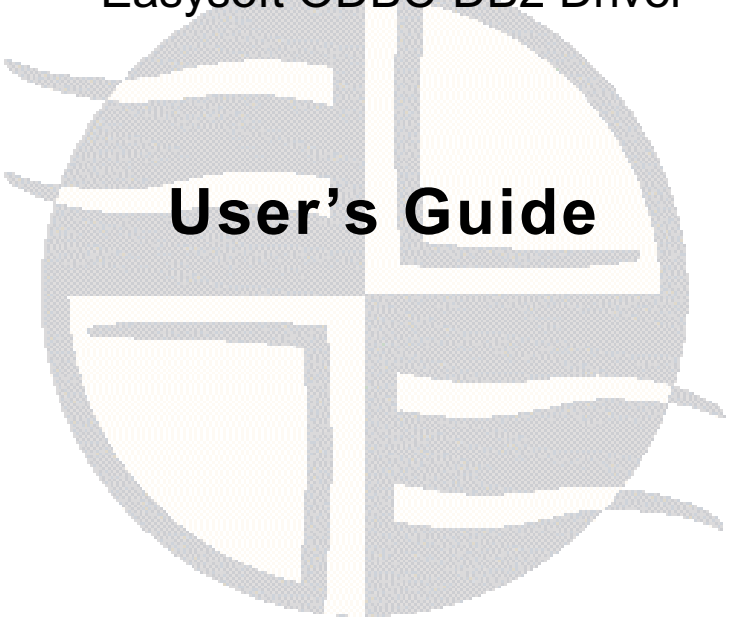
Easysoft Data Access

Easysoft ODBC-DB2 Driver



User's Guide

This manual documents version 1.0.n of the Easysoft ODBC-DB2 Driver.

Publisher: Easysoft Limited

Thorp Arch Grange

Thorp Arch

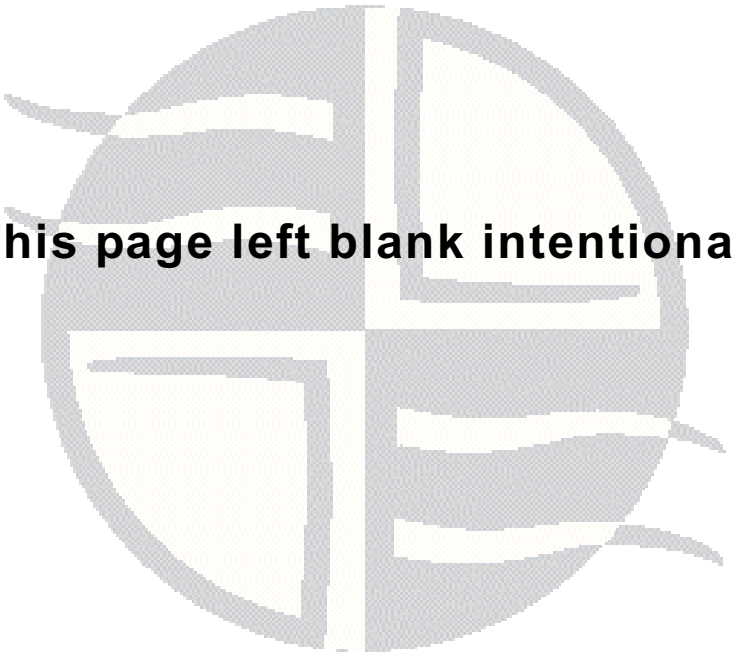
Wetherby

LS23 7BA

United Kingdom

CONTENTS

LIST OF FIGURES



This page left blank intentionally

PREFACE

About this manual

This manual is intended for use by anyone who wants to install the Easysoft ODBC-DB2 Driver, configure it, and then access DB2 data sources from an ODBC-enabled application.

Chapter Guide

Intended Audience

The Unix-based sections require experience of using Unix shell commands. You need to be able to do basic tasks such as editing text files. More complex tasks are described in detail, but it helps to understand how your system handles dynamic linking of shared objects.

Displaying the Manual

This manual is available in the following formats:

- Portable Document Format (PDF), which can be displayed and printed by using the Adobe Reader, available free from Adobe at **http://www.adobe.com**.
- HTML.

Notational Conventions

A *note box* provides additional information that may further your understanding of a particular topic in this manual:

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

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

Linux	On Linux, you must log on as the `root` user to make many important changes.

A *caution box* provides 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!

Typographical Conventions

This manual uses the following typographical conventions:

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

  Click **Next** to continue.

- Commands to be typed are displayed in a `monotype` font, for example:

  At the command prompt, type `admin`.

- File listings and system names (such as file names, directories and database fields) are displayed in a `monotype` font.

Contents

Trademarks

Throughout this manual, *Windows* refers generically to Microsoft Windows 2000, XP, 2003 Server, Vista, 2008 server, 7 or 8, 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*.

DB2 is a registered trademark of International Business Machines Corporation in the United States or other countries or both.

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’.

Mac OS is a trademark of Apple Computer, Inc., registered in the U.S. and other countries.

Easysoft and Easysoft Data Access are trademarks of Easysoft Limited.

CHAPTER 1 INTRODUCTION

Introducing the Easysoft ODBC-DB2 Driver

The Easysoft ODBC-DB2 Driver is an ODBC 3.52 driver for DB2. It lets ODBC-enabled applications access DB2 databases from Linux, Unix and Windows platforms.

Chapter Guide

Overview

The Easysoft ODBC-DB2 Driver connects ODBC-enabled applications on Linux, Unix and Windows to DB2 databases. For example, access DB2 databases from Apache, ApplixWare, Informatica, Apache OpenOffice, LibreOffice, OpenOffice.org and StarOffice. In addition, the Easysoft ODBC-DB2 Driver supports the Perl DBI and DBD::ODBC modules, PHP, PEAR DB, the Python pyodbc and mxODBC interfaces, C and any other ODBC-enabled programming language or interface.

Product Status

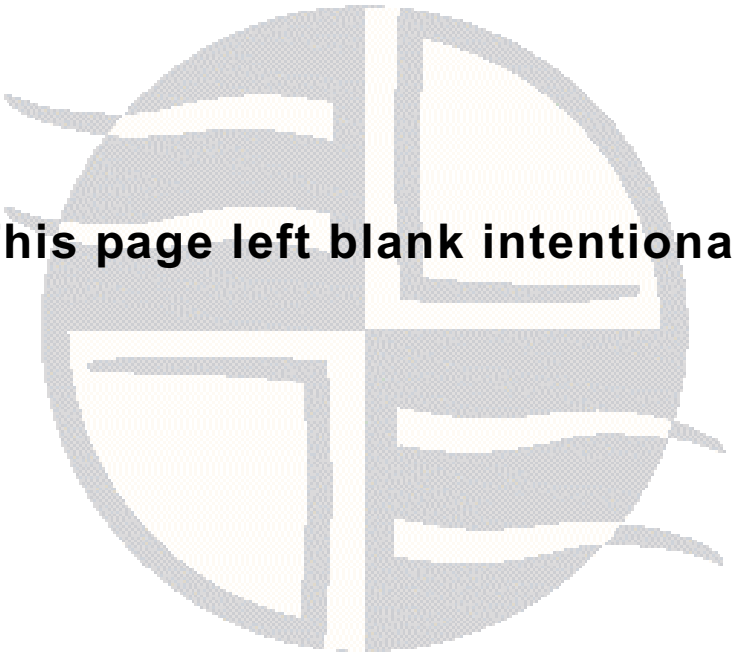
The Easysoft ODBC-DB2 Driver is currently available on Unix, Linux and Windows platforms. The most up to date list of Easysoft ODBC-DB2 Driver platforms is available at:

http://www.easysoft.com/products/data_access/odbc-db2-driver/index.html

Software problems can be reported to **support@easysoft.com** by users who have either purchased support or registered at the Easysoft web site at **http://www.easysoft.com** and are evaluating Easysoft products.

Deployment

The Easysoft ODBC-DB2 Driver uses the Distributed Relational Database Architecture (DRDA) data transfer protocol to communicate with DB2. No additional software needs to be installed on the client machine or the DB2 machine.



This page left blank intentionally

CHAPTER 2 INSTALLATION

Installing the Easysoft ODBC-DB2 Driver

This chapter explains how to install, license and remove the Easysoft ODBC-DB2 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 instructions assume you are, or are able to consult with, a system administrator.

Chapter Guide

Obtaining the Easysoft ODBC-DB2 Driver

There are three ways to obtain the Easysoft ODBC-DB2 Driver:

- The Easysoft web site is available 24 hours a day at **http://www.easysoft.com** and lets you download product releases and documentation.

  Choose **Download** from the Easysoft ODBC-DB2 Driver section of the web site and then choose the platform release that you require.

  If you have not already done so, you will need to register at the web site to download Easysoft software.

- The Easysoft FTP site is available 24 hours a day at **ftp://ftp.easysoft.com** and lets you download free patches, upgrades, documentation and beta releases of Easysoft products, as well as definitive releases.

  Change to the `pub/db2` subdirectory and then choose the platform release that you require.

- You can order Easysoft software on CD. To do this, **contact us** by email, telephone or post.

What to Install

The name of the Easysoft ODBC-DB2 Driver distribution file varies from platform to platform. The file name format is:

- `odbc-db2-x_y_z-windows.exe` (Windows)

– OR –

- `odbc-db2-`*`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 minor changes are made.

  `platform` depends on the operating system distribution you require. File names may have this format:

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

Caution! On Unix, as long as you stop all Easysoft software first (or software that uses the Easysoft drivers under Unix), it is safe to reinstall or upgrade the Easysoft ODBC-DB2 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).

To continue, refer to the installation instructions for your platform:

- **"Installing the Easysoft ODBC-DB2 Driver on Unix" on page 21**
- **"Installing the Easysoft ODBC-DB2 Driver on Windows" on page 41**

Installing the Easysoft ODBC-DB2 Driver on Unix

These instructions show how to install the Easysoft ODBC-DB2 Driver on Unix platforms. Please read this section carefully **before** installing the Easysoft ODBC-DB2 Driver.

BEFORE YOU INSTALL

Requirements

To install the Easysoft ODBC-DB2 Driver on Unix you need:

- The Bourne shell in `/bin/sh`. If your Bourne shell is not located there, you may need to edit the first line of the installation script.
- 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 do not have any of these commands, they can usually be obtained from the **Free Software Foundation**. As the `tee` command does not work correctly on some systems, the distribution includes a `tee` replacement.
- For Easysoft Licensing to work, you must do one of the following:
  - Install the Easysoft ODBC-DB2 Driver in `/usr/local/easysoft`.
  - Install the Easysoft ODBC-DB2 Driver 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`.

- Install the Easysoft ODBC-DB2 Driver elsewhere and set the `EASYSOFT_ROOT` environment variable.

  For more information about setting the `EASYSOFT_ROOT` environment variable, see **"Post installation" on page 37**.

- An ODBC Driver Manager. Easysoft ODBC-DB2 Driver distributions include the unixODBC Driver Manager.
- 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-DB2 Driver with unixODBC.
  2. Create the example data source in the SYSTEM `odbc.ini` file.
  3. Update the dynamic linker entries (some platforms only).

  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-DB2 Driver.
- The unixODBC Driver Manager.

You will need an ODBC Driver Manager to use the Easysoft ODBC-DB2 Driver from your applications. The distribution therefore contains the **unixODBC Driver Manager**. Most (if not all) Unix applications and interfaces support the unixODBC Driver Manager. For example, Perl DBD::ODBC, PHP, Python and so on.

You do not have to install the unixODBC Driver Manager included with this distribution. You can use an existing copy of unixODBC. For example, a version of unixODBC installed by another Easysoft product, a version obtained from your operating system vendor or one that you built yourself. However, as Easysoft ensure that the unixODBC distributed with the Easysoft ODBC-DB2 Driver has been tested with that driver, we recommend you use it.

If you choose to use an existing unixODBC Driver Manager, the installation script will attempt to locate it. The installation script looks for the Driver Manager in the standard places. If you have installed it in a non-standard location, the installation script will prompt you for the location. The installation primarily needs unixODBC's `odbcinst` command to install drivers and data sources.

Where to Install

This installation needs a location for the installed files. The default location 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 accept the default location `/usr/local`, the product will be installed in `/usr/local/easysoft` and below.

If you choose a different installation path, the installation script will try to symbolically link `/usr/local/easysoft` to the `easysoft` subdirectory in your chosen location. 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, the installation will be unable to create the symbolic link.

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 installation path or as the installation path itself.
- You have set `EASYSOFT_ROOT` to *`installation_path`*`/easysoft`.

Changes Made to Your System

This installation script installs files in subdirectories of the path requested at the start of the installation, Depending on what is installed, a few changes may be made to your system:

1. If you choose to install the Easysoft ODBC-DB2 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` is in *`installation_path`*`/easysoft/unixODBC/bin`, if you are using the unixODBC included with this distribution.)

   The `odbcinst.ini` entry for the Easysoft ODBC-DB2 Driver will look similar to this:

```
[Easysoft ODBC-DB2]
Description    = Easysoft ODBC-DB2 Driver
Driver         = /usr/local/easysoft/db2/lib/libesdb2.so
Setup          =
DontDLClose    = 1
FileUsage      = 1
UsageCount     = 1
```

For information about removing these entries, see **"Uninstalling the Easysoft ODBC-DB2 Driver on Unix" on page 40**.

2. The installation script installs example data sources into unixODBC. The data sources will be added to your SYSTEM `odbc.ini` file. You can locate your SYSTEM `odbc.ini` file by using `odbcinst -j`. The data source for the standard driver will look similar to this:

   For information about removing these data sources, see **"Uninstalling the Easysoft ODBC-DB2 Driver on Unix" on page 40**.

3. Dynamic Linker.

   On operating systems where the dynamic linker has a file listing locations for shared objects (Linux), the installation script will attempt to add paths under the path you provided at the start of the installation to the end of this list. On Linux, this is usually the file `/etc/ld.so.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 *`installation_path`*`/easysoft/lib`. When you run the installation script, the dates and versions of these files will be compared with the same files in the distribution. The files are 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. This is because on some platforms, files in use cannot be replaced. If a file cannot be updated, you will see a warning during the installation. All warnings are written to a 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 pieces of information. Before installing, you need to find out whether you have unixODBC already installed and where it is installed. The installation script searches standard places like `/usr` and `/usr/local`. However, if you installed the Driver Manager in a non-standard place and you do not install the included unixODBC, you will need to know the location.

INSTALLATION

Unpacking the Distribution

The distribution for Unix platforms is a tar file (`.tar`). To extract the installation files from the tar file, use:

```
tar -xvf odbc-db2-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 installation script and various other installation files.

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

License to Use

The End-User License Agreement is contained in the file `license.txt`. Be sure to understand the terms of the agreement 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 answer some questions. In each case, the default choice will be displayed in square brackets and you need only press Enter to accept the default. If there are alternative responses, these will be shown in round brackets; to choose one of these, type the response 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 answer when you type nothing and press Enter is `n`.

Running the Installer

Before you run the installer, make sure you have read **"Installation" on page 28**. 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 25**.)

To start the installation, run:

```
./install
```

You will need to:

- Confirm your acceptance of the license agreement by typing "yes" or "no".

For more information about the license agreement, see **"License to Use" on page 28**.

- Supply the location where the software is to be installed. Easysoft recommend accepting the default installation path.

  For more information, see **"Where to Install" on page 24**.

> **Note** 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 script is designed to work with unixODBC and can automatically add Easysoft ODBC-DB2 Driver and data sources during the installation.
- Most ODBC-enabled applications and interfaces support unixODBC. The Easysoft ODBC-DB2 Driver and any data sources that you add during the installation will be automatically available to your applications and interfaces therefore.
- The unixODBC project is currently led by Easysoft developer Nick Gorham. This means that there is a great deal of experience at Easysoft of unixODBC in general and of supporting the Easysoft ODBC-DB2 Driver 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 installation starts by searching for unixODBC. There are two possible outcomes here:

1. If the installation script finds unixODBC, the following message will be output:

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

2. If the installation script cannot find unixODBC in the standard places, you will be asked whether you have it installed.

   If unixODBC is installed, you need to provide the unixODBC installation path. Usually, the path required is the directory above where `odbcinst` is installed. For example, if `odbcinst` is in `/opt/unixODBC/bin/odbcinst`, the required path is `/opt/unixODBC`.

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

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

   The unixODBC in the Easysoft ODBC-DB2 Driver distribution is not built with the default options in unixODBC's configure line.

Option	Description
--prefix=/etc	This means the default SYSTEM `odbc.ini` file where SYSTEM data sources are located will be `/etc/odbc.ini`.
--enable-drivers=no	This means other ODBC drivers that come with unixODBC are not installed.

Option	Description
--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 use system semaphores to keep track of used handles. Many systems do not have sufficient semaphore resources to keep track of used handles. In addition, the statistics are only available in the GUI ODBC Administrator.
--enable-readline=no	This disables readline support in `isql`. We disabled this because it ties `isql` to the version of libreadline on the system we build on. We build on as old a version of the operating system as we can for forward compatibility. Many newer Linux systems no longer include the older readline libraries and so enabling readline support makes `isql` unusable on these systems.
--prefix=/usr/local/easysoft/unixODBC	This installs unixODBC into `/usr/local/easysoft/unixODBC`.

Figure 1: Easysoft unixODBC configure line options.

Installing the Easysoft ODBC-DB2 Driver

The Easysoft ODBC-DB2 Driver installation script:

- Installs the driver.
- Registers the driver with the unixODBC Driver Manager.

  If the Easysoft ODBC-DB2 Driver is already registered with unixODBC, a warning will be displayed that lists the drivers unixODBC knows about. If you are installing the Easysoft ODBC-DB2 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. unixODBC's `odbcinst` will not update these paths if a driver is already registered.
- Creates an example Easysoft ODBC-DB2 Driver data source.

  If unixODBC is installed and you registered the Easysoft ODBC-DB2 Driver with unixODBC, an example data source will be added to your `odbc.ini` file.

  If a data source called "DB2_SAMPLE", the existing data source will be displayed and you have the option to replace it.

Licensing

The *`installation_path`*`/easysoft/license/licshell` program lets you obtain or list licenses.

Licenses are stored in the *`installation_path`*`/easysoft/license/licenses` file. After obtaining a license, you should make a backup copy of this file.

The installation script asks you if you want to request an Easysoft ODBC-DB2 Driver license:

```
Would you like to request a Easysoft ODBC-DB2
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, the installation runs the `licshell` script. The process of obtaining a license is best described in the **Licensing Guide**.

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, the License Client can create a license request file that you can mail or fax to Easysoft. You can also supply the details to us by telephone.

Start the License Client. The following menu is displayed:

```
[0] exit

[1] view existing license

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

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 generates a key that is used to identify the product and operating system (we need this key to license you).

After you have chosen the product to license (Easysoft ODBC-DB2 Driver), you need to supply:

- Your full name.
- Your company name.
- An email contact address. This **must** be the email address that you used when you registered on the Easysoft web site.
- Your telephone number (you need to specify this if you telephone us to request a license).
- Your fax number (you need to specify this if you fax the license request to us).
- A reference number. When applying for a trial license, just press Enter when prompted for a reference number. This field is used to enter a reference number that we will supply you for full (paid) licenses.

You will then be asked to specify how you want to obtain the license. The choices are:

```
[1] Automatically by contacting the Easysoft
License Daemon
```

This requires a connection to the Internet and the ability to support an outgoing TCP/IP connection to `license.easysoft.com` on port 8884.

```
[2] Write information to file so you can fax,
telephone it
```

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

```
[3] Cancel this operation
```

If you choose to obtain the license automatically, the License Client will start a TCP/IP connection to `license.easysoft.com` on port 8884 and send the details you supplied and your machine number. No other data is sent. The data sent is transmitted as plain text, so if you want to avoid the possibility of this information being intercepted by someone else on the Internet, 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`. You should then exit the License Client by choosing option `[0]` and complete the installation. After you have mailed, faxed or telephoned the license request to us, we will return a license key. Add this to the end of the file *`installation_path`*`/easysoft/license/licenses`.

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

POST INSTALLATION

Supplied Documents and Examples

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

- The Easysoft ODBC-DB2 Driver documentation is installed in *installation_path*/easysoft/db2/doc:
  - The Easysoft ODBC-DB2 Driver manual in PDF format.
  - The Easysoft ODBC-DB2 Driver EULA.

installation_path/easysoft/db2/doc/CHANGES.txt lists all the changes in each version of the Easysoft ODBC-DB2 Driver.

There are also many resources at the **Easysoft web site**.

SETTING DYNAMIC LINKER SEARCH PATHS

Your applications will be linked against an ODBC Driver Manager, which will load the ODBC Driver you require. The dynamic linker needs to know where to find the ODBC Driver Manager shared object. The ODBC Driver Manager will load the Easysoft ODBC-DB2 Driver, which is dependent on further common Easysoft shared objects; the dynamic linker needs to locate these too.

On operating systems where the dynamic linker has a file specifying locations for shared objects (Linux, for example), the installation will attempt to add paths under the path you provided at the start of the installation to the end of this list; no further action should be required. For more information, see **"Dynamic Linker." on page 27**.

On other Unix platforms, there are two methods of telling the dynamic linker where to look for shared objects:

1. You add the search paths to an environment variable and export it.

   This method always works and overrides the second method, described below.

2. At build time, a run path is inserted into the executable or shared objects. On most System V systems, Easysoft distribute Easysoft ODBC-DB2 Driver shared objects with an embedded run path. The dynamic linker uses the run path to locate Easysoft ODBC-DB2 Driver shared object dependencies.

   For the first method, the environment variable you need to set depends on the platform (refer to the platform documentation for `ld(1)`, `dlopen` or `ld.so(8)`).

Environment Variable	Platform
LD_LIBRARY_PATH	System V based operating systems and Solaris.
LIBPATH	AIX
SHLIB_PATH	HP-UX
LD_RUN_PATH	Many platforms use this in addition to those listed above.

Figure 2: Dynamic linker search path environment variables.

To use the Easysoft ODBC-DB2 Driver, you need to add:

installationdir`/easysoft/lib`

where *installationdir* is the directory in which you chose to install the Easysoft ODBC-DB2 Driver. If you accepted the default location, this is `/usr/local`.

An example of setting the environment path in the Bourne shell on Solaris is:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/easysoft/lib

export LD_LIBRARY_PATH
```

> **Note** The exact command you need to set and export an environment variable depends on your shell.

If you installed the unixODBC Driver Manager included in the Easysoft ODBC-DB2 Driver distribution, you also need to add *installationdir*`/easysoft/unixODBC/lib` to the dynamic linker search path.

Uninstalling the Easysoft ODBC-DB2 Driver on Unix

There is no automated way to remove the Easysoft ODBC-DB2 Driver in this release. However, removal is quite simple. To do this, follow these instructions.

To uninstall the Easysoft ODBC-DB2 Driver

1. Change directory to *installation_path*`/easysoft` and delete the `db2` directory. *installation_path* is the Easysoft ODBC-DB2 Driver installation directory, by default `/usr/local`.
2. If you had to add this path to the dynamic linker search paths (for example, `/etc/ld.so.conf` on Linux), remove it. You may have to run a linker command such as `/sbin/ldconfig` to get the dynamic linker to reread its configuration file. Usually, this step can only be done by the `root` user.
3. If you were using unixODBC, the Easysoft ODBC-DB2 Driver entry needs to be removed from the `odbcinst.ini` file. To check whether the Easysoft ODBC-DB2 Driver is configured under unixODBC, use `odbcinst -q -d`. If the command output contains `[Easysoft ODBC-DB2 Driver]`, uninstall the drivers from unixODBC by using:

   ```
   odbcinst -u -d -n 'Easysoft ODBC-DB2'
   ```

Installing the Easysoft ODBC-DB2 Driver on Windows

INSTALLING THE EASYSOFT ODBC-DB2 DRIVER

- Execute the file distribution that you downloaded in **"Obtaining the Easysoft ODBC-DB2 Driver" on page 18**

  Follow the on screen instructions.

UPDATING FILES THAT ARE IN USE

To avoid rebooting your computer, the Easysoft ODBC-DB2 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 and later, the Easysoft ODBC-DB2 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-DB2 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**), because you cannot use the Easysoft ODBC-DB2 Driver until a license is obtained.

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.

4. 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.

5. Click **Request License**.

   You are asked for a license type.

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

   The License Manager asks what software you are licensing:

   Select your required version of the Easysoft ODBC-DB2 Driver (Standard or Remote, for example) 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**.

7. The License Manager displays a summary of the information you entered and allows you to choose the method of applying for your license.

8. 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. You can now go to **step 9 on page 45**.

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 ways to obtain a license if your machine is off-line (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 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.

9. 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 the licensing procedure refer to the **Licensing Guide**.

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

The installation is complete.

REPAIRING THE EASYSOFT ODBC-DB2 DRIVER INSTALLATION

The installer can repair a broken Easysoft ODBC-DB2 Driver installation. For example, you can use the installer to restore missing Easysoft ODBC-DB2 Driver files or registry keys.

In Windows Vista and later versions of Windows:

1. In **Control Panel**, open **Programs and Features**.
2. Right-click **Easysoft ODBC-DB2 Driver**, and then click **Repair**.

In earlier versions of Windows:

1. In **Control Panel**, open **Add or Remove Programs**.
2. Select **Easysoft ODBC-DB2 Driver** and click **Change/Remove**.

Uninstalling the Easysoft ODBC-DB2 Driver on Windows

This section explains how to remove the Easysoft ODBC-DB2 Driver from your system.

Removing Easysoft ODBC-DB2 Driver Data Sources

Easysoft ODBC-DB2 Driver data sources are not removed when you uninstall. You therefore do not need to recreate your Easysoft ODBC-DB2 Driver data sources if you reinstall or upgrade. If you do not want to keep your Easysoft ODBC-DB2 Driver data sources, use ODBC Administrator to remove them, **before** uninstalling the Easysoft ODBC-DB2 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-DB2 DRIVER

In Windows Vista and later versions of Windows:

1. In **Control Panel**, open **Programs and Features**.
2. Double-click **Easysoft ODBC-DB2 Driver**.

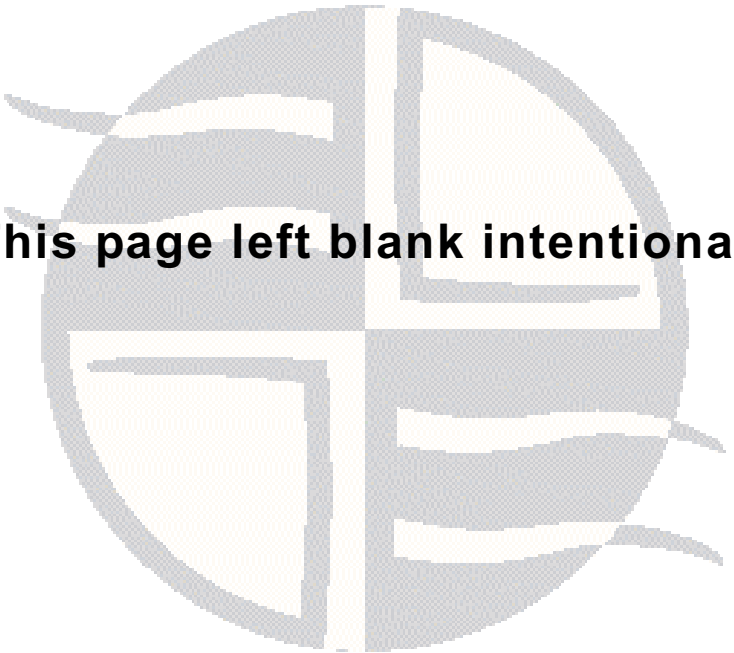
In earlier versions of Windows:

1. In **Control Panel**, open **Add or Remove Programs**.
2. Select **Easysoft ODBC-DB2 Driver** and click **Change/Remove**.

The uninstall process is complete.

Any licenses you obtained for the Easysoft ODBC-DB2 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.



This page left blank intentionally

CHAPTER 3 CONFIGURATION

Configuring the Easysoft ODBC-DB2 Driver

The Easysoft ODBC-DB2 Driver is installed on the computer where your applications are running. ODBC applications access ODBC drivers through the ODBC Driver Manager and a data source. The data source tells the Driver Manager which ODBC driver to load, which DB2 server to connect to and how to connect to it. This chapter describes how to create data sources, use DSN-less connections and configure the Easysoft ODBC-DB2 Driver.

Before setting up a data source, you must have successfully installed the Easysoft ODBC-DB2 Driver.

For Easysoft ODBC-DB2 Driver installation instructions, see **"Installation" on page 17**.

Chapter Guide

Configuring the Easysoft ODBC-DB2 Driver

This section describes how to configure the Easysoft ODBC-DB2 Driver to connect to a DB2 database by using a data source or a DSN-less connection string. The section assumes you are, or are able to consult with, a database administrator.

Refer to the section relevant to your platform to continue:

- **"Setting Up Data Sources on Unix" on page 51**
- **"Setting Up Data Sources on Windows" on page 55**

Setting Up Data Sources on Unix

There are two ways to set up a data source to your DB2 data:

- Create a SYSTEM data source, which is available to anyone who logs on to this Unix machine.

– OR –

- Create a USER data source, which is only available to the user who is currently logged on to this Unix machine.

By default, the Easysoft ODBC-DB2 Driver installation creates a SYSTEM data source named `[DB2_SAMPLE]`. If you are using the unixODBC included in the Easysoft ODBC-DB2 Driver distribution, the SYSTEM `odbc.ini` file is in `/etc`.

If you built unixODBC yourself, or installed it from some other source, 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`.

If you accepted the default choices when installing the Easysoft ODBC-DB2 Driver, USER data sources must be created and edited in `$HOME/.odbc.ini`.

Note

To display the directory where unixODBC stores SYSTEM and USER data sources, type `odbcinst -j`.

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

You can either edit the sample data source or create new 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.

> **Note** Attribute names in `odbc.ini` are not case sensitive.

The `Driver` attribute identifies the ODBC driver in the `odbcinst.ini` file to use for a data source.

When the Easysoft ODBC-DB2 Driver is installed into unixODBC, it places an `Easysoft ODBC-DB2` entry in `odbcinst.ini`. For Easysoft ODBC-DB2 Driver data sources therefore, you need to include a `Driver = Easysoft ODBC-DB2` entry.

To configure a DB2 data source, in your `odbc.ini` file, you need to specify:

- The host name or IP address of the machine where the DB2 instance is running (`Server`). To connect to a DB2 instance that is **not** listening on the default port (50000), you also need to specify the port number (`Port`).
- A valid operating system user name (`User`) and password (`Password`).

For example:

```
[DB2]
Driver    = Easysoft ODBC-DB2
Server    = my_db2_hostname
Database  = sample
User      = my_os_user
Password  = my_os_password
```

ENVIRONMENT

The Easysoft ODBC-DB2 Driver must be able to find the following shared objects, which are installed during the Easysoft ODBC-DB2 Driver installation:

- `libodbcinst.so`

  By default, this is located in `/usr/local/easysoft/unixODBC/lib`.

- `libeslicshr_r.so`

  By default, this is located in `/usr/local/easysoft/lib`.

- `libessupp_r.so`

  By default, this is located in `/usr/local/easysoft/lib`.

You may need to set and export `LD_LIBRARY_PATH`, `SHLIB_PATH` or `LIBPATH` (depending on your operating system and run-time linker) to include the directories where `libodbcinst.so`, `libeslicshr.so` and `libessupp.so` are located.

> **Note** The shared object file extension (`.so`) may vary depending on the operating system (`.so`, `.a` or `.sl`).

ESTABLISHING A TEST CONNECTION

The `isql` query tool lets you test your Easysoft ODBC-DB2 Driver data sources.

To test the Easysoft ODBC-DB2 Driver connection

1. Change directory into `/usr/local/easysoft/unixODBC/bin`.
2. Type `./isql.sh -v` *`data_source`*, where *data_source* is the name of the target data source.
3. At the prompt, type an SQL query. For example:

```
SQL> select * from mytable;
```

– OR –

Type `help` to return a list of tables:

```
SQL> help
```

Setting Up Data Sources on Windows

To connect an ODBC application on a Windows machine to a DB2 database:

1. Open ODBC Data Source Administrator:

   - For Microsoft Windows 8, in the Windows desktop, point to the upper-right corner of the screen, move the mouse pointer down, and then choose **Settings** > **Control Panel** > **Administrative Tools** > **ODBC Data Sources (64-bit)**.
   - For Microsoft Windows Vista and Windows 7, choose **Start** > **Control Panel** > **Administrative Tools** > **Data Sources (ODBC)**.
   - For Microsoft Windows Server 2008 and Windows Server 2008 R2, choose **Start** > **Administrative Tools** > **Data Sources (ODBC)**.
   - For Microsoft Windows 2000, Windows XP and Windows Server 2003, choose **Start** > **Settings** > **Control Panel** > **Administrative Tools** > **Data Sources**.

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

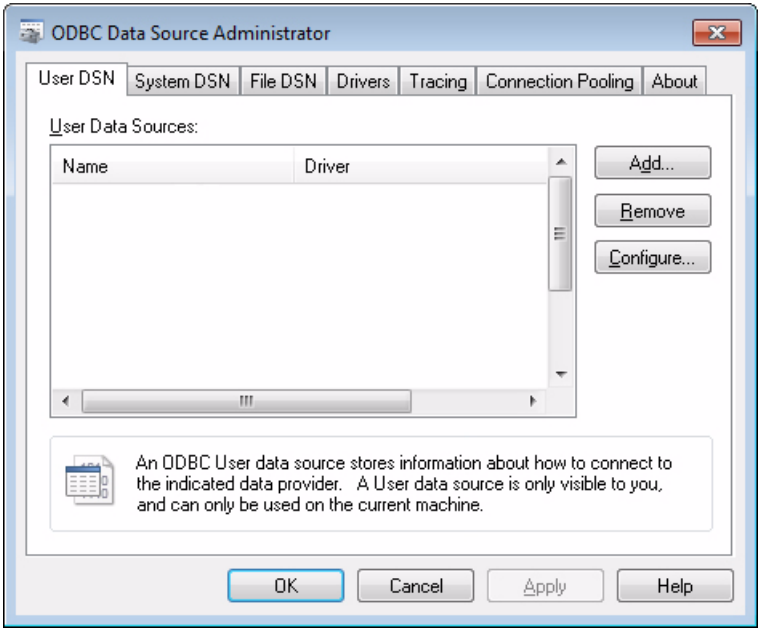


Figure 3: The ODBC Data Source Administrator dialog box

2. Select the **User DSN** tab to set up a data source that only you can access.

   – OR –

   Select the **System DSN** tab to create a data source which is available to anyone who logs on to this Windows machine.

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

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

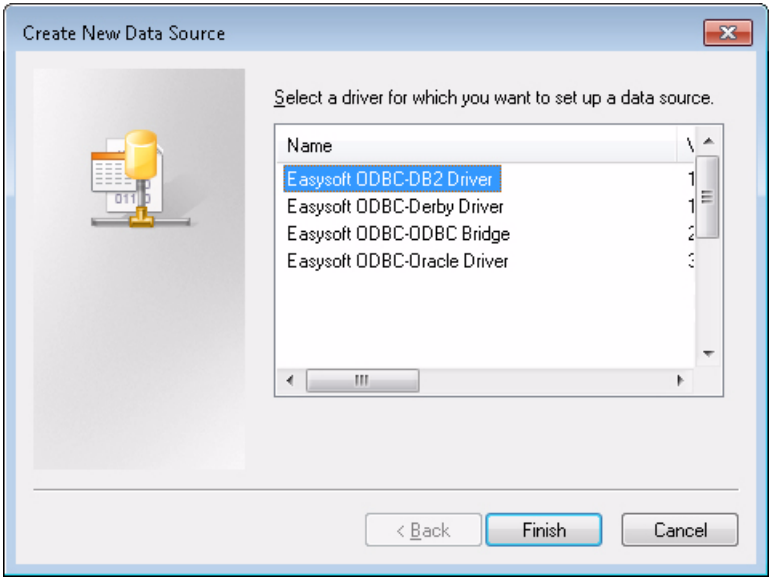


Figure 4: The Create New Data Source dialog box

4. Select `Easysoft ODBC-DB2 Driver` and click **Finish**.

The Easysoft ODBC-DB2 Driver DSN Setup dialog box is displayed:

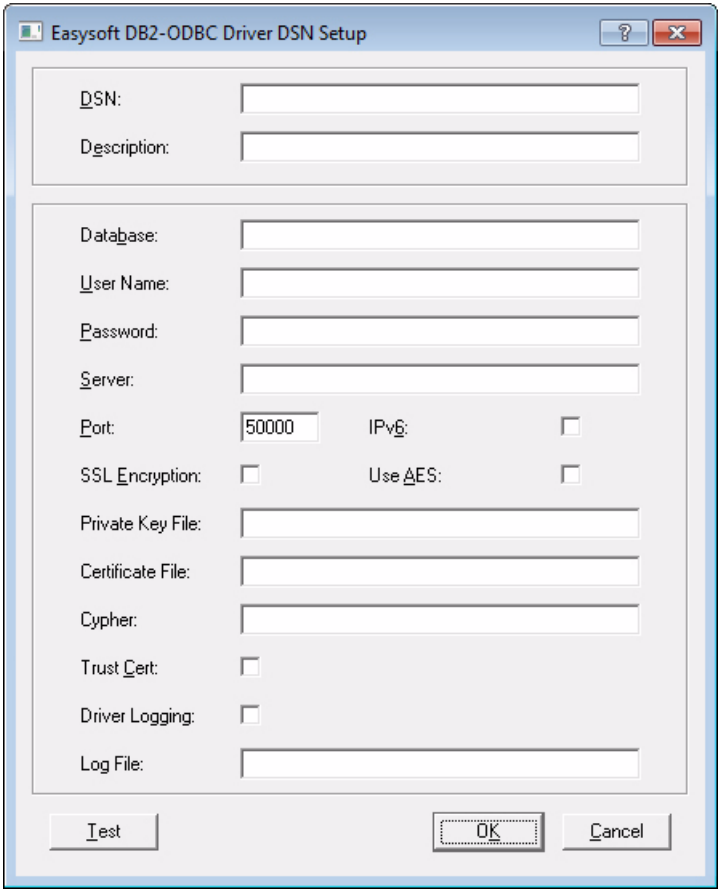


Figure 5: The Easysoft ODBC-DB2 Driver DSN Setup dialog box

For details of the other attributes that can be set on this dialog box, see **"Attribute Fields" on page 60**.

64-bit Windows

The Easysoft installer program installs both a 32-bit and a 64-bit version of the Easysoft ODBC-DB2 Driver. If you want to use a 64-bit ODBC application, you need to use the 64-bit Easysoft ODBC-DB2 Driver. If you want to use a 32-bit ODBC application, you need to use the 32-bit Easysoft ODBC-DB2 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 Windows 7 and earlier, in the Windows Run dialog box, type:

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

On Windows 8, both the 32-bit and 64-bit ODBC Administrator are located in Control Panel under Administrative tools: ODBC Data Sources (32-bit) and ODBC Data Sources (64-bit).

Easysoft ODBC-DB2 Driver data sources created in the 64-bit ODBC Administrator will specify the 64-bit version of the Easysoft ODBC-DB2 Driver. Easysoft ODBC-DB2 Driver data sources created in the 32-bit ODBC Administrator will specify the 32-bit version of the Easysoft ODBC-DB2 Driver.

If you want to create an Easysoft ODBC-DB2 Driver System data source for use with a 64-bit application, use the 64-bit ODBC Administrator. If you want to create an Easysoft ODBC-DB2 Driver System data source for use with a 32-bit application, use the 32-bit ODBC Administrator.

For Easysoft ODBC-DB2 Driver User data sources, it does not matter which version of the ODBC Administrator that you use.

Attribute Fields

This section lists the attributes which can be set for the Easysoft ODBC-DB2 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:

```
DIALECT=1|2|3
```

the value entered may be "1", "2" or "3".

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 that 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

The database to to connect to. For example, to connect to the DB2 SAMPLE database set the `Database` attribute value to `SAMPLE`.

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

USER NAME

The operating system user name to use when connecting to DB2. You must specify a name of a user who has access to the machine on which DB2 is running.

To specify the user name in the connection string, use `UID` rather than `User`. For more information about specifying Easysoft ODBC-DB2 Driver attributes in the connection string, see **"DSN-less Connections" on page 77**.

Interface	Value
DSN Dialog Box (Windows)	User Name
odbc.ini file (Unix)	User = *value*
Connect String	USER=*value*

PASSWORD

The password for the user name.

To specify the password in the connection string, use `PWD` rather than `Password`.

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

SERVER

The host name or IP address of the machine on which the DB2 instance is running.

Connection Failover

If your DB2 database is available on more than one DB2 machine, you can define a primary server for the database and additional fallback database servers. By default, the Easysoft ODBC-DB2 Driver will try to connect to the first server that you specify. If that server is unavailable, the Easysoft ODBC-DB2 Driver will try to connect to the next server in the list and so on. Use the format:

```
Server = primaryserver[:port] [, fallbackserver[:port]...]
```

where:

- *primaryserver* is the name or IP address of the primary DB2 machine on which your database is available.
- *port* is the TCP port on which the instance is listening. If omitted, the driver will try to connect to the instance that is listening on port 50000.
- *fallbackserver* is the name or IP address of an alternative DB2 machine on which your database is available.

For example:

```
Server = db2hostA,db2hostB,db2hostC:50001
```

Connection attempts continue until either a connection is successfully made or all the servers in the list have been tried once.

Note that authentication details (as specified by `User` and `Password`) needs to be valid on each DB2 machine in the list.

On Unix, if you want to balance the load between database servers, configure the driver to randomly choose the database server it connects to. To do this, set the `ClientLB` attribute to 1.

Interface	Value
DSN Dialog Box (Windows)	Server
odbc.ini file (Unix)	Server = *value*
Connect String	SERVER = *value*

PORT

The TCP port that the DB2 instance is listening on.

If you are connecting to a default instance that is listening on port 50000, the `Port` setting can be omitted.

Interface	Value
DSN Dialog Box (Windows)	Port
odbc.ini file (Unix)	Port = *num*
Connect String	PORT=*num*

IPv6

Set `IPv6` to 1 when connecting to a DB2 instance that is listening on an IPv6 address.

By default, IPv6 is OFF (set to 0), which means that the Easysoft ODBC-DB2 Driver assumes that the target DB2 instance is listening on an IPv4 address.

Interface	Value
DSN Dialog Box (Windows)	IPv6
odbc.ini file (Unix)	IPv6 = 0|1
Connect String	IPV6=0 | 1

SSL ENCRYPTION

Whether the Easysoft ODBC-DB2 Driver requests an encrypted connection to DB2.

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

USE AES

The encryption algorithm to use when encrypting the user name and password. When ON (set to 1), the Easysoft ODBC-DB2 Driver encrypts the user name and password by using an Advanced Encryption Standard (AES) encryption algorithm. Otherwise, the Easysoft ODBC-DB2 Driver encrypts the user name and password by using a Data Encryption Standard (DES) encryption algorithm.

By default, `Use AES` is OFF (set to 0).

Interface	Value
DSN Dialog Box (Windows)	Use AES
odbc.ini file (Unix)	AESEncAlg = 0 | 1
Connect String	AESENCALG = 0 | 1

PRIVATE KEY FILE

The private key for use with the SSL session.

Interface	Value
DSN Dialog Box (Windows)	Private Key File
odbc.ini file (Unix)	PrivateKeyFile = *value*
Connect String	PRIVATEKEYFILE = *value*

CERTIFICATE FILE

The file that contains the public key certificate of the CA that signed the DB2 certificate. The CA certificate file must be in base-64 PEM format.

If the CA certificate is not installed on your client machine, you need to export the certificate on the DB2 machine and install it on the client.

Interface	Value
DSN Dialog Box (Windows)	Certificate File
odbc.ini file (Unix)	CertificateFile = *value*
Connect String	CERTIFICATEFILE = *value*

CYPHER

The cypher suite that the Easysoft ODBC-DB2 Driver will request during the SSL handshake with the DB2 machine.

A cypher suite is a set of authentication, encryption, and data integrity algorithms used to protect data exchanged between machines. During the SSL handshake part of the connection process, the SSL layer in the ODBC driver and the Schannel layer on the DB2 machine negotiate to decide which cipher suite they will use.

To see which cypher suite is being used for a particular connection, enable Easysoft ODBC-DB2 Driver logging.

Connect and then examine the driver log file. Look for a log file entry similar to:

```
SSL using cypher 'RC4-MD5 SSLv3 Kx=RSA Au=RSA
Enc=RC4(128) Mac=MD5'
```

This entry shows that the ODBC driver and the DB2 machine negotiated the following cryptographic protection for the connection:

Encryption: RC4

Encryption strength: 128-bit

Cryptographic checksum: MD5

Authentication: RSA

(You can also display the cryptographic settings negotiated during the SSL handshake by enabling Schannel logging. Enable the "`Log informational and success events`" Schannel logging option to write this information to the Windows Event Viewer logs. For information about how to do this, see **http://support.microsoft.com/kb/260729**.)

Use the `Cypher` setting, if you want to request a different encryption or data integrity algorithm to the ones negotiated during the SSL handshake. For example:

```
Cypher = 3DES+SHA
```

If you specify a cypher suite that is not available on the server machine, the Easysoft ODBC-DB2 Driver returns the error "`Required SSL (failed to receive packet)`".

If you specify a cypher suite that the Easysoft ODBC-DB2 Driver does not recognise, the driver returns the error `"SSL3_CLIENT_HELLO:no ciphers available"`.

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

TRUST CERT

Whether the Easysoft ODBC-DB2 Driver tries to validate the server certificate to verify the identity of the DB2 machine. Set `Trust Cert` to `Yes` if your DB2 machine is using a self-signed SSL certificate.

Interface	Value
DSN Dialog Box (Windows)	Trust Cert
odbc.ini file (Unix)	TrustServerCertificate = 0 | 1
Connect String	TRUSTSERVERCERTIFICATE = 0 | 1

CLIENTLB

Whether the Easysoft ODBC-DB2 Driver tries to balance the load between the servers specified by the `Server` setting. The `ClientLB` setting only has an effect if you specify a primary server and additional fallback servers with `Server`.

When `ClientLB` is ON (set to 1), the Easysoft ODBC-DB2 Driver randomly selects a server to connect to. If the server is unavailable, the Easysoft ODBC-DB2 Driver then moves sequentially through the list of other servers.

When `ClientLB` OFF (set to 0, the default), the Easysoft ODBC-DB2 Driver tries to connect to the servers in the order that they are defined in. (Primary server first and then each additional fallback server.)

Example

You specify a primary server (`db2hostA`) and two fallback servers (`db2hostB` and `db2hostC`):

```
Server = db2hostA,db2hostB,db2hostC:1583
```

When `ClientLB` is ON, the Easysoft ODBC-DB2 Driver will randomly choose a server to connect to. If, for example, the driver tries to connect to `db2hostB` first, it will then try to connect to `db2hostC` (if `db2hostB` is unavailable) and `db2hostA` (if `sqlsrvhostC` is unavailable).

When `ClientLB` is OFF, the Easysoft driver will try to connect to `db2hostA` and then `db2hostB` (if `db2hostA` is unavailable) and finally `db2hostC` (if `db2hostB` is unavailable).

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

RCVBUFFER

The size of the receive buffer for the socket in bytes. Possible values for *num* are:

0, do not set the receive buffer size, use the system default value.

n, where *n* is a number greater than 0, set the receive buffer to the specified size by passing *n to* the `setsockopt()` function .

By default, the system default receive buffer size is used.

Interface	Value
DSN Dialog Box (Windows)	Not available.
odbc.ini file (Unix)	RcvBuffer = *num*
Connect String	RCVBUFFER = *num*

DPREC

The precision to use when converting `SQL_DOUBLE` data in a result set to a string

If an application specifies a string as the target type for non-character data in a `SQLBindCol` or `SQLGetData` call, the Easysoft ODBC-DB2 Driver converts the data to the target type. Use the `DPrec` attribute to specify the precision to use when the driver does this conversion for `SQL_DOUBLE` data.

The default precision is 7.

Interface	Value
DSN Dialog Box (Windows)	Not available.
odbc.ini file (Unix)	DPrec = *num*
Connect String	DPREC = *num*

FPREC

The precision to use when converting `SQL_FLOAT` data in a result set to a string

If your application specifies a string as the target type for non-character data in a `SQLBindCol` or `SQLGetData` call, the Easysoft ODBC-DB2 Driver converts the data to the target type. Use the `FPrec` attribute to specify the precision to use when the driver does this conversion for `SQL_FLOAT` data.

The default precision is 7.

Interface	Value
DSN Dialog Box (Windows)	Not available.
odbc.ini file (Unix)	FPrec = *num*
Connect String	FPREC = *num*

LIMITVARCHAR

Use `LimitVarchar` to restrict the size returned by the Easysoft ODBC-DB2 Driver when describing `VARCHAR` data.

Interface	Value
DSN Dialog Box (Windows)	Not available.
odbc.ini file (Unix)	LimitVarchar = *num*
Connect String	LIMITVARCHAR = *num*

LOCALE

The locale on the Easysoft ODBC-DB2 Driver machine.

If you do not set the `Locale` attribute, the Easysoft ODBC-DB2 Driver will use the value set for the `LC_CTYPE` environment variable. If `LC_CTYPE`, is not set, the Easysoft ODBC-DB2 Driver will use the value set for the `LANG` environment variable. If neither `LC_CTYPE` nor `LANG` are set, the Easysoft ODBC-DB2 Driver will set the locale value to C.

Interface	Value
DSN Dialog Box (Windows)	Not available.
odbc.ini file (Unix)	Locale = *value*
Connect String	LOCALE = *value*

CONVWTOUTF

When ON (set to 1), the Easysoft ODBC-DB2 Driver converts strings passed to Unicode ODBC calls (with suffix "W") to UTF-8. The Easysoft ODBC-DB2 Driver also converts metadata and result sets returned by Unicode ODBC calls to UTF-8.

By default, `ConvWToUtf` is OFF (set to 0).

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

SbUTF8

Controls how single bytes in a string are converted to Unicode. When ON (set to 1), UTF-8 sequences are regarded as single Unicode values. Otherwise, UTF-8 sequences are regarded as individual 8-bit values.

For example, setting `SbUTF8` controls whether a UTF-8 Euro symbol (0xE2 0x82 0xAC) converts to 0x20AC (single character) or 0x00E2, 0x0082, 0x00AC (three characters).

By default, `SbUTF8` is OFF (set to 0).

Interface	Value
DSN Dialog Box (Windows)	Not available
odbc.ini file (Unix)	SbUTF8 = 0 | 1
Connect String	SBUTF8 = 0 | 1

GSSLIB

The Easysoft ODBC-DB2 Driver uses `libgssapi_krb5.so`, the Kerberos GSS-API library, to request service tickets for accessing DB2 instances. If the Easysoft ODBC-DB2 Driver is unable to open this library, the connection will fail with the error:

```
Krb5: failed to open gss lib (libgssapi_krb5.so)
```

If the Kerberos GSS-API library is not called `libgssapi_krb5.so` in your GSS-API distribution, use the `GSSLIB` attribute in your data source to specify the alternative GSS-API library. For example:

```
GSSLIB = /opt/extension/lib/libgssapi.so
```

Interface	Value
DSN Dialog Box (Windows)	Not available.
odbc.ini file (Unix)	GSSLib = *value*
Connect String	GSSLIB = *value*

GSSFLAG

The Easysoft ODBC-DB2 Driver allows you to pass `req_flags` to the `gss_init_sec_context()` function, which is used to initiate a security context for the driver. The Key Distribution Center (KDC) uses this security context to verify the identity of the client. To pass `req_flags` to `gss_init_sec_context()`, use the `GSSFLAG` attribute:

`GSSFLAG = req_flags`

where `req_flags` is a bitmask specifying the requested GSS services. To look up the available bitmask values, refer to the `gssapi.h` header file for the GSS-API distribution on the Easysoft ODBC-DB2 Driver machine. The driver default `GSSFLAG` value is 4, which sets the `GSS_C_REPLAY_FLAG` flag.

As an example, to request credential delegation, set the `GSS_C_DELEG_FLAG` flag by including this entry in your data source `GSSFLAG = 1`.

Interface	Value
DSN Dialog Box (Windows)	Not available.
odbc.ini file (Unix)	GSSFlag = *req_flags*
Connect String	GSSFLAG = *req_flags*

DSN-less Connections

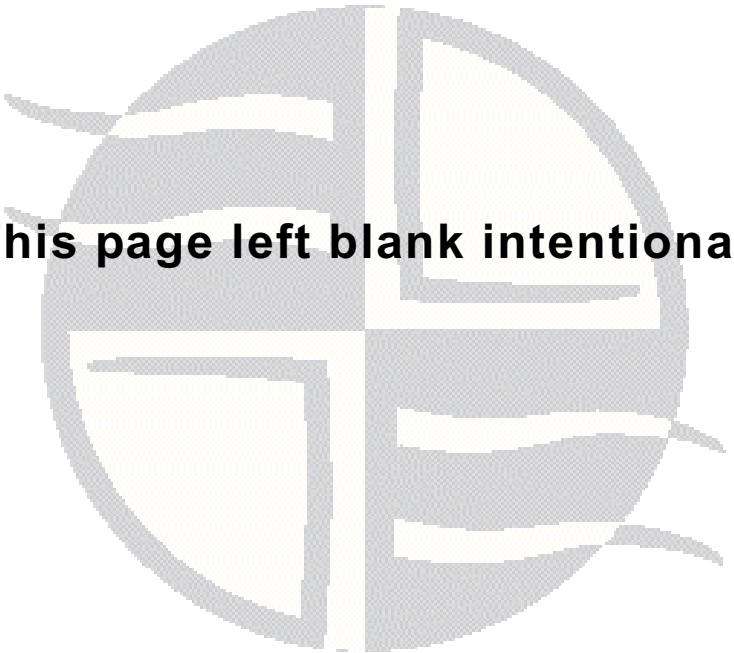
In addition to using a data source, you can also connect to a database by using a DSN-less connection string of the form:

```
SQLDriverConnect(..."DRIVER={Easysoft ODBC-DB2};
                    Server=server;UID=user;PWD=password;
                    Database=database;"...)
```

where *server* is the host name or IP address of the machine on which the DB2 instance is running, *user* and *password* are a valid operating system user name and password and *database is the DB2* database you want to connect to. You need to use the `Easysoft ODBC-DB2` DRIVER keyword to identify the Easysoft ODBC-DB2 Driver.

Other Easysoft ODBC-DB2 Driver attribute settings, as described in **"Setting Up Data Sources on Unix" on page 51**, can be added to the connection string using the same `PARAMETER=value;` format. For example, the following connection string connects to a DB2 instance that is listening on a non-standard port:

```
"DRIVER={Easysoft ODBC-
DB2};Server=myhost;UID=myuser;PWD=mypassword;Database=SAMPLE;Port=50001"
```



This page left blank intentionally

APPENDIX A TECHNICAL REFERENCE

Technical Reference for the Easysoft ODBC-DB2 Driver

This section contains extra information relating to the deployment of the Easysoft ODBC-DB2 Driver.

Appendix Guide

ODBC Conformance

The Easysoft ODBC-DB2 Driver complies with the ODBC 3.52 specification.

The Easysoft ODBC-DB2 Driver is Level 2 compliant.

ODBC API Support

All ODBC 3.52 calls are supported.

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-DB2 Driver.

Within the Driver Manager by an application

An application can turn tracing on in the Driver Manager by using the ODBC API `SQLSetConnectAttr` (...,`SQL_ATTR_TRACE`,...).

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

From within the Driver Manager on Windows

- For Microsoft Windows 8, in the Windows desktop, point to the upper-right corner of the screen, move the mouse pointer down, and then choose **Settings** > **Control Panel** > **Administrative Tools** > **ODBC Data Sources (64-bit)**.
- For Microsoft Windows Vista and Windows 7, choose **Start** > **Control Panel** > **Administrative Tools** > **Data Sources (ODBC)**.
- For Microsoft Windows Server 2008 and Windows Server 2008 R2, choose **Start** > **Administrative Tools** > **Data Sources (ODBC)**.
- For Microsoft Windows 2000, Windows XP and Windows Server 2003, choose **Start** > **Settings** > **Control Panel** > **Administrative Tools** > **Data Sources**.

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

64-bit Windows

There is both a 64-bit and 32-bit version of the ODBC Data Source Administrator. The 64-bit version enables you to trace the ODBC API calls that are made by a 64-bit application. The 32-bit version enables you to trace the ODBC API calls that are made by a 32-bit application.

The 64-bit version is located in Control Panel (see the previous step). To run the 32-bit version on Windows 7 and earlier, in the Windows Run dialog box, type:

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

On Windows 8, both the 32-bit and 64-bit ODBC Administrator are located in Control Panel under Administrative tools: ODBC Data Sources (32-bit) and ODBC Data Sources (64-bit).

If you are not sure which version to use, and do not get a log file after completing the steps in this article, try enabling tracing in the other version of ODBC Data Source Administrator.

FROM WITHIN THE DRIVER MANAGER ON UNIX

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

```
Trace = Yes
TraceFile = logfile
```

For example:

```
[ODBC]

Trace = Yes

TraceFile = /tmp/unixodbc.log
```

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

FROM WITHIN THE EASYSOFT ODBC-DB2 DRIVER

Driver manager trace files show all the ODBC calls applications make, their arguments and return values. Easysoft ODBC-DB2 Driver driver tracing is specific to the Easysoft driver and is of most use when making a support call.

FROM WITHIN THE EASYSOFT ODBC-DB2 DRIVER ON UNIX

To enable Easysoft ODBC-DB2 Driver logging, add a `LOGFILE` and a `LOGGING` attribute to the relevant DSN section of the `odbc.ini` file.

For example:

```
[DB2_SAMPLE]

.

.

.

LOGFILE = /tmp/db2-driver.log

LOGGING = Yes
```

The `LOGFILE` value is the path and file name of the log file. The value shown in the example specifies a log file named `/tmp/db2.log`. The `LOGGING` value specifies the actions to log. The value shown in the example specifies that all actions should be logged.

Ensure that the user who is running the application to be traced has write permission to the log file (and to the directory containing it).

By default, the Easysoft ODBC-DB2 Driver appends log information to the file specified by `LOGFILE`. If you want the driver to generate a new log file for each ODBC session, enable logging on a per process basis. To do this, change the `LOGGING` entry to:

```
LOGGING = Process
```

When you set `LOGGING` to `Process`, the Easysoft ODBC-DB2 Driver creates separate log files that only contain trace output related to a particular driver process. The log file name contains the ID of the driver process that the log output is associated with. For example, `/tmp/db2-driver.log.0000266F.B7D766C0`.

FROM WITHIN THE EASYSOFT ODBC-DB2 DRIVER ON WINDOWS

To enable logging:

1. Open the relevant Easysoft ODBC-DB2 Driver data source in ODBC Data Source Administrator.
2. Click **Driver Logging**.
3. Type the file name and path of the file you want the driver to write log information to in the space provided.

APPENDIX B GLOSSARY

Terms and definitions

Application Programmer Interface (API)

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

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.

Batch

A set of SQL statements submitted together and run as a group. A script is often a series of batches submitted one after the other.

Column

The vertical component of a database table. A column has a name and a particular data type (for example, character, decimal, or integer).

Commit

To end a unit of work by releasing locks so that the database changes made by that unit of work can be perceived by other processes. This operation makes the data changes permanent.

Cursor

An entity that maps over a result set and establishes a position on a single row within the result set. After the cursor is positioned on a row, operations can be performed on that row, or on a block of rows starting at that position. The most common operation is to fetch (retrieve) the current row or block of rows.

Data Definition Language

The subset of SQL statements that define all attributes and properties of a database and its objects. DDL statements typically start with CREATE, ALTER, or DROP.

Data Manipulation Language

The subset of SQL statements that is used to retrieve and manipulate data. DML statements typically start with SELECT, INSERT, UPDATE, or DELETE.

Data source

A database or other data repository coupled with an ODBC Driver, which has been given a Data Source Name (see **"DSN" on page 87**) to identify it to the ODBC Driver Manager.

Data type

An attribute that specifies what type of information can be stored in a column, parameter, or variable.

DBMS

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

Distributed query

A single query that accesses data from multiple data sources.

Distributed Relational Database Architecture (DRDA)

The architecture that defines formats and protocols for providing transparent access to remote data.

Driver

See **"ODBC driver" on page 89**.

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. On 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.

DSN-less connection

A type of data connection that is created based on information in a data source name (DSN), but is stored as part of a project or application. DSN-less connections are especially useful for Web applications because they let you move the application from one server to another without re-creating the DSN on the new server.

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.

Host

A computer visible on the network.

Index

In a relational database, a database object that provides fast access to data in the rows of a table, based on key values. Indexes can also enforce uniqueness on the rows in a table. The primary key of a table is automatically indexed. In full-text search, a full-text index stores information about significant words and their location within a given column.

Isolation level

An attribute that defines the degree to which an application process is isolated from other concurrently executing application processes. Isolation levels generally relate to the behavior of an application with respect to locks.

License key

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

NULL

An entry that has no explicitly assigned value. `NULL` is not equivalent to zero or blank. A value of `NULL` is not considered to be greater than, less than, or equivalent to any other value, including another value of `NULL`.

ODBC

Open Database 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.

Record

A group of related fields (columns) of information treated as a unit. A record is more commonly called a row in a relational database.

Result set

A set of row values as returned by, for example, a cursor or procedure.

Row

The horizontal component of a table, consisting of a sequence of values, one for each column of the table.

Structured Query Language (SQL)

A standardised language for defining and manipulating data in a relational database.

SQL-92

The version of the SQL standard published in 1992. The international standard is ISO/IEC 9075:1992 Database Language SQL. The American National Standards Institute (ANSI) also published a corresponding standard (Data Language SQL X3.135-1192), so SQL-92 is sometimes referred to as ANSI SQL in the United States.

Stored procedure

An application program, possibly containing SQL statements, that is stored on the database server and can be invoked with the SQL CALL statement.

Table

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

Transaction

An atomic series of SQL statements that make up a logical unit of work. All of the data modifications made during a transaction are either committed together as a unit or rolled back as a unit.

INDEX

A

C

D

E

F

G

I

L

M

O

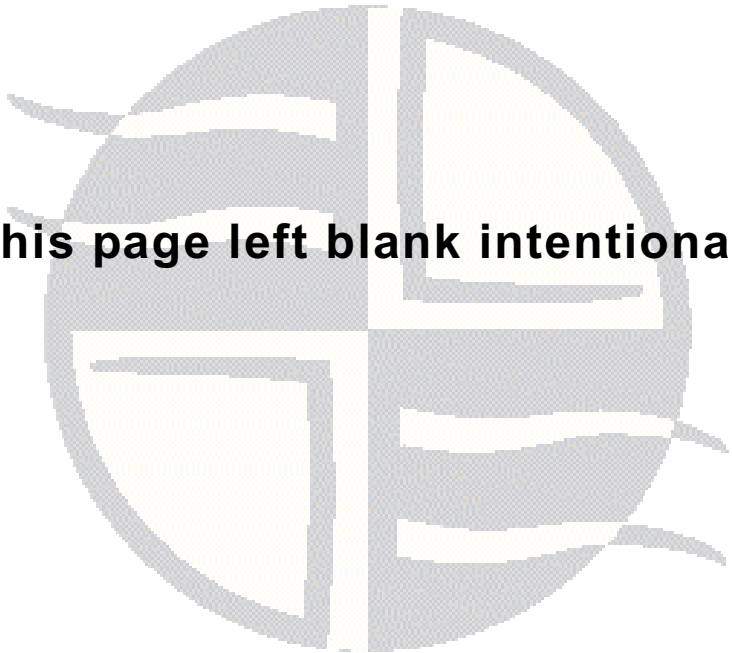
P

R

S

T

U



This page left blank intentionally.