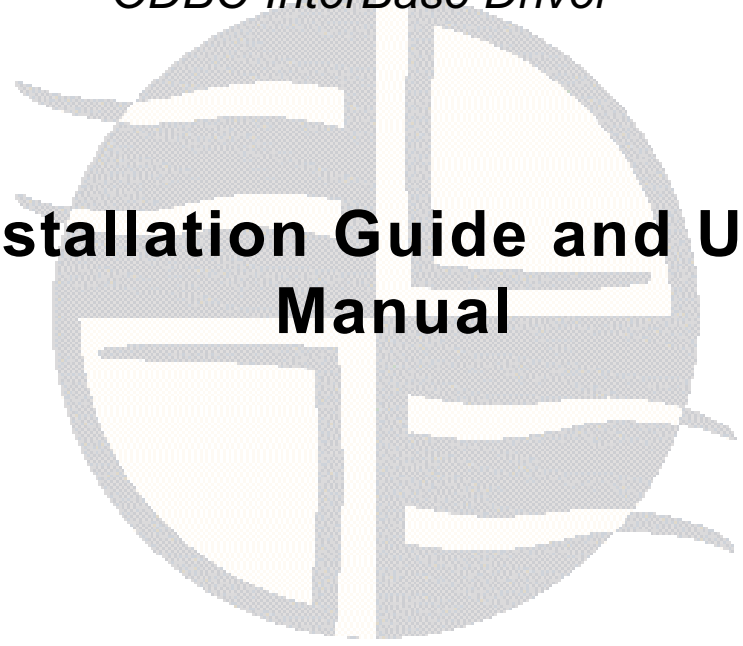
Easysoft Data Access

ODBC-InterBase Driver



Installation Guide and User Manual

Version 4.x.

This manual documents version 4.x of the Easysoft ODBC-InterBase Driver.

Publisher: Easysoft Limited

Thorp Arch Grange

Thorp Arch

Wetherby

LS23 7BA

United Kingdom

Copyright © 1993-2024 by Easysoft Limited.

All rights reserved.

You may not reverse engineer, decompile or disassemble this manual. Information in this manual is subject to change without notice. Companies, names, and data used in examples are fictitious unless otherwise noted.

The names of companies referred to herein, their corporate logos, the names of their hardware and software may be trade names, trademarks or registered trademarks of their respective owners.

Easysoft and the Easysoft logo are registered trademarks of Easysoft Limited.

CONTENTS

List of Figures 5

Preface 7

Intended Audience 8
Displaying the Manual 8
Notational Conventions 9
Typographical Conventions 10
Contents 11
Trademarks 12

Chapter 1 Introduction 13

Overview 14
Feature List 14

Chapter 2 Installation 19

Obtaining the Easysoft ODBC-InterBase Driver 20
What to install 21
Installing on Windows 23
Uninstalling on Windows 30
Installing on Unix 32
Licensing on Unix 43
Uninstalling on Unix 49

Chapter 3 Configuration 53

Configuring the driver 54
Windows Setup 55
Unix Setup 60
Attribute Fields 61

Appendix A Technical Reference 85

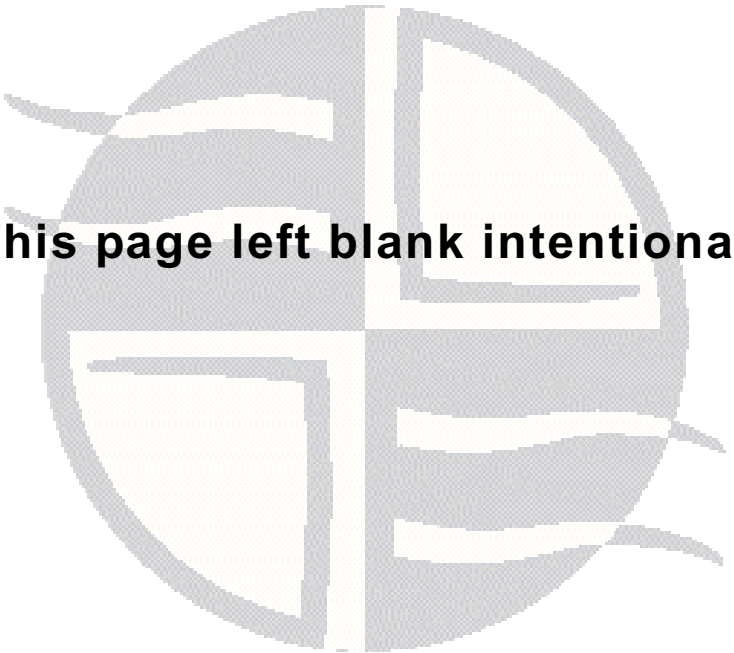
ODBC Compliance 86
Multiple transactions on a single connection 86
InterBase data types 87
Runtime SQL Dialect support 87
Rollback Retaining 87
Chili!soft 87
unixODBC 87

Appendix B Glossary 91

Index 97

LIST OF FIGURES

Figure 1: The License Manager dialog box 25
Figure 2: The ODBC Data Source Administrator dialog box 55
Figure 3: The Create New Data Source dialog box 56
Figure 4: The Easysoft ODBC-InterBase Driver DSN Setup dialog box 57



This page left blank intentionally

PREFACE

About this manual

This manual is intended for use by anyone who wants to access InterBase application data, stored on a Windows or Unix machine, from an ODBC-compliant application.

Chapter Guide

- **Intended Audience**
- **Displaying the Manual**
- **Notational Conventions**
- **Typographical Conventions**
- **Contents**
- **Trademarks**

Intended Audience

The sections written for the Microsoft Windows platforms require some familiarity with the use of buttons, menus, icons and text boxes. If you have any experience of Apple Macintosh computers, Microsoft Windows or the X Window System, you will have no difficulty with these sections.

The Unix-based sections require that you are experienced at using a Unix shell, and can perform basic functions like editing a file. More complex activities are detailed more clearly and do not require any knowledge of specialist Unix shells.

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:

Linux	When using Linux and Solaris 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

- **Introduction**

  An overview of ODBC and the Easysoft ODBC-InterBase Driver.

- **Installation**

  A step-by-step guide to installing the Easysoft ODBC-InterBase Driver.

- **Configuration**

  Explains how to set up and manage user access to client data sources.

- **Appendices**

  Comprising a Technical Reference and a Glossary.

Trademarks

Throughout this manual, *Windows* refers generically to Microsoft Windows 7, 8, 2012 R2, 10, 2016, 2019, 11, 2022 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*.

UNIX is a registered trademark of The Open Group.

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’.

Easysoft and Easysoft Data Access are trademarks of Easysoft Limited.

InterBase is a trademark of the Embarcadero Technologies Inc.

1 INTRODUCTION

Introducing the Easysoft ODBC-InterBase Driver

With Easysoft software you can connect applications on more platforms to more database systems than ever.

Chapter Guide

- **Overview**
- **Feature List**

Overview

The Easysoft ODBC-InterBase Driver provides ODBC 3.5 access to InterBase. The Easysoft ODBC-InterBase Driver is available on a variety of platforms and is compatible with the Windows ODBC Driver Manager and unixODBC, the open source ODBC driver manager.

Data sources created with the driver can be configured to support a wide range of InterBase features.

UNICODE support is provided on some platforms.

Feature List

ODBC Conformance

The Easysoft ODBC-InterBase Driver is a ODBC 3.5 driver providing efficient operation with ODBC 3 applications such as ADO, and full backwards conformance with older ODBC applications.

Support for InterBase data types

All the older IB 5.x data types are supported, as well as the new IB 6 data types (such as `BIGINT`), numeric storage as floating types (IB 5) and exact precision types (`SHORT`, `INT`, `BIGINT`) in IB 6.

Native InterBase data types such as `BLOB`, `SUB`, `TYPE` and `TEXT` are mapped to the correct ODBC data type (`SQL_LONGVARCHAR` in this case), but the native format is still available. The InterBase 7 `BOOLEAN` datatype is also implemented to the SQL 99 standard. and is available when the driver is connected to a InterBase 7 database via the InterBase 7 GDS library.

LONGER METADATA NAMES

Support is provided for longer Metadata names (67 bytes plus a null terminator), when the driver is connected to a InterBase 7 database via the InterBase 7 GDS library.

METADATA CALLS

Direct access is provided to all the InterBase catalog tables. This information is also available via the standard ODBC calls such as `SQLTables` and `SQLColumns`.

INTERBASE ROLES

Support is provided for Roles, both via the DSN setup and as an option to the `SQLDriverConnect` function.

UNICODE

The driver provides both ANSI and UNICODE access to InterBase databases, converting from the internal UTF8 storage format used by the InterBase server.

Default character sets may be configured and all conversions are done transparently and automatically. ANSI only applications such as Microsoft Query will only “see” 8 Bit ANSI data, while Unicode applications such as Microsoft Access are provided with direct access to 16-Bit Unicode data.

SUPPORT FOR OLDER INTERBASE DATABASES

Due to changes in the metadata storage for older (pre 6.0) InterBase databases, the Easysoft ODBC-InterBase Driver will detect the InterBase database version in use at run time and transparently adjust its operation to match the available data.

In cases where the detection fails (for instance with partially upgraded databases) the version may be manually configured.

Support is also provided for older InterBase interface libraries and the operation of the driver will be adjusted to ensure the most efficient execution.

For example, if the InterBase `rollback_retaining` function is available, it will be used by the Easysoft ODBC-InterBase Driver (with potential performance improvements), but if it is absent the driver will adjust its operation in such a way that the calling application is unaware of the change in server capability.

The InterBase interface libraries prior to version 7 were not thread safe, so if the driver detects it is being used with a version of the library before 7, the driver will ensure thread safe operation by employing exclusion primitives to only allow single thread access to the library. Because of this multi threaded applications may show performance improvements when using the driver with the InterBase7 GDS library.

SUPPORT FOR OLDER SQL DIALECTS

The required SQL Dialect may be configured, and the Easysoft ODBC-InterBase Driver will adjust its operation to ensure that only SQL that matches the selected dialect is passed to the InterBase server.

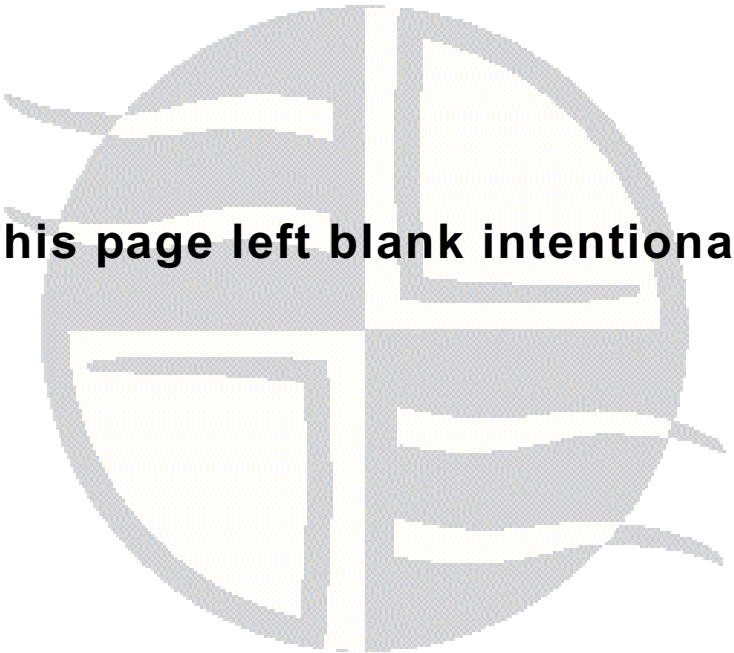
This enables the Easysoft ODBC-InterBase Driver to be used with many versions of InterBase and it has been successfully used with versions from 4 to 2020.

TRANSACTIONS

The driver maps the ODBC transaction interface to the provided InterBase model in a transparent and efficient manner. There is also provision to select the behavior on detection of lock conflicts. The driver provides “wait” and “no wait” operations.

STORED PROCEDURE SUPPORT

The driver supports InterBase stored procedures and provides mapping from the ODBC call methods to the native InterBase calls. Provision is made for both the “`SELECT`” and “`EXEC PROCEDURE`” calling methods and the driver will pick the correct translation. A manual override on this selection is provided for unusual procedures.



This page left blank intentionally

INSTALLATION

2

Installing the Easysoft ODBC-InterBase Driver

This section explains how to install, license and remove the Easysoft ODBC-InterBase 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

- **Obtaining the Easysoft ODBC-InterBase Driver**
- **What to install**
- **Installing on Windows**
- **Licensing on Windows**
- **Uninstalling on Windows**
- **Installing on Unix**
- **Licensing on Unix**
- **Uninstalling on Unix**

Obtaining the Easysoft ODBC-InterBase Driver

There are three ways to obtain the Easysoft ODBC-InterBase 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-InterBase 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 free documentation and beta releases of Easysoft products, as well as definitive releases.

  Change to the `pub/interbase` 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 distribution file for the Easysoft ODBC-InterBase Driver varies from platform to platform, but is of the form:

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

– OR –

- `odbc-interbase-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*" will vary depending on the operating system distribution you require and you may come across files of the form:

- `odbc-interbase-x.y.z-platform -variation.tar`

within specific Unix platforms, where "*platform-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 using 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-InterBase 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 23**
- **"Uninstalling on Windows" on page 30**
- **"Installing on Unix" on page 32**
- **"Uninstalling on Unix" on page 49**

After installing the Easysoft ODBC-InterBase Driver you can then add or amend any of the available attribute fields which you require in order to correctly configure your data source (see **"Configuring the Easysoft ODBC-InterBase Driver" on page 53** for details of the Easysoft ODBC-InterBase Driver DSN setup).

Installing on Windows

INSTALLING THE INTERBASE CLIENT

The Easysoft ODBC-InterBase Driver uses the InterBase client to connect to InterBase. The InterBase client is installed by executing the InterBase database distribution. Select the Client Only installation option to install just the InterBase client.

64-bit Windows

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

You need to use the 64-bit Easysoft ODBC-InterBase Driver with a 64-bit InterBase client (`ibclient64.dll`). You need to use the 32-bit Easysoft ODBC-InterBase Driver with a 32-bit InterBase client (`gds32.dll`).

Note that you can use the 64-bit Easysoft ODBC-InterBase Driver to connect to either a 32-bit or a 64-bit InterBase server. Likewise, you can use the 32-bit Easysoft ODBC-InterBase Driver to connect to either a 32-bit or a 64-bit InterBase server.

INSTALLING THE EASYSOFT ODBC-INTERBASE DRIVER

- Execute the file distribution that you downloaded in **"Obtaining the Easysoft ODBC-InterBase Driver" on page 20**

  Follow the on screen instructions.

Updating Files That are in Use

To avoid rebooting your computer, the Easysoft ODBC-InterBase 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-InterBase 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-InterBase 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-InterBase 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.

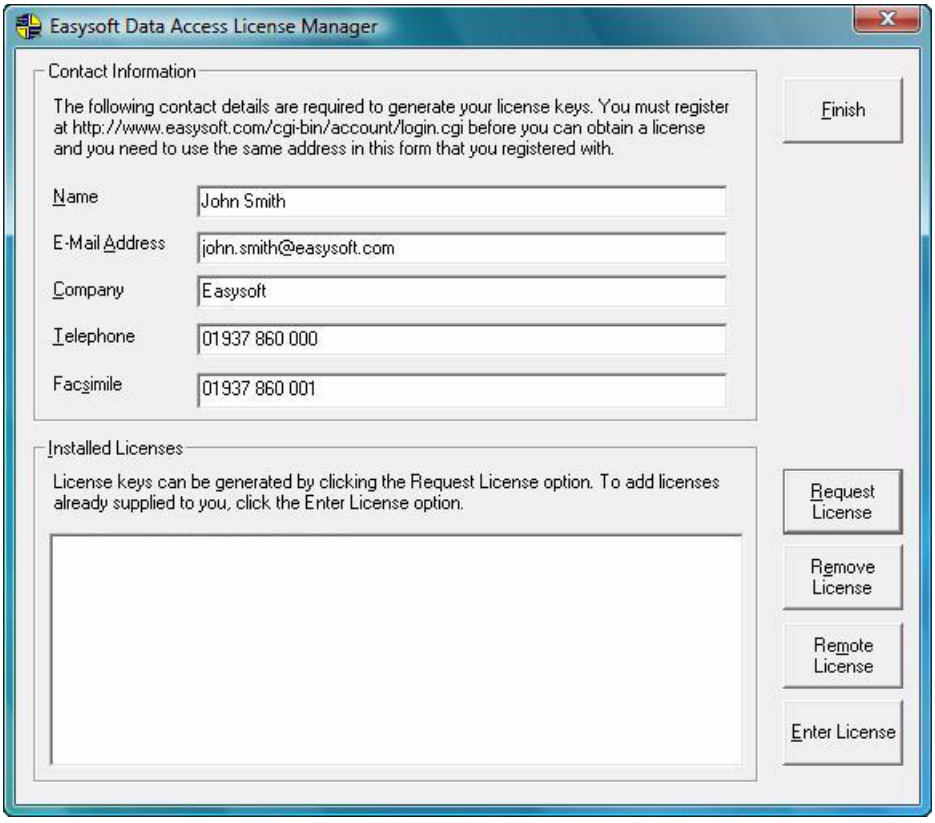


Figure 1: The License Manager dialog box

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.

3. 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-InterBase 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**.

   The License Manager displays a summary of the information you entered 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. You can now go to **step on page 28**.

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

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

   The installation is complete.

   You should have a new **Start > All Programs > Easysoft > InterBase ODBC** program group with links to the Easysoft InterBase home page and this manual.

REPAIRING THE EASYSOFT ODBC-INTERBASE DRIVER INSTALLATION

The installer can repair a broken Easysoft ODBC-InterBase Driver installation. For example, you can use the installer to restore missing Easysoft ODBC-InterBase 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-InterBase Driver**, and then click **Repair**.

In earlier versions of Windows:

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

Uninstalling on Windows

Removing Easysoft ODBC-InterBase Driver Data Sources

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

In Windows Vista and later versions of Windows:

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

In earlier versions of Windows:

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

The uninstall process is complete.

Any licenses you obtained for the Easysoft ODBC-InterBase 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

Although this section covers a range of platforms and the precise output may vary from system to system, the installation process is essentially the same.

INSTALLATION REQUIREMENTS

The Unix installation routine has the following requirements:

- The Bourne shell (or BASH) must either be named (or linked to) `/bin/sh` or the first line of the `install` file updated to the correct location.
- Various common Unix commands such as `grep`, `awk`, `cut`, `ps`, `sed`, `cat`, `wc`, `uname`, `tr` and `find`.

  If any of these commands are missing they can be obtained from the Free Software Foundation (**http://www.fsf.org**).
- Depending on the platform, you will need up to 10Mb of free disk space for the installed programs and up to 10Mb temporary space for the installation files themselves.

PREPARATION

1. Log on to your Unix machine as the `root` user.
2. Download the Easysoft ODBC-InterBase Driver (see).**"Obtaining the Easysoft ODBC-InterBase Driver" on page 20**
3. Place the distribution file in a temporary directory on your Unix machine.

EXTRACTING THE INSTALLATION FILES

4. Change to the directory in which the distribution file resides.
5. Uncompress the installation files from the distribution file:

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

   `gunzip odbc-interbase-`*`x.y.z-platform`*`.tar.gz`

   – OR –

   If the distribution file has been compressed with `bzip`2 (i.e. the filename ends in `.bz2`), then use:

   `bunzip2 odbc-interbase-`*`x.y.z-platform`*`.tar.bz2`

   – OR –

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

   `uncompress odbc-interbase-`*`x.y.z-platform`*`.tar.Z`

6. If the distribution file has not been compacted at all (i.e. the filename ends in `.tar`), then it is ready for extraction:

   `tar -xvf odbc-interbase-`*`x.y.z-platform`*`.tar`

   The `tar` program creates a directory with the same name as the `tar` file (without the final `.tar`) containing further archives, checksum files, a script called `install` and a text file called `INSTALL.txt`.

It also contains a versioned directory to ensure that any shared components already installed by other Easysoft products are only overwritten if those included in this distribution are newer.

NB If you do not wish to keep the original downloaded distribution file you can now delete it safely.

7. Change into the newly-created `odbc-interbase-x.y.z-platform` directory.

Caution! Check through the `INSTALL.txt` file before continuing. It gives full installation instructions for the Unix-literate, and if you are confident in the use and administration of your system, you can follow the instructions in the `INSTALL.txt` file instead of working through the remainder of this section.

8. There are copies of the license agreement file provided in the archive.

Caution! You must read and accept the terms of the license to use the software. The license text can be found in the file `license.txt`, be sure to understand the terms before continuing.

BEGINNING THE INSTALLATION

9. Type:

   ```
   ./install
   ```

   NB

   During the installation, you are asked to answer some questions. The default response is displayed in square brackets [], which you can press *<Enter>* to accept or you can choose any of the alternative responses shown in round brackets () by typing the required response and then pressing *<Enter>*.

   Occasionally, the install program pauses to give you time to read the information displayed on screen. Press *<Enter>* to continue when you have read the current screen of information.

10. If the installation locates a program which can display the license (e.g. your **PAGER** environment variable is set or 'more' or 'less' is found) then the license will be displayed and you will need to page to the end and quit the pager program. Once you have read and agree to the **Easysoft License Agreement**, type `yes` and then press *<Enter>* to continue..

    NB You must type `yes`, not `y`, to continue.

11. The script pauses to allow you to read its output so far. Up to this point it has checked the following:

    - that you have the minimum set of Unix programs it requires
    - the platform you are running
    - any platform-specific components, such as the version of the C runtime library

12. Press `<Enter>` to continue.

    The script checks the archive package, with three possible outcomes:

    - The files are checked and they pass
    - The files are checked and they fail
    - The files are not checked because some component required for the check is not found

13. If the check failed because of missing components, enter `y` to continue regardless or `n` to quit and investigate the missing components.

    – OR –

    If the check was carried out and the files failed then the files have been damaged.

    Either:

    - return to **"Obtaining the Easysoft ODBC-InterBase Driver" on page 20** and download the install archive again

    – OR –

    - call the Easysoft support team (see **Contact Details** for more information)

    If the files passed the check then you are asked for a directory in which to place the `easysoft` installation directory tree.

    The default is `/usr/local`, which would be the normal location to install software for system-wide use.

    If you wish to install the software in another directory then specify a directory name here.

If you have any other Easysoft products already installed, then choose the same directory that you chose for the other Easysoft product(s).

The installation script will create a directory called `easysoft` under the directory you specify here, into which all Easysoft ODBC-InterBase Driver files will be placed.

NB	The installation script can accept a path to a directory that does not exist, provided a parent directory exists. For example, if the directory `/usr/local` exists on your machine, the script can accept the non-existent directory `/usr/local/odbc` and will create it, but will not create `/usr/local/odbc/interbase`. The script always creates an `easysoft` directory beneath the specified path.

14. If you have `root` permission and want a typical system-wide installation, press *<Enter>* to place the `easysoft` directory in `/usr/local/`.

    – OR –

    If you do not have `root` privileges, or wish to select a custom installation directory, type the desired directory and press *<Enter>*.

    The script now creates the `easysoft` directory under the directory you specified. If your chosen directory does not exist, you will be asked whether or not the script should create it. Enter `y` or `n`.

    If the chosen directory already contains an `easysoft` directory, the script warns that you may be installing over a previous Easysoft ODBC-InterBase Driver installation (this can also arise if you have another Easysoft product installed).

Normally you should continue with the installation to install into the existing `easysoft` directory. Enter `y` or `n`.

If you chose a directory other than `/usr/local` the script creates a symbolic link `/usr/local/easysoft`, pointing to the `easysoft` directory in the directory you specified, in order to ensure that licensing will work.

15. Press *<Enter>* to continue.

INSTALLING UNIXODBC AND COMMON FILES

Installing unixODBC permits ODBC applications on the local machine to choose a data source at runtime, as the driver manager loads the required driver..

NB	unixODBC is an open source project sponsored by Easysoft and other industry members which is rapidly becoming the standard driver manager across the Unix data access community. Easysoft recommend it for use in all Unix ODBC installations (see **"unixODBC" on page 87**).

You should install the unixODBC driver manager if:

- the application(s) you will be using with ODBC specifically support unixODBC (e.g. Perl, PHP, Applixware)
- you already have other ODBC drivers installed on this machine
- you intend using other Easysoft products in combination with the Easysoft ODBC-InterBase Driver
- you want to use the unixODBC GUI configuration tools to define DSNs

In general, Easysoft recommend installing unixODBC, as there is little to be gained by not doing so.

The installation will examine your system in an attempt to locate an already installed unixODBC. If unixODBC is already installed, the installation script can use your existing unixODBC, but you should bare in mind the following possible complications of doing this:

- If there are multiple copies of unixODBC on your system already you will have to choose one of them. If you pick the wrong one (i.e. not the one your applications are linked with, or not the one your run-time linker uses) the Easysoft ODBC-InterBase Driver ODBC driver will not be visible to your applications until this is corrected.
- If you have built unixODBC yourself from sources you must make sure it has been configured correctly for use with your applications. For example, the unixODBC configuration script defaults to building a version linked with `pthreads` which is incompatible with Perl (on some platforms) and Apache/PHP (prior to Apache 2).

Easysoft ODBC-InterBase Driver distributions contain matched builds of unixODBC and the Easysoft ODBC-InterBase Driver, so installing the incorporated unixODBC build is often the safest policy.

16. Press *<Enter>* to continue and then choose whether to install unixODBC.
17. If you do not wish to install unixODBC, enter `n` and skip to **"Licensing on Unix" on page 43**.

    – OR –

    If you wish to install unixODBC, enter `y` at the prompt.

18. The script pauses at this point. Press *<Enter>* to continue.

    The script extracts the unixODBC files.

19. The script pauses again. Press *<Enter>* to continue.

    ### CONFIGURING THE EASYSOFT ODBC-INTERBASE DRIVER IN UNIXODBC

    If unixODBC was installed as part of this installation, or unixODBC was found to be already installed, the script will offer to configure the InterBase Driver for use with unixODBC.

    This allows any ODBC application which is using the unixODBC driver manager to connect to remote data sources through the InterBase Driver.

20. If you do not want to set up the Easysoft ODBC-InterBase Driver under unixODBC or you do not have unixODBC, enter `n` and skip to **"Licensing on Unix" on page 43**.

    – OR –

    If you have unixODBC installed on your system, and wish to use it with your InterBase Driver, enter `y`.

    The script runs `odbcinst`, the command for installing data sources under unixODBC.

21. If the `odbcinst` program is not in its expected location, you are asked to provide the directory into which unixODBC was installed.

    If you do not know the answer, enter q to abandon the attempt to configure unixODBC and skip to **"Licensing on Unix" on page 43**.

    – OR –

    If known, supply the unixODBC install path.

The script checks whether or not the Easysoft ODBC-InterBase Driver has already been configured into unixODBC.

22. If the unixODBC driver manager has already been configured for the Easysoft ODBC-InterBase Driver then a warning is given.

Caution!

If unixODBC has been previously configured with the Easysoft ODBC-InterBase Driver then it will continue to access the InterBase Driver software from the directory in which it was originally installed.

If the current install directory is different then unixODBC will pick up the old version of the Easysoft ODBC-InterBase Driver, resulting in incorrect behavior.

The script now calls `odbcinst` with the relevant details for the InterBase Driver.

NB

The messages `"Target directory is /usr/lib"` and `"added to ODBC.INI"` displayed whilst the `odbcinst` script is running are inaccurate and should be ignored.

If this part of the installation fails, refer to the unixODBC documentation for details of how to manually reconfigure the driver manager.

REF

Comprehensive unixODBC documentation can be found at **http://www.unixodbc.org**.

The settings that the install script attempted to set can be found in the file `unixODBC.template`.

The install script now offers to create a example data source under unixODBC that can be used as a template for your actual InterBase connections.

23. If you do not want to create a example data source, enter `n` at the prompt and go to **"Licensing on Unix" on page 43**.

    – OR –

    If you want to create a example data source, then enter `y` and continue.

    The script calls `odbcinst` again, this time with the data source details.

    Once the installation is complete, you will have a example data source that can be modified to connect to your own InterBase databases

Licensing on Unix

24. The script prepares to license the product and then asks if you would like to run the License Manager.

    If you intend to license the software after finishing the installation, enter `n` and go to **"Completing the installation" on page 45**.

    – OR –

    If you want unrestricted use of the software, either via a free time-limited trial license or with a full (purchased) license, enter `y` and continue.

    The License Manager displays a menu of options.

25. Enter the number corresponding to the Easysoft ODBC-InterBase Driver.

    The License Manager requests some contact information.

26. Enter your **Name**.

27. Enter your **Company Name**.

28. Enter your **Email**, **Phone** and **Fax** (if applicable).The **E-Mail Address MUST** be the same as the address used to register and download from the easysoft web site.

29. At the **Ref** prompt, do either of the following:

    - If you want a trial license, leave it blank and press <*Enter*>
    - If you want a full license, enter your authorization code and then press <*Enter*>

30. The License Manager displays a menu of options for acquiring your license.

If you have an internet connection you should select `1, Automatic`.

This is the quickest and easiest method, unless your firewall prevents the message getting through.

> **NB** Opening up port `8884` will allow you to get around this if you wish. You may need to consult your system administrator to do this

If you do not have an internet connection, or the `Automatic` option fails, select option `2` to write the required information to a text file in the current working directory called `license_request.txt`, which you will then need to view to obtain a license.

31. The License Manager displays the menu again. Select `0` to exit.

> **NB** The **View Existing Licenses** option will not display any license you obtained during this License Manager session. You must quit the License Manager and allow the install script to finish applying the license.

If you chose the automatic licensing method and licenses are retrieved, the licensing script outputs them to a file called `licenses.out`. If the install script detects possible new licenses in the file `licenses.out` it asks whether you want to add them to the license database.

32. If you obtained licenses and you want to add them now, enter `y`.

    – OR –

    If you did not obtain licenses, enter `n`.

    – OR –

If you do not see this message, then read on. You will need to complete the offline licensing procedure as explained in **"Completing the offline licensing procedure" on page 45**.

COMPLETING THE INSTALLATION

33. Press *<Enter>* if necessary to return to the shell prompt.

    Don't forget:

    - If the install failed to register the InterBase Driver with unixODBC then you should do so manually now.

34. You can remove the installation files. Unless you specified the installation directory (at **step 14 on page 37**) to be within the temporary directory, then you can safely remove the temporary directory and all its contents.

NB	Any warnings generated during the installation are appended to the `./warnings` file and a message output containing a list of all the warnings. You should review this file and satisfy yourself that none of the warnings have adversely affected your installation. You should mail this file to Easysoft support if you are unsure. In particular, one warning you may see is caused by the installation attempting to untar a file which is in use. If you see this warning it will be necessary to make sure all applications using the file in question are stopped and the installation is rerun.

COMPLETING THE OFFLINE LICENSING PROCEDURE

If at **step 30 on page 43** you chose to write the licensing information to a file, you still need to complete the licensing procedure before the Easysoft ODBC-InterBase Driver can be used.

The licensing information is written to the `license_request.txt` file. This file contains information including a machine number (a number unique to your machine) which Easysoft require before a license key can be issued.

1. Do any one of the following:

   - Display the `license_request.txt` file (e.g. using `cat license_request.txt`) and note the machine number. Now run a web browser and go to:

     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.

   - Email the file `license_request.txt` to **autolicense@easysoft.com**. Your license key(s) will be emailed to you automatically.

   - Email the file `license_request.txt` to **license@easysoft.com**. A member of the Licensing Department will email the license keys(s) to you.

2. When you receive your license key(s), append them to the file `/usr/local/easysoft/license/licenses`, removing any `LIC:` prefixes.

NB	The Easysoft license responder puts an attachment in its outgoing emails that allows Windows users to activate their licenses with a double-click. If you read your email in Windows, this attachment will be visible but it will not work for licensing Unix versions of the software.

The Easysoft ODBC-InterBase Driver is now licensed and you can begin using it.

If you need further information about licensing, please refer to the **Licensing Guide**.

USING THE EASYSOFT ODBC-INTERBASE DRIVER ON A PLATFORM OTHER THAN LINUX

If you are installing to a non-Linux platform, before you can use the InterBase Driver you must set and export the shell environment variable LD_LIBRARY_PATH, LD_RUN_PATH, DYLD_LIBRARY_PATH or SHLIB_PATH to

`<InstallDir>/easysoft/ib:<InstallDir>/easysoft/lib`

where `<InstallDir>` is the directory in which you chose to install the Easysoft ODBC-InterBase Driver.

If you accepted the default install directory you might use:

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

NB

The exact command you need to use to set and export an environment variable depends on your shell and the variable that needs to be set depends on the run-time linker used on your Unix platform. Refer to the platform documentation for your current shell and `ld(1)`, `dlopen` or `ld.so(8)`.

If unixODBC has been installed then you will also need to add *<InstallDir>*`/easysoft/unixODBC/lib` to your LD_LIBRARY_PATH, LD_RUN_PATH, DYLD_LIBRARY_PATH or SHLIB_PATH shell environment variable.

Uninstalling on Unix

To uninstall the Easysoft ODBC-InterBase Driver under Unix:

- If unixODBC is installed, the Easysoft ODBC-InterBase Driver driver must be removed from its database.

Caution!

This is only required if the Easysoft ODBC-InterBase Driver is not going to be upgraded, or is to be upgraded using a different configuration.

If the Easysoft ODBC-InterBase Driver is going to be upgraded with the existing configuration then do NOT make these changes.

- If the system has a dynamic linker (such as `ld.so` on Linux), the Easysoft ODBC-InterBase 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-InterBase 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. Close down all client programs attached to your service.
2. Log in as `root`.

Removing from unixODBC

3. Check whether the Easysoft ODBC-InterBase Driver is configured under unixODBC by typing:

```
odbcinst -q -d
```

4. If "`INTERBASE`" is returned in the output then remove the Easysoft ODBC-InterBase Driver by typing:

   ```
   odbcinst -u -d -n INTERBASE
   ```

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

5. If you have the file `/etc/ld.so.conf` file, make a backup copy, e.g.

   ```
   cp /etc/ld.so.conf /etc/ld.so.conf.save
   ```

6. Open `/etc/ld.so.conf` and manually remove the path to the Easysoft ODBC-InterBase Driver client shared objects. The line is of the form:

   <InstallDir>`/easysoft/ib`

7. If you are not using any other Easysoft software then you may remove the path to the common Easysoft shared objects:

   <InstallDir>`/easysoft/lib`

8. If you are no longer using unixODBC then you can also remove the reference:

   <InstallDir>`/easysoft/unixODBC`

9. 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 your system's hard drive.

10. 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-InterBase Driver.

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

11. Remove the Easysoft ODBC-InterBase Driver installation directory:

    `ls`

    Check that you are in the right directory.

    `rm -r ib`

    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`

12. 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.

13. If you left the Easysoft ODBC-InterBase 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-InterBase Driver and other Easysoft products are stored in the *<InstallDir>*`/easysoft/license/licenses` file.

    After uninstalling the Easysoft ODBC-InterBase 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-InterBase Driver

In terms of the Easysoft ODBC-InterBase Driver, the client is the machine running the ODBC application with which you want to access data on your InterBase server.

To allow an ODBC application on the client machine to access data on the remote server, you need to create a data source on the client.

This data source uses the Easysoft ODBC-InterBase Driver and specifies the attributes required to connect to the InterBase database on the server.

Before setting up a data source on your client machine, you must have successfully installed the Easysoft ODBC-InterBase Driver on this machine.

Instructions for installing the Easysoft ODBC-InterBase Driver on Windows and Unix platforms are provided in **“Installation” on page 19**.

Chapter Guide

- **Configuring the driver**
- **Windows Setup**
- **Unix Setup**
- **Attribute Fields**

Configuring the driver

This section outlines how to configure the attribute fields which are available to be set for the Easysoft ODBC-InterBase Driver.

Attribute fields are configured either:

- by entering values into the DSN setup dialog box (Windows or Unix GUI)

– OR –

- by text-editing the `odbc.ini` file (Unix only)

Refer to the section relevant to your platform to continue:

- **“Windows Setup” on page 55**
- **“Unix Setup” on page 60**

Windows Setup

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

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

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

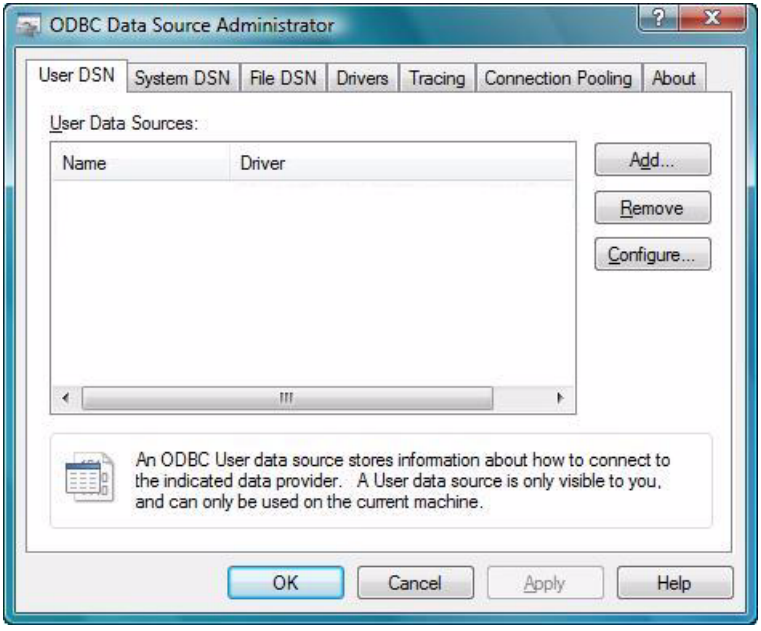


Figure 2: The ODBC Data Source Administrator dialog box

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:

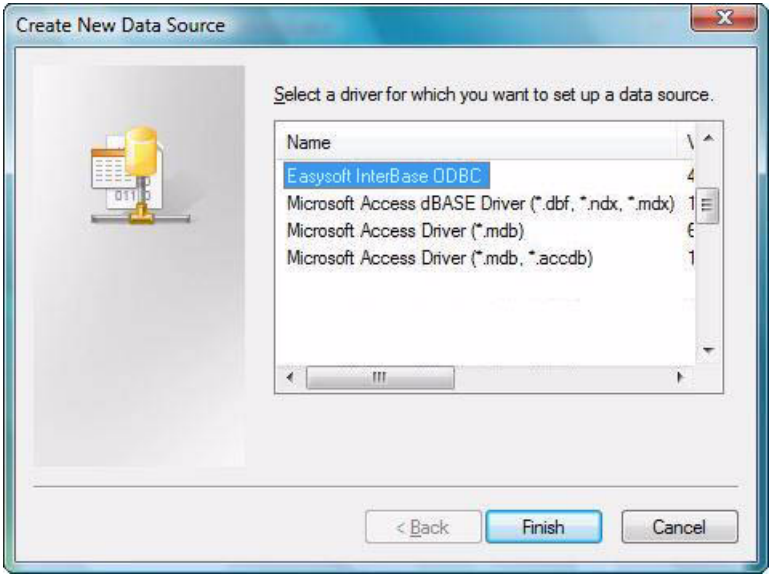


Figure 3: The Create New Data Source dialog box

4. Select `Easysoft InterBase ODBC` and click **Finish**.

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

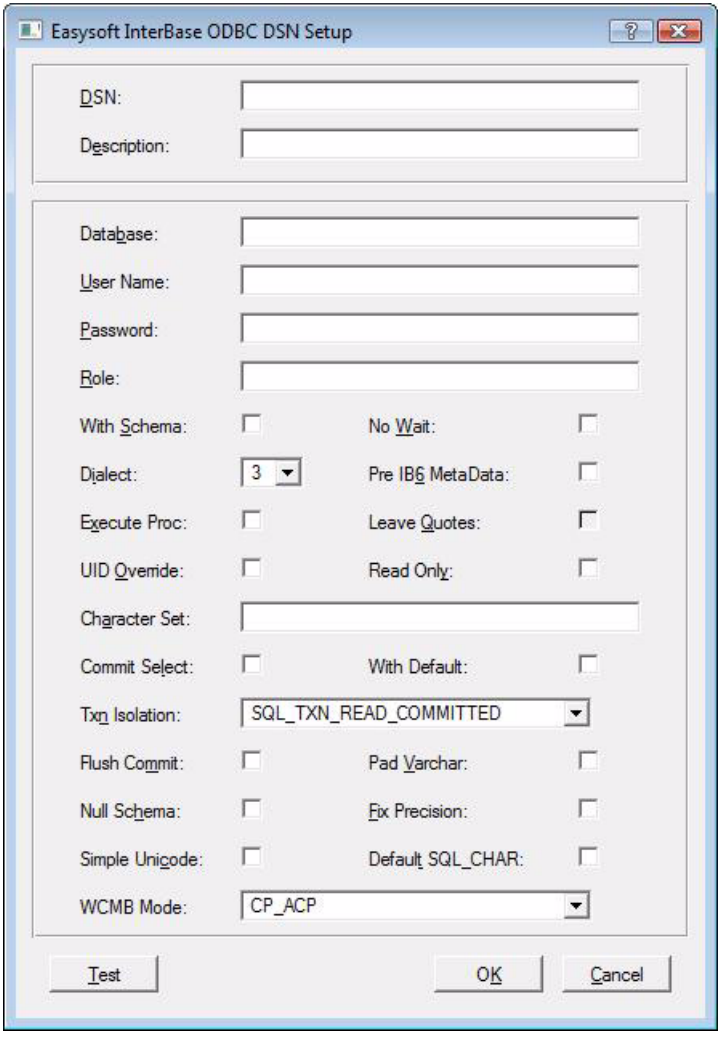


Figure 4: The Easysoft ODBC-InterBase Driver DSN Setup dialog box

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

For details of the other attributes that can be set on this dialog box, see **"Attribute Fields" on page 61**.

64-bit Windows

The Easysoft installer program installs both a 32-bit and a 64-bit version of the Easysoft ODBC-InterBase Driver. If you want to use a 64-bit ODBC application, you need to use the 64-bit Easysoft ODBC-InterBase Driver. If you want to use a 32-bit ODBC application, you need to use the 32-bit Easysoft ODBC-InterBase 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-InterBase Driver data sources created in the 64-bit ODBC Administrator will specify the 64-bit version of the Easysoft ODBC-InterBase Driver. Easysoft ODBC-InterBase Driver data sources created in the 32-bit ODBC Administrator will specify the 32-bit version of the Easysoft ODBC-InterBase Driver.

System data sources created in the 64-bit ODBC Administrator are only visible to 64-bit applications. If you want to create an Easysoft ODBC-InterBase Driver System data source for use with a 64-bit application, use the 64-bit ODBC Administrator therefore. Likewise, System data sources created in the 32-bit ODBC Administrator are only visible to 32-bit applications.

64-bit Windows

(The reason for this is that System data sources created in the 64-bit ODBC Administrator are stored in a registry key called HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI. System data sources created in the 32-bit ODBC Administrator are stored in a registry key called HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI. The WOW64 layer intercepts registry calls to HKEY_LOCAL_MACHINE\Software that are made by 32-bit applications, and then redirects them to the HKEY_LOCAL_MACHINE\Software\WOW6432node key.)

User data sources are visible to both 32-bit and 64-bit applications, irrespective of the version of ODBC Administrator they were created in. If a 64-bit application connects to an Easysoft ODBC-InterBase Driver User data source created in the 32-bit ODBC Administrator, it will load the 64-bit version of the Easysoft ODBC-InterBase Driver. Likewise, a 32-bit application that connects to a 64-bit Easysoft ODBC-InterBase Driver data source will load the 32-bit version of the Easysoft ODBC-InterBase Driver.

Unix Setup

1. Using the sample DSN entry created during the install as a template, create a DSN entry in the driver manager odbc.ini file. The entry will look something like the following.

```
[IB_SAMPLE]
Driver      = INTERBASE
Description = InterBase driver
Database    = server:/path/to/database.gdb
User        = test
Password    = easysoft
With_Schema = 0
Dialect     = 3
Charset     =
Role        =
Nowait      = 0
OldMetaData = 0
ExecProc    = 0
DQuote = 0
UIDOverride = 0
```

2. Amend the `Database`, `User` and `Password` attributes in the new `odbc.ini` section to suit your specific user installation.

   For details of the other attributes that can be set in `odbc.ini`, see **"Attribute Fields" on page 61**.

3. The installation will create a odbcinst.ini entry that allows the driver manager to use the InterBase Driver To do this manually, add a entry containing the following entries.

```
[INTERBASE]
Description = Easysoft InterBase ODBC Driver
Driver      = /usr/local/easysoft/ib/libibodbc.so
Setup       = /usr/local/easysoft/ib/libibodbcS.so
FileUsage   = 1
DontDLClose = 1
```

4. Amend the `Driver and Setup` attributes to the reflect the install directory you chose in **step 14 on page 37**.

Attribute Fields

This section lists the attributes which can be set for the Easysoft ODBC-InterBase 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 NAME

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

DSN 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 NAME

The database name passed to the InterBase `Connect` function.

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

This attribute may be entered in several formats, depending on where the database is located and the connection method required:

Format	Connection Method	Examples

db_file	Local	`C:\CodeGear\InterBase\examples\database\employee.gdb`
		Connects to the sample InterBase database on Windows. The InterBase server is installed on the same machine as the InterBase client.
		`/opt/interbase/examples/employee.gdb`
		Connects to the sample InterBase database on Linux/Unix. The InterBase server is installed on the same machine as the InterBase client.

Format	Connection Method	Examples

server:db_file	TCP/IP	`my_remote_database_server:C:\CodeGear\InterBase\examples\database\employee.gdb`
		Connects to the sample InterBase database on Windows. The InterBase server is installed on a different machine (`my_remote_database_server`) to the InterBase client.
		`192.168.0.10:/opt/interbase/examples/employee.gdb`
		Connects to the sample InterBase database on Linux/Unix. The InterBase server is installed on a different machine (`192.168.0.10`) to the InterBase client.

server/instance:db_file	TCP/IP	`my_remote_database_server/instance2:C:\CodeGear\InterBase\examples\database\employee.gdb`
		Connects to the sample InterBase database on Windows. This InterBase server supports multiple instances. The sample database is served by a instance named `instance2`.
		`192.168.0.10/instance2:/opt/interbase/examples/employee.gdb`
		Connects to a named instance serving the sample InterBase database on Linux/Unix.

The Easysoft ODBC-InterBase Driver may prompt for this attribute if no value is supplied.

- Connecting to a local database (i.e. on the same machine as the client):

  The database string contains only the location of the InterBase database.

For example, on Windows:

```
DATABASE=c:\programfiles\borland\interbase\
examples\database\employee.gdb
```

or Linux:

```
DATABASE=/opt/interbase/isc4.gdb
```

As this type of connection is made in the process context of the client application, it can fail if the privileges of a user do not permit database access (e.g. if the connection is being made by a Web Server).

- Specifying a remote database connection:

  The client process uses a network protocol to connect to a server process, which then connects to the server database, so avoiding many problems with permissions encountered when connecting to a local database.

  The server may be located either on a remote computer, or on the same computer as the client and the communication protocol used may be specified by the contents of the database string.

  To establish a connection using TCP/IP, prefix the database with "`servername:`" (where "`servername:`" resolves via the normal DNS name lookup to a machine on the network).

  It may be possible to use a dotted quad IP address instead of "`servername`", but this is dependent on the IP protocol stack being used.

  For example, to connect via TCP/IP to a server on a Windows machine:

```
DATABASE=servername:c:\program files\
borland\interbase\examples\database\employee.gdb
```

or Linux:

```
DATABASE=servername:/opt/interbase/isc4.gdb
```

- Connecting via NetBEUI/Named Pipes:

```
DATABASE=\\servername\c:\program
files\borland\interbase\examples\database\employee.gdb
```

USER NAME

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

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

Note that this attribute must contain a valid user name for the InterBase server database.

If a value is specified in the connect string then any entry in the DSN will be ignored (see **“UID Override” on page 74**).

The Easysoft ODBC-InterBase Driver may prompt for this attribute if no value is supplied.

PASSWORD

The password supplied to InterBase to authenticate the connection

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

If a value is specified in the connect string then any entry in the DSN will be ignored (see **"UID Override" on page 74**).

The Easysoft ODBC-InterBase Driver may prompt for this attribute if no value is specified.

ROLE

An optional InterBase role by which a user can connect.

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

See the InterBase documentation for more details on the use of roles.

SCHEMA INFORMATION

Specifies that owner and schema information should be returned from metadata calls.

Interface	**Value**
DSN Dialog Box (Windows)	With Schema
odbc.ini file (Unix)	With_Schema=0|1
Connect String	SCHEMA=0|1

The InterBase metadata store contains a value for the Owner or Schema of a table.

So, for example, if the following CREATE TABLE statement was executed while logged in as USR1:

```
CREATE TABLE TEST1 (I INTEGER NOT NULL PRIMARY KEY)
```

a table called TEST1 would be created and its schema would be set to USR1.

However, InterBase does not have the ability to use this information in later SQL statements, so for instance

```
SELECT * FROM USR1.TEST1
```

would fail with a syntax error.

To avoid this, the default behavior of the Easysoft ODBC-InterBase Driver is to not return this information from metadata calls (such as `SQLTables`) unless this attribute is selected.

Selecting this attribute will cause most applications to fail when used with the Easysoft ODBC-InterBase Driver.

DEADLOCKS

Instructs the Easysoft ODBC-InterBase Driver to report update conflicts.

Interface	**Value**
DSN Dialog Box (Windows)	No Wait
odbc.ini file (Unix)	Nowait=0|1
Connect String	NOWAIT=0|1

InterBase uses a transaction model that avoids deadlocks between conflicting updates from concurrent applications.

This is achieved by pausing the transaction within any secondary application until the active transaction within a primary application has finished and then executing that secondary transaction.

By default, the Easysoft ODBC-InterBase Driver will also act in this manner, but selecting this attribute will return an error message reporting the update conflict.

SQL DIALECT

The SQL dialect passed to the server when processing SQL queries.

Interface	**Value**
DSN Dialog Box (Windows)	Dialect
odbc.ini file (Unix)	Dialect=1|2|3
Connect String	DIALECT=1|2|3

For more details consult the InterBase documentation detailing the difference between dialects.

See **"Leave Quotes" on page 73** for details of how amending the value of this attribute may also affect the way in which the Easysoft ODBC-InterBase Driver converts certain SQL strings.

METADATA PRECISION

Indicates to the Easysoft ODBC-InterBase Driver not to use the precision field in metadata queries.

Interface	**Value**
DSN Dialog Box (Windows)	Pre IB6 MetaData
odbc.ini file (Unix)	OldMetaData=0|1
Connect String	OLDMETADATA=0|1

In current versions of InterBase, the metadata describing NUMERIC columns contains both a precision and scale value, so that, for instance, a column created as

```
NUMERIC (10, 2)
```

would have a precision of 10 and a scale of 2.

However, in versions of InterBase prior to version 6, this column would have been stored in a double precision field, which cannot supply a fixed precision value, so the metadata for this version contains a column length only (8 in the case of a double precision field), rather than the precision value.

The Easysoft ODBC-InterBase Driver will convert the length value into a precision value, but in this case it would not be a 10, but an 18, this being the maximum precision that could be stored in the field.

In most cases the Easysoft ODBC-InterBase Driver will attempt to determine if the metadata contains a precision value without help, but there are some InterBase version 6 databases that do not have the precision value.

Selecting this attribute indicates to the Easysoft ODBC-InterBase Driver not to use the precision field in metadata queries.

This attribute is cleared by default, which is correct in most cases, but it should be selected if problems are encountered with the `SQLColumns` ODBC API.

EXECUTE PROC

Whether stored procedures should be called by using a `Select` or an `Execute` statement. If you are calling an Executable procedure, enable `Execute Proc` (check the attribute or set it to 1). Otherwise, leave `Execute Proc` set its default value OFF (set to 0 or clear).

Interface	**Value**
DSN Dialog Box (Windows)	Execute Proc
odbc.ini file (Unix)	ExecProc=0|1
Connect String	EXECPROC=0|1

There are two types of InterBase procedure:

- Select procedures, which return a result set and are called from a `Select` statement.
- Executable procedures, which optionally return values in output parameters and are called with an `Execute Procedure` statement.

`Execute Proc` allows applications to call either type of procedure by using the standard ODBC escape sequence. For example:

```
{call myselectproc('myarg1', 'myarg2')}
```

```
{call myexecutableproc('myarg')}
```

When `Execute Proc` is disabled (the default), the Easysoft ODBC-InterBase Driver converts procedure calls to use the Select statement method. For example:

```
select * from myselectproc('myarg1', 'myarg2')
```

The called procedure then generates a result set that can be accessed in the same way as any other result set.

When `Execute Proc` is enabled, the Easysoft ODBC-InterBase Driver converts procedure calls to use the Execute Procedure method. For example:

```
execute procedure myexecutableproc('myarg')
```

The called procedure does not return a result set. It may return data if at least one of the procedure arguments has been defined as an output parameter into which values can be placed.

The Easysoft ODBC-InterBase Driver will generate a virtual result set consisting of a single row that contains any output values from the procedure. Unless the driver does this, applications that call `SQLFetch` following an executable procedure call can cause the InterBase server process to fail.

If an application uses a direct InterBase stored procedure call (rather than the ODBC escape sequence), the Easysoft ODBC-InterBase Driver will adapt to match that usage when `Execute Proc` is enabled.

LEAVE QUOTES

Instructs the Easysoft ODBC-InterBase Driver to strip double quotes from any SQL string if the SQL Dialect attribute (see **"SQL Dialect" on page 69**) is set to 1 or 2.

Interface	**Value**
DSN Dialog Box (Windows)	Leave Quotes
odbc.ini file (Unix)	DQuote=0|1
Connect String	DQUOTE=0|1

Although SQL92 makes use of both single and double quotes in SQL strings, they have different meanings.

Given the SQL statement

```
SELECT * from TABLE WHERE
"COLUMN NAME" = 'COLUMN VALUE'
```

the result set would contain all the rows where the column with a name of "COLUMN NAME" contains the value 'COLUMN VALUE'.

Double quotes are used to quote identifiers (in this case column names) and single quotes are used to create character literals (normally values with a type of `SQL_VARCHAR`) containing printable characters.

InterBase treats quotes in this way when using Dialect 3, but double and single quotes are treated the same manner when using Dialect 1 or Dialect 2, so that the expression

```
WHERE "COLUMN NAME" = 'COLUMN VALUE'
```

would be identical to

```
WHERE 'COLUMN NAME' = 'COLUMN VALUE'
```

which would always evaluate to false.

This can cause problems with applications that use quotes around column and table names, because even though the Easysoft ODBC-InterBase Driver is queried as to the character to use for quoted identifiers, at least one application (Microsoft Access) fails if the application is instructed not to use any character for this purpose.

To overcome this issue, if the SQL Dialect attribute is set to 1 or 2 then the Easysoft ODBC-InterBase Driver will strip double quotes from any SQL string by default.

This default operation may be disabled by selecting this attribute.

UID Override

Allows the user and password information passed in as part of the connect string in the `SQLDriverConnect` function and the user and password used in the `SQLConnect` function to be overridden by the values in the DSN or `.ini` file.

Interface	**Value**
DSN Dialog Box (Windows)	UID Override
odbc.ini file (Unix)	UIDOverride=0|1
Connect String	Not Used

This functionality is useful as some applications (e.g. Seagate Crystal Reports in server mode) will pass the Windows (rather than the InterBase) user name and password to the Easysoft ODBC-InterBase Driver.

Normal use would not require the selecting of this attribute.

READ ONLY

Restricts a database connection to read-only access.

Interface	**Value**
DSN Dialog Box (Windows)	Read Only
odbc.ini file (Unix)	ReadOnly=0|1
Connect String	READONLY=0|1

Only `Select` statements are permitted and no `Update`, `Insert` or `Delete` statements are allowed.

CHARACTER SET

Allows a default InterBase character set to be specified for the database connection.

Interface	Value
DSN Dialog Box (Windows)	Character Set
odbc.ini file (Unix)	Charset=value
Connect String	CHARSET=value

See the InterBase documentation for a list of valid character sets (such as `UNICODE_FSS`).

COMMIT SELECT

Enable `Commit Select` (check the attribute or set it to 1), if you are calling a Select procedure that executes an `Insert`, `Update` or `Delete` statement.

Interface	Value
DSN Dialog Box (Windows)	Commit Select
odbc.ini file (Unix)	CommitSelect=0|1
Connect String	COMMITSELECT=0|1

When `Commit Select` is disabled (the default), the Easysoft ODBC-InterBase Driver automatically commits `Insert`, `Update` or `Delete` statements when in auto-commit mode, but does not commit `Select` statements. InterBase Select procedures are called from a `Select` statement and can contain `Insert`, `Update` or `Delete` statements (or call other procedures that contain these statements). Enabling `Commit Select` forces the Easysoft ODBC-InterBase Driver to call a `Commit` statement after each call to a `Select` statement when auto-commit mode is on. This ensures that updates made by Select procedures are committed.

WITH DEFAULT

Returns the default column type from the `SQLColumns` statement.

Interface	**Value**
DSN Dialog Box (Windows)	With Default
odbc.ini file (Unix)	WithDefault=0|1
Connect String	WITHDEFAULT=0|1

However, this may not provide usable information, as the BLOB that comes back is the internal BLR representation of the default value.

TRANSACTION ISOLATION

Causes the database connection to use a different default isolation method for transactions.

Interface	**Value**
DSN Dialog Box (Windows)	Txn Isolation
odbc.ini file (Unix)	TxnMode=0|2|8
Connect String	TXNMODE=0|2|8

- 0 = SQL_TXN_REPEATABLE_READ

  the database default; in InterBase terms, either `isc_tpb_read` or `isc_tpb_write`, depending on mode

- 2 = SQL_TXN_READ_COMMITTED

  in InterBase terms, `isc_tpb_read_committed` and `isc_tpb_rec_version`

- 8 = SQL_TXN_SERIALIZABLE

  in InterBase terms, `isc_tpb_consistency`

Any non-zero value specified in this attribute will override any isolation method specified in the calling application.

NB The default value for the Easysoft ODBC-InterBase Driver is set to `2` (SQL_TXN_READ_COMMITTED). This is NOT the same as the database default, which is 0 (SQL_TXN_REPEATABLE_READ).

The default isolation method can also be set via the ODBC API, but this attribute provides a way of setting it without altering the calling client application.

FLUSH COMMIT

Use a `Commit` statement instead of a `Commit_Retaining` statement when writing to a database.

Interface	**Value**
DSN Dialog Box (Windows)	Flush Commit
odbc.ini file (Unix)	FlushCommit=0|1
Connect String	FLUSHCOMMIT=0|1

Forces a `Commit` after a `Select` statement if required (e.g. with a `Select` statement from a stored procedure that does a update).

Note that although this ensures that transactions are restarted, the visibility of altered rows in other connections can be affected.

PAD VARCHAR

Returns VARCHAR data in the same format as CHAR data (i.e. they are not right space truncated).

Interface	**Value**
DSN Dialog Box (Windows)	Pad Varchar
odbc.ini file (Unix)	PadVarchar=0|1
Connect String	PADVARCHAR=0|1

NULL SCHEMA

Controls the schema data returned from metadata calls.

Interface	**Value**
DSN Dialog Box (Windows)	Null Schema
odbc.ini file (Unix)	NullSchema=0|1
Connect String	NULLSCHEMA=0|1

This option alters the way the driver returns schema information in the metadata call, and is used in conjunction with the With Schema option.

The ODBC specification states that for ODBC drivers that support schema (owner) information for the tables, but the current database does not have that information, the Schema field in the ODBC metadata fields (SQLTables, SQLColumns, etc.) should contain a empty string. And for those drivers that do not support the use of a schema, a NULL should be returned in those fields.

This option allows the selection of which of these two types of result the application requires. If the Null Schema option is set, then a NULL is returned, otherwise an empty string.

This option is normally only required with a few application, using the Driver under Microsoft SQL Server as a “Linked Table” is one such application.

FIX PRECISION

Control the precision information that the driver returns for NUMERIC fields

Interface	**Value**
DSN Dialog Box (Windows)	Fix Precision
odbc.ini file (Unix)	FixPrecision=0|1
Connect String	FIXPRECISION=0|1

Under normal use, the InterBase Driver will return the precision and scale of fixed precision data types, for example, NUMERIC(10, 3).

However, while InterBase allows such a definition, it will actually allow larger numbers to be stored in the field, effectively ignoring the precision value.

Because of this, if a application used the above field, the driver will report a precision of 10 and a scale of 3, so the application will expect to only extract fields that conform to this specification. However in some cases, if the value in the database exceeds this precision, the application will report an error.

This option allows the behavior to be altered, such that the ODBC driver reports a precision that is the maximum possible precision that could be stored in the field, overriding the actual stored precision. This prevents applications reporting a error under these conditions.

SIMPLE UNICODE

Alters the method used to convert between UNICODE and ANSI data

Interface	**Value**
DSN Dialog Box (Windows)	Simple UNICODE
odbc.ini file (Unix)	SimpleUnicode=0|1
Connect String	SIMPLEUNICODE=0|1

Under normal conditions, the InterBase Driver uses the operating system provided conversion routine to change the InterBase representation of UNICODE (UTF8) to the ODBC UNICODE standard (UCS-16). However it is possible to store single character values in a InterBase database, that is not a valid UTF8 character sequence. In this case, the operating system routine will not be able to perform the conversion, and will not give the expected result.

This option prevents the driver from using the conversion provided by the Operating System, but instead to use a simple conversion that will allow non UTF-8 data to be correctly converted to UCS-16. This simple conversion cannot correctly convert true UTF8 data, so the use of this option is governed by the actual data stored.

WCHAR DEFAULT TO CHAR

Select the DEFAULT datatype for wide character conversion

Interface	**Value**
DSN Dialog Box (Windows)	Default SQLCHAR
odbc.ini file (Unix)	WCharDefaultC=0|1
Connect String	WCHARDEFAULTC=0|1

The ODBC specification allows applications to request that data is returned in a number of forms, and for each SQL datatype there is a DEFAULT type. The specification states, that for a wide character field, the default return type is a SQL_WCHAR, a UNICODE representation.

However, some Microsoft applications do not expect this. It seems that Microsoft Access has been created expecting the default return from a wide character field to be a SQL_CHAR, the ANSI representation.

Microsoft Access only operates in this way, if the field in question is in the primary key of the table, and the symptoms of this are the table opens, but displays #deleted for all the fields.

This option allows the driver to be configured to use the conversion that Microsoft Access expects. It should be remembered that this is enabling non-standard driver behavior, so this may cause problems with other application. If this is the case, create a separate DSN with this option set for use only with Access.

WCMB Mode

Select the code page to use when mapping a character string to a wide character (Unicode) string.

Interface	**Value**
DSN Dialog Box (Windows)	WCMB Mode
Connect String	WCMB=*value*

The Windows version of the Easysoft ODBC-InterBase Driver uses the `MultiByteToWideChar` and `WideCharToMultiByte` functions to map character strings to and from wide character strings. Use `WCMB Mode` to specify the code page to use when doing this conversion. Refer to the Windows documentation for `MultiByteToWideChar` and `WideCharToMultiByte` for the meanings of the possible `WCMB Mode` values.

Hide LONGVARCHAR

Report LONGVARCHARs as VARCHAR(32000).

Interface	**Value**
DSN Dialog Box (Windows)	Hide LONGVARCHAR
odbc.ini file (Unix)	HideVarchar=0|1
Connect String	HIDEVARCHAR=0|1

Enable this setting if you get "out of memory" errors when using an InterBase driver under SQL Server to work with LONGVARCHAR data.

TECHNICAL REFERENCE

A

Technical Reference for the Easysoft ODBC-InterBase Driver

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

Appendix Guide

- **ODBC Compliance**
- **Multiple transactions on a single connection**
- **InterBase data types**
- **Runtime SQL Dialect support**
- **Rollback Retaining**
- **Chili!soft**
- **unixODBC**

ODBC Compliance

The Easysoft ODBC-InterBase 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 104, 105 and 109 of Level 1 requirements
- The driver conforms to 202, 203, 205, 209 and 211 of Level 2 requirements
- The driver implements all ODBC 3 functions with the exception of:

  `SQLBrowseConnect`

  `SQLBulkOperations`

  `SQLSetPos`

Cursor support in the Easysoft ODBC-InterBase Driver is restricted to FORWARD_ONLY, as this is the only cursor type supported by InterBase. As a result, the Easysoft ODBC-InterBase Driver will work with ADO snapshots, but not with dynasets.

Multiple transactions on a single connection

Multiple transactions on a single connection are not supported by ODBC 3 and therefore cannot be supported by the Easysoft ODBC-InterBase Driver.

InterBase data types

The Easysoft ODBC-InterBase Driver supports the DATE, TIME, TIMESTAMP and INT64 IB 6 data types.

Runtime SQL Dialect support

The required SQL Dialect can be set for a connection, but cannot be set at runtime, as it is not a statement level property.

Easysoft are considering adding this property in a future release.

Rollback Retaining

The Easysoft ODBC-InterBase Driver will support the `isc_rollback_retaining` function if it is present in a `GDS32.DLL` library.

Chili!soft

The Easysoft ODBC-InterBase Driver can be configured to work with Sun Chili!Soft ASP.Specific installation information is provided on the **Easysoft web site**

unixODBC

The Easysoft ODBC-InterBase Driver does not require a driver manager on the client machine, although there are some advantages in using one, such as:

- being able to pick the ODBC driver from the DSN and have it loaded for you dynamically
- the ability to work with multiple ODBC drivers

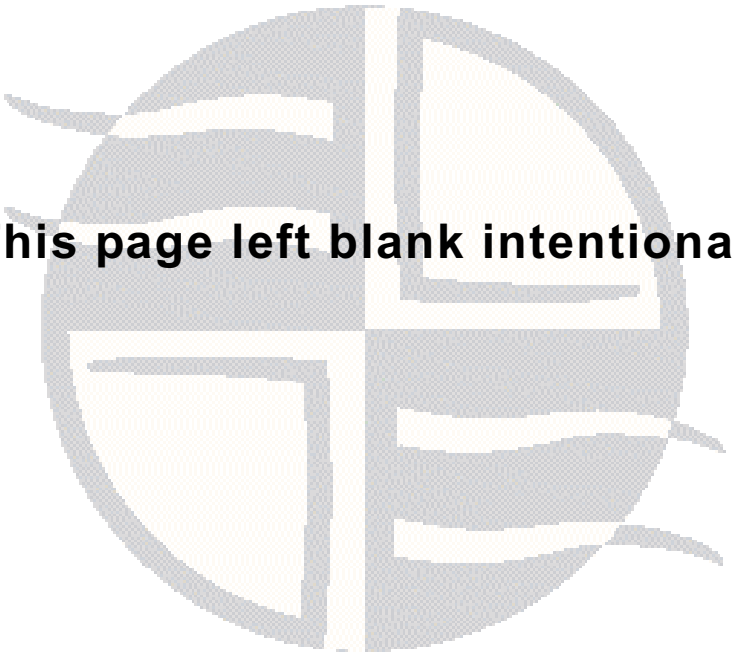
There are currently two open source driver managers available for Unix, unixODBC and iODBC. If you want to use a driver manager then Easysoft recommend the unixODBC driver manager.

There are a number of reasons for this:

1. The unixODBC project started by Peter Harvey is now maintained by Nick Gorham who is an Easysoft developer. This means there is much greater experience with unixODBC within Easysoft and we will be able to provide better support for the InterBase 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.
2. Easysoft test all their ODBC drivers with unixODBC and although they may work with iODBC it is more difficult for us to support it as we are less familiar with it.
3. 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 which may be used to explore your ODBC data. the InterBase Driver contains the code and shared object which is used by unixODBC's GUI ODBCConfig utility to add/delete and configure InterBase DSNs.
4. From 4.0.0, the InterBase Driver installation for UNIX can automatically install itself using the unixODBC installer program to run under unixODBC.

For the latest information about unixODBC, please refer to the unixODBC document in <*InstallDir*>`/easysoft/ib/doc` in case any of the information here has been superseded.

REF	For more information on unixODBC see **http://www.unixodbc.org**.



This page left blank intentionally

GLOSSARY

B

Terms and definitions

ADO (ActiveX Data Objects)

A Microsoft interface designed to provide a consistent method of accessing different data types, such as web pages, spreadsheets and relational databases.

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).

Database

A collection of data files.

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 93**) 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 94**.

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.

Remote Licensing

Easysoft products require licenses for each machine they are used. This may not be sutable in some installations. Because of this Easysoft can provide a remote license system, where licenses are applied to a central server, and the products use the license stored on this server. For more details, contact Easysoft

Row

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

Schema

A specification of the structure of a database, including the tables, their column headings and keys.

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 (SQL) - an international standard text language for querying and manipulating databases.

System data source

In the context of ODBC under Microsoft Windows, a data source which can be accessed by any user on a given system. Compare with **"User data source" on page 95**.

Table

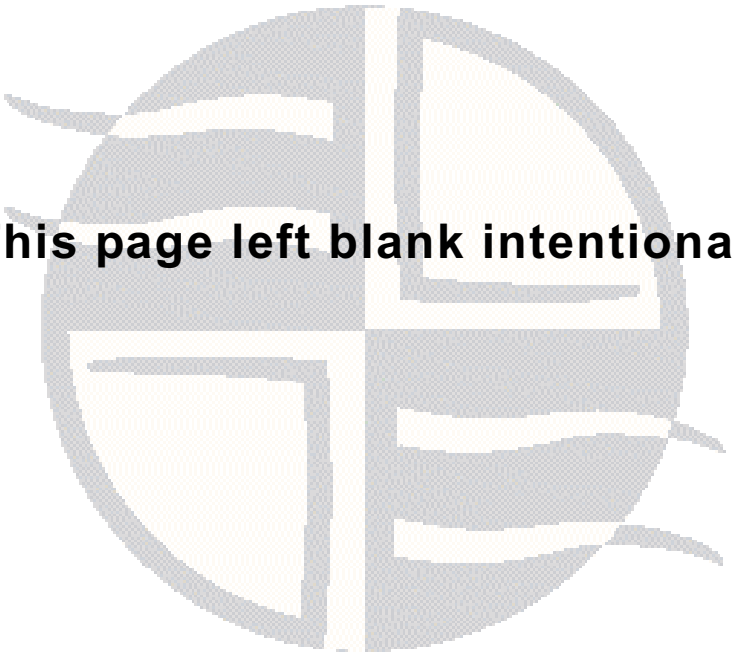
A data set in a relational database, composed of rows and columns.

TCP/IP

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

User data source

In the context of ODBC under Microsoft Windows, a data source which can only be accessed by a specific user on a given system. Compare with **"System data source" on page 95**.



This page left blank intentionally

INDEX

Symbols

/easysoft directory 37
/usr/local 36

A

ADO 14, 61, 86, 91
ANSI 15
API 91

B

beta releases 20
bunzip 33
bzip2 21

C

C runtime library
    version of 35
Caution box 9
CD 20
Character Set 76
Chili!soft 87
client
    setup 53
    setup on Windows 55
Commit Select 76
compress 21

Control Panel
    ODBC 55
create data source
    for client on Windows 55

D

data types
    IB 5 14
    IB 6 ..... 14, 87
database
    connection 63
    remote connections 65
    user privileges 65
Database Name 63
Deadlocks 69
documentation 20
driver manager
    unixODBC 38
DSN configuration dialog box 57
DSN Description 62
DSN Name 62
DSN setup 54

E

Execute Proc 71

F

Flush Commit 79
FORWARD_ONLY cursors 86
FTP 20

G

gds32.dll 23
glibc
    see C runtime library
gunzip 33
gzip 21

I

ibclient64.dll 23
install directory 36
installation
    requirements 32
    requirements for Unix 32
Installing on Windows 23

L

LD_LIBRARY_PATH 47, 48
LD_RUN_PATH 47, 48
Leave Quotes 73
libc6
    see C runtime library
license
    authorization code 25
license agreement 34

M

Metadata 80
Metadata Precision 70
Multiple transactions 86

N

Note box 9

O

ODBC Compliance 86
ODBC Data Source Administrator 55
odbc.ini 54, 60
odbcinst 40
odbcinst.ini 61

P

Pad Varchar 79, 80, 81, 82, 83
Password 67
patches 20
Platform note 9

R

Read Only 75
Reference box 9
Roles 15, 67
Rollback Retaining 87
rollback_retaining function 16

S

Schema Information 68
SHLIB_PATH 47, 48
SQL Dialect 16, 69, 87
stored procedures 17
symbolic link 38
System DSN tab 56

T

tar 33
Test button 58
Transaction Isolation 78
transactions 17

U

UCS 82
UID Override 74
uncompress 33
UNICODE 14, 15, 82
unixODBC
    why it should be used 87-89
upgrades 20
User DSN tab 55
User Name 66
UTF8 82

W

web site 20
With Default 77