onto the Oracle server.

```
Oracle_XA+sqlnet=ninetwo.oracle+SesTm=35+Acc=P/system/manager+Threads=true+DB=payroll
```

To make this XA connection available for use by the Easysoft ODBC-Oracle Driver, the data source used to access Oracle needs to contain the following corresponding entries.

```
[ORACLE-XA]
Driver                  = ORACLE
Database                = ninetwo.oracle
User                    = system
Password                = manager
XA_Enlist               = 1
XA_Connection_String    = payroll
```

In addition, the `XA_Enlist` attribute must be set to 1. When this setting is present, the Easysoft ODBC-Oracle Driver accesses Oracle by using the Oracle XA library. Note that the data source can then only be used to access an Oracle database as an XA resource under the control of a Transaction Manager. If you need to access the same database with a non-XA connection, configure a separate data source without the `XA_Enlist` attribute.

The `XA_Connection_String` attribute is only required if the `DB` field is present in the `xa_open` string. The `XA_Connection_String` attribute value must be the same as that of the `DB` field.

A tutorial that provides more information about using the Easysoft ODBC-Oracle Driver in the context of an XA transaction is available from the Easysoft ODBC-Oracle Driver section of the Easysoft web site. The tutorial includes a code sample that shows the point at which an ODBC connection needs to be created and closed to participate in a distributed transaction.

Application Specific Issues

StarOffice 5.2

If problems occur with metadata, the `METADATA_ID` parameter needs to be set to 1 in the `odbc.ini` file.

OpenOffice.org 1.0

For OpenOffice.org 1.0, you need to check the No Metadata Schema check box in the dialogue or set METADATA_DONT_DO_SCHEMA to 1 in your odbc.ini file. This is not necessary for OpenOffice.org 2.0.

PHP4

PHP4 calls the `SQLDescribeParams` ODBC function when processing SQL statements that have parameters.

The Easysoft ODBC-Oracle Driver correctly reports that it does not support the `SQLDescribeParams` function, but PHP4 ignores this and continues with undefined results. See **"Describe Parameters" on page 88**.

Perl DBI

If attempting to fetch LONG data, set the `DBI_LONG_SIZE` to an appropriate value. See **"DBI Long Size" on page 89**.

Threading

The Easysoft ODBC-Oracle Driver is thread safe in accordance with the ODBC 3.5 specification and can be used behind threaded applications with confidence. Usually applications use one connection handle and multiple threads executing SQL statements on that connection.

Tracing

The ODBC calls an application makes can be traced:

- within the driver manager by an application
- from within the driver manager
- from within the Easysoft ODBC-Oracle Driver

Within the driver manager by an application

An application can turn tracing on in the driver manager via the ODBC API `SQLSetConnectAttr` (...,`SQL_ATTR_TRACE`,...).

The trace filename may also be specified with the `SQLSetConnectAttr` attribute `SQL_ATTR_TRACEFILE`.

From within the driver manager

Tracing in the driver manager is platform-specific:

- On Windows:

  Start the ODBC driver manager administration interface via **Start Menu > Control Panel > ODBC Data Sources**.

Click on **Tracing**, ensure the specified filename is valid and click **Start Tracing Now**.

- On Unix:

  For the unixODBC driver manager, add two attributes to the [ODBC] section (create one if none exists) in `odbcinst.ini`.

  ```
  Trace = Yes|No
  ```

  ```
  TraceFile = /path/filename
  ```

  e.g.

  ```
  [ODBC]
  ```

  ```
  Trace = Yes
  ```

  ```
  TraceFile = /tmp/sql.log
  ```

  Ensure that the user who is running the application to be traced has write permission to `TraceFile` (and to the directory containing it), or no tracing information will be produced.

FROM WITHIN THE EASYSOFT ODBC-ORACLE DRIVER

Tracing in the Easysoft ODBC-Oracle Driver is platform-specific:

- On Windows:

  Update the **Registry** by running `regedit` and edit the key:

  `HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\`*`DSN-NAME`*

  where *`DSN-NAME`* is the name of the Easysoft ODBC-Oracle Driver data source.

Add a `LOG` string value and set it to the path and file name of the log file (e.g. `\temp\oracle.log`).

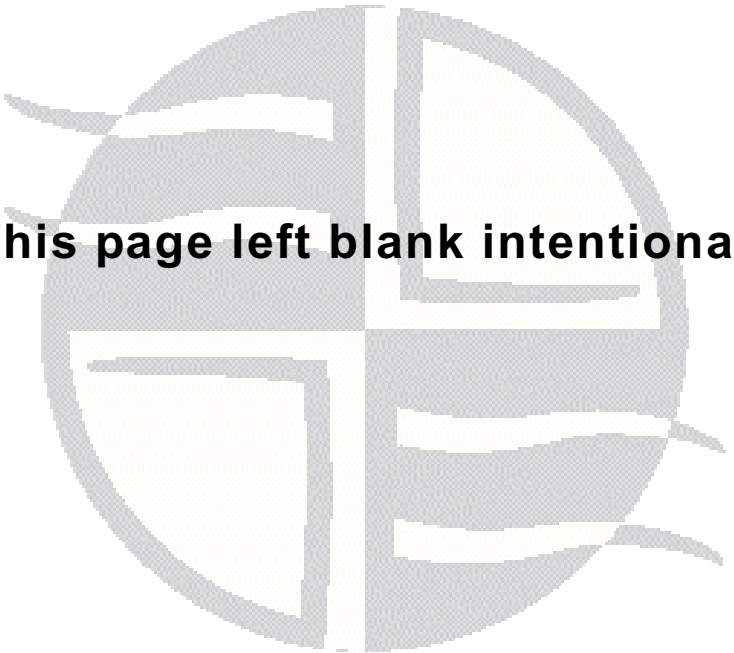
- On Unix:

  Add a `LOG` attribute to the DSN section of the `odbc.ini` file.

  e.g.

  ```
  [ORACLE]
  .
  .
  .
  LOG = /tmp/oracle.log
  ```

  The `LOG` value is the path and file name of the log file (e.g. `/tmp/oracle.log`).



This page left blank intentionally

GLOSSARY

B

Terms and definitions

API (Application Programmer Interface)

A published set of function calls and constants allowing different programmers to utilize a ready-written library of subroutines.

Application

A program that applies the computer to solving some real-world problem. In ODBC terms, it is the program connecting to the data source.

Authorization code

You must have an authorization code for the Easysoft product you wish to license in order to obtain a purchased license. When you purchase a product your authorization code is emailed to you. You do not need an authorization code to obtain a trial license.

Client

A process performing tasks local to the current user, for example, formatting and displaying a report from data retrieved from the server.

Client/Server

The architecture whereby one process (the server) keeps track of global data, and another task (the client) is responsible for formatting and presenting the data. The client connects to the server and requests queries or actions be performed on its behalf. Often these processes run on different hosts across a local-area network.

Column

The vertical dimension of a table. Columns are named and have a domain (or type).

Data source

In ODBC terms, a data source is a database or other data repository coupled with an ODBC Driver, which has been given a Data Source Name (see **“DSN” on page 131**) to identify it to the ODBC Driver Manager.

Data type

The specification of permitted values. A data type limits the values which are allowed to be used.

DBMS

Database Management System - software that handles access to a database.

Download

To retrieve data from a remote machine (or the Internet) to your local machine. Mechanisms for achieving this include FTP and the World Wide Web.

Driver

See **“ODBC driver” on page 132**.

Driver Manager

Software whose main function is to load ODBC drivers. ODBC applications connect to the Driver Manager and request a data source name (DSN). The Driver Manager loads the driver specified in the DSN's configuration file. In Windows, the ODBC Data Source Administrator is used to set up the Driver Manager.

DSN

Data Source Name. A name associated with an ODBC data source. Driver Managers, such as unixODBC or the Microsoft Windows Driver Manager, use the Data Source Name to cross-reference configuration information and load the required driver.

Field

A placeholder for a single datum in a record, for example you can have a Surname field in a Contact Details record. Fields are sometimes referred to as cells.

FTP

File Transfer Protocol. A standard method of transferring files between different machines.

Host

A computer visible on the network.

HTTP

HyperText Transfer Protocol. The means of transferring web pages.

Middleware

Software that is placed between the client and the server to improve or expand functionality.

License key

A string which is provided by Easysoft for use in the licensing process.

ODBC

Open Data Base Connectivity - a programming interface that enables applications to access data in database management systems that use Structured Query Language (SQL) as a data access standard.

ODBC driver

Software that accesses a proprietary data source, providing a standardized view of the data to ODBC.

Row

The horizontal dimension of a table. At its most basic, a row equates to a record within a file.

OCI

Oracle Call Interface - a set of low-level APIs (Application Programming Interface Calls) used to interact with Oracle databases. It allows the use of operations like logon, execute, parse, fetch, etc. OCI programs are normally written in C or C++, although they can be written in almost any programing language. Unlike with the Oracle Precompilers (like Pro*C and Pro*COBOL), OCI programs are not precompiled.

ORACLE_HOME

The environment variable which contains the path where Oracle software has been installed.

Server

A computer, or host, on the network, designed for power and robustness rather than user-friendliness and convenience. Servers typically run around-the-clock and carry central corporate data.

OR

A process performing the centralized component of some task, for example, extracting information from a corporate database.

SQL

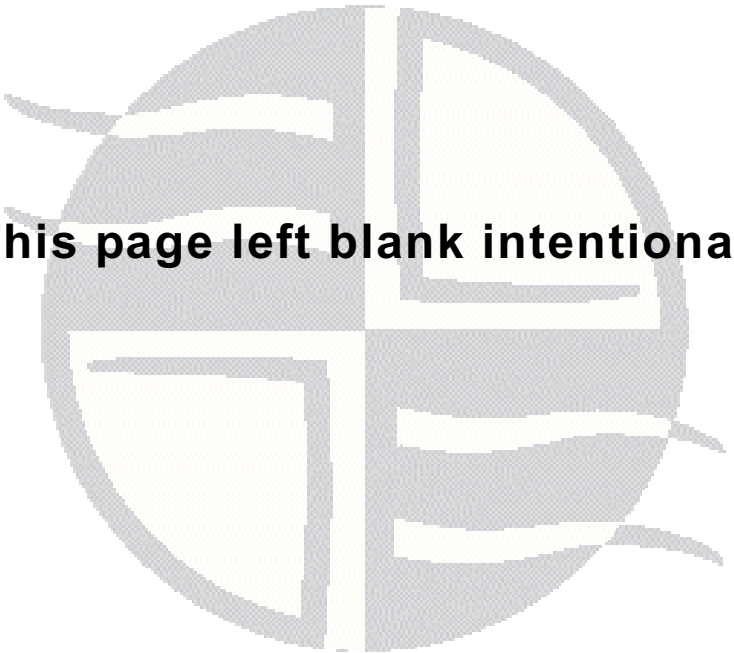
Structured Query Language - an international standard text language for querying and manipulating databases.

Table

A data set in a relational database, composed of rows and columns.

TCP/IP

Transmission Control Protocol/Internet Protocol - a standard method of accessing data on different machines.



This page left blank intentionally

Symbols

A

B

C

D

E

F

G

I

J

L

M

N

O

P

R

S

T

U

V

W

X