

Easysoft ODBC- InterBase Driver User's Guide

This manual documents version 4.9.n of the Easysoft ODBC-InterBase Driver.

Copyright © 1993-2025 Easysoft Limited.

All rights reserved.

You may not reverse engineer, decompile, or disassemble this manual. Information in this document 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.

The software described in this document is provided under a [licence agreement](#) and may be used only in accordance with the terms of that agreement.

Table of contents

Getting started	4
Installing the Easysoft ODBC-InterBase Driver	5
Installing on Linux or UNIX	5
Uninstalling on Linux or UNIX	13
Installing on Windows	14
Uninstalling on Windows	16
Connecting to InterBase	17
Connecting from Linux or UNIX	17
Connecting from Windows	18
Connection attributes	19
DSN-less connections	29
Logging	30
ODBC Driver Manager logging on Linux or UNIX	30
Easysoft ODBC-InterBase Driver logging on Linux and UNIX	30
ODBC Driver Manager logging on Windows	30
Finding out what product version you have on Windows	33
Client applications.	34
Microsoft Access	35
Linking a table.	35
Importing a table	35
Microsoft Excel	36
Data Connection Wizard	36
Microsoft Query	36
PowerPivot	36
Microsoft Power BI	37
Microsoft SQL Server	38
Oracle	40
Connecting InterBase to Oracle on Windows	40
Connecting InterBase to Oracle on Linux and UNIX.	41
LibreOffice	44
Go	45
Node.js	46
Perl	48
PHP	51
Python	53
R	55
About the Easysoft ODBC-InterBase Driver	57
ODBC API and scalar functions.	58
API functions	58
Data type mapping	61
Finding out more about data types on Windows	61
Cursor support	62
Multiple transactions on a single connection	63
Runtime SQL dialect support.	64
Rollback retaining	65
SQL support.	66
Example queries	66
Example inserts, updates, and deletes	72
Index	75

Getting started

This section shows you how to install the Easysoft ODBC-InterBase Driver and configure the ODBC data source that stores the connection details for your InterBase database. You're then ready to work with InterBase data in your application.

Note	You need to install the InterBase client on the same computer as the Easysoft ODBC-InterBase Driver, otherwise the driver won't function. To install the InterBase client, choose the client installation option in InterBase setup.
-------------	--

- [Installing the Easysoft ODBC-InterBase Driver](#)
- [Connecting to InterBase](#)
- [Logging](#)

Installing the Easysoft ODBC-InterBase Driver

Install the Easysoft ODBC-InterBase Driver on the computer where the application you want to connect to InterBase is running.

- [Installing on Linux or UNIX](#)
- [Uninstalling on Linux or UNIX](#)
- [Installing on Windows](#)
- [Uninstalling on Windows](#)

Installing on Linux or UNIX

The installation can be done by anyone with root access.

1. [Download the Easysoft ODBC-InterBase Driver distribution for your client application platform.](#)

If your [client application is 64-bit](#), choose the 64-bit driver distribution from the **Platforms** list. If your [client application](#) is 32-bit, choose the 32-bit driver distribution from the **Platforms** list.

2. Copy the distribution to a temporary directory on the machine where the application you want to connect to InterBase is installed.
3. Unpack the distribution and cd into the resultant directory.
4. As root, run:

```
./install
```

5. Follow the onscreen instructions to progress through the installation.

Further information

- [Preinstallation requirements](#)
- [What you can install](#)
- [Where to install](#)
- [Changes made to your system](#)
- [Installing alongside other existing Easysoft product installations](#)
- [Gathering information required during the installation](#)
- [Unpacking the distribution](#)
- [License to use](#)
- [Answering questions during the installation](#)
- [Running the installer](#)
- [Locating or installing unixODBC](#)
- [Installing the Easysoft ODBC driver](#)
- [Licensing](#)
- [Post installation steps for non-root installations](#)

Preinstallation requirements

To install the Easysoft ODBC-InterBase Driver 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 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.

6 Installing on Linux or UNIX

- Depending on the platform, you'll need up to 10 MB of temporary space for the installation files and up to 10 MB of free disk space for the installed programs. If you also install the unixODBC Driver Manager, these numbers increase by approximately 1.5 MB.
- For Easysoft licensing to work, you must do one of the following:
 - Install the Easysoft ODBC-InterBase Driver in /usr/local/easysoft.
 - Install the Easysoft ODBC-InterBase 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-InterBase Driver elsewhere and set the EASYSOFT_ROOT environment variable. For more information about setting the EASYSOFT_ROOT environment variable, refer to [Post installation steps for non-root installations](#).
- An ODBC Driver Manager.

Easysoft ODBC-InterBase 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-InterBase 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.

We recommend that you install all components as the root user.

What you can install

This distribution contains:

- The Easysoft ODBC-InterBase Driver.
- The unixODBC Driver Manager.

You need an ODBC Driver Manager to use the Easysoft ODBC-InterBase Driver from your applications. The distribution therefore contains the unixODBC Driver Manager. Most (if not all) UNIX and Linux applications 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-InterBase 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 ODBC Driver Manager in the standard places. If you have installed it in a non-standard location, the installation script prompts 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're 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 tries 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. For information about how to correct this manually, refer to [Post installation steps for non-root installations](#).

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

The 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-InterBase 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.)
2. The installation script installs an example data source into unixODBC. This data source will be added to your `SYSTEM` `odbc.ini` file. You can locate your `SYSTEM` `odbc.ini` file by using `odbcinst -j`.
3. Dynamic linker. On operating systems where the dynamic linker has a file listing locations for shared objects (Linux and FreeBSD), 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`.
 - On FreeBSD this is usually the file `/etc/defaults/rc.conf`.

Installing alongside other existing Easysoft product installations

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 are 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 get 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're 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're 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.

Unpacking the distribution

The distribution for UNIX and Linux platforms is a tar file. To extract the installation files from the tar file, use:

```
tar -xvf odbc-interbase-4.3.0-linux-x86-glibc.tar
```

This creates 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 into the directory created by unpacking the tar file to run the installation script. For example:

```
# cd odbc-interbase-4.3.0-linux-x86-glibc
```

License to use

The end-user license agreement (EULA) is in the file license.txt. Be sure to understand the terms of the agreement before continuing, as you're required to accept the license terms at the start of the installation.

Answering questions during the installation

Throughout the installation, you're prompted to answer some questions. In each case, the default choice displays in square brackets and you need only press Enter to accept the default. If there are alternative responses, these are 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

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. We recommend installing as the root user. (If you're concerned about the changes that will be made to your system, refer to [Changes made to your system](#).)

To start the installation, run:

```
./install
```

You need to:

- Confirm your acceptance of the license agreement by typing "yes" or "no". For more information about the license agreement, refer to [License to use](#).
- Supply the location where the software is to be installed.

We recommend accepting the default installation path.

For more information, refer to [Where to install](#).

Locating or installing unixODBC

We strongly recommend you use the unixODBC Driver Manager because:

- The installation script is designed to work with unixODBC and can automatically add Easysoft ODBC-InterBase Driver and data sources during the installation.
- Most applications and interfaces that support ODBC are compatible with unixODBC. The Easysoft ODBC-InterBase Driver and any data sources that you add during the installation are 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-InterBase Driver running under unixODBC. It also means that if you find a problem in unixODBC, it's 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 displays:

```
Found unixODBC under path and it is version n.n.n
```

2. If the installation script can't 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 we provide.

The unixODBC in the Easysoft ODBC-InterBase 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 is /etc/odbc.ini.
--enable-drivers=no	This means other ODBC drivers that come with unixODBC are not installed.
--enable-iconv=no	This means unixODBC does not look for libiconv. Warnings about not finding an iconv library were confusing our customers.
--enable-stats=no	Turns off 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.

Option	Description
<code>--enable-readline=no</code>	This turns off readline support in isql. We did 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 turning on readline support makes isql unusable on these systems.
<code>--prefix=/usr/local/easysoft/unixODBC</code>	This installs unixODBC into /usr/local/easysoft/unixODBC.

Installing the Easysoft ODBC driver

The Easysoft ODBC-InterBase Driver installation script:

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

If the Easysoft ODBC-InterBase Driver is already registered with unixODBC, a warning displays that lists the drivers unixODBC knows about. If you're installing the Easysoft ODBC-InterBase Driver into a different directory than it was installed before, you need to edit your `odbcinst.ini` file after the installation and correct the Driver and Setup paths. unixODBC's `odbcinst` doesn't update these paths if a driver is already registered.

- Creates an example Easysoft ODBC-InterBase Driver data source. If unixODBC is installed and you registered the Easysoft ODBC-InterBase Driver with unixODBC, the installation script adds example data source to your `odbc.ini` file.

Licensing

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

Licenses are stored in `installation_path/easysoft/license/licenses`.

Important 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-InterBase Driver license:

```
Would you like to request a Easysoft ODBC-InterBase 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 `y`, the installation runs the `licshell` script.

To obtain a license automatically, you need to be connected to the Internet and allow outgoing connections to `license.easysoft.com` on port 8884. If you're not connected to the Internet or don't allow outgoing connections on port 8884, the License Client can create a license request file that you can email to us.

When you start the License Client, the following menu displays:

```
[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're installing. The License Client then runs a program that generates a key that's 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-InterBase 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.
- A reference number (also referred to as an authorization code). When applying for a trial license, press Enter when prompted for a reference number. This field only applies to full (paid) licenses.

You're then asked to choose 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
The license request is output to `license_request.txt`.
- [3] Cancel this operation

If you choose to obtain the license automatically, the License Client tries to open a TCP/IP connection to `license.easysoft.com` on port 8884 and send the details you supplied along with 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 send the the request to us. The License daemon returns the license key, prints 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 sent the license request to us, we'll return a license key. Add this to the end of the file `installation_path/easysoft/license/licenses`.

Post installation steps for non-root installations

If you installed the Easysoft ODBC-InterBase Driver as a non-root user (not recommended), there may be some additional steps you need to do manually:

1. If you attempt to install the Easysoft ODBC-InterBase Driver under the unixODBC Driver Manager and you do not have write permission to unixODBC's `odbcinst.ini` file, the driver can't be added.
You can manually install the driver under unixODBC by adding an entry to the `odbcinst.ini` file. Run `odbcinst -j` to find out the location of the `DRIVERS` file then append the lines from `drv_template` file to `odbcinst.ini`. (`drv_template` is in the directory where the Easysoft distribution was untarred to.)
2. No example data sources can be added into unixODBC if you do not have write permission to the `SYSTEM odbc.ini` file. Run `odbcinst -j` to find out the location of the `SYSTEM DATA SOURCES` file then add your data sources to this file.
3. On systems where the dynamic linker has a configuration file defining the locations where it looks for shared objects (Linux and FreeBSD), you need to add:

```
installation_path/easysoft/lib  
installation_path/easysoft/unixODBC/lib
```

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

4. If you didn't install the Easysoft ODBC-InterBase Driver in the default location, you need to do one of the following:

- Link `/usr/local/easysoft` to the easysoft directory in your chosen installation path.

For example, if you installed in `/home/user`, the installation creates `/home/user/easysoft` and you need to symbolically link `/usr/local/easysoft` to `/home/user/easysoft`:

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

- Set and export the `EASYSOFT_ROOT` environment variable to `installation_path/easysoft`.
5. If your system doesn't have a dynamic linker configuration file, you need to add the paths listed in step 3 to whatever environment path the dynamic linker uses to locate shared objects. You may want to add these paths to a system file run whenever someone logs. For example, `/etc/profile`.

The environment variable depends on the dynamic linker. Refer to your `ld` or `ld.so` man page. It is usually:

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

Uninstalling on Linux or UNIX

There is no automated way to remove the Easysoft ODBC-InterBase Driver in this release. However, removal is quite simple. To do this:

1. Change directory to *installation_path*/easysoft and delete the product directory. *installation_path* is the Easysoft ODBC-InterBase 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-InterBase Driver entry needs to be removed from the odbcinst.ini file. To check whether the Easysoft ODBC-InterBase Driver is configured under unixODBC, use odbcinst -q -d. If the command output contains [INTERBASE], uninstall the driver from unixODBC by using:

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

If a reduced usage count message is displayed, repeat this command until odbcinst reports that the driver has been removed.

1. If you created any Easysoft ODBC-InterBase Driver data sources under unixODBC, you may want to delete these. To do this, first use odbcinst -j to locate USER and SYSTEM odbc.ini files. Then check those files for data sources that have the driver attribute set to INTERBASE.
2. Remove the install.info for the Easysoft ODBC-InterBase Driver from the /usr/local/easysoft directory.

Installing on Windows

The Windows installation can be done by anyone with local administrator privileges.

1. [Download the Easysoft ODBC-InterBase Driver installer.](#)
2. Follow the onscreen instructions to progress through the installation wizard.

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. The 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 **Restart Manager** to try to stop and restart each application or service in the list. If possible, **Restart Manager** restores applications to the same state that they were in before it shut them down.

Licensing

By default, the installer starts the Easysoft License Manager, because you can't use the Easysoft ODBC-InterBase Driver until you have a license. If you choose not to run Easysoft License Manager as part of the installation process, run License Manager from the **Easysoft** group in the Windows **Start** menu when you're ready to license the Easysoft ODBC-InterBase Driver. These 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.

To license the Easysoft ODBC-InterBase Driver:

1. In License Manager, enter your contact details.

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

The e-mail address **must** be the same as the one used to register at the Easysoft web site. Otherwise, you won't be able to obtain a trial license.

2. Choose **Request License**.

You're prompted to choose a license type.

3. Do one of the following:

- For a trial license, choose **Time Limited Trial**, and then choose **Next**.

-Or-

- For a purchased license, choose **Non-expiring License**, and then choose **Next**.

4. Choose your product from the drop-down list when prompted, and then choose **Next**.

5. For a purchased license, enter your authorization code when prompted, and then choose **Next**.

6. Choose how to get your license when prompted.

7. Do one of the following:

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

With this method, License Manager automatically requests and then applies your license.

-Or-

- Choose **View Request**. Then open a web browser and go to https://www.easysoft.com/support/licensing/trial_license.html or https://www.easysoft.com/support/licensing/full_license.html, as appropriate. In the web page, enter your machine number (labelled **Number** in the license request). For purchased licenses, you also need to enter your authorization code (labelled **Ref** in the license request).

We'll automatically email your license to the email address you supplied in License Manager.

-Or-

- Choose **Email Request** to email your license request to our licensing team. Once we've processed your request, we'll email your license to the email address you supplied in License Manager.

8. Close the License Manager windows and then choose **Finish**.

If you chose either **View Request** or **Email Request**, apply your license by double-clicking the email attachment when you get the license email from us. Alternatively, start License Manager from the **Easysoft** folder in the Windows **Start** menu. Then choose **Enter License** and paste the license in the space provided.

Once you've licensed the Easysoft ODBC-InterBase Driver, the installation is complete.

Repairing the 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. To do this:

1. In the Windows **Taskbar**, enter Add or remove programs in the Windows **Search** box.
2. Select Easysoft ODBC-InterBase Driver in the list, and then choose **Repair**.

Uninstalling on Windows

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

Removing Easysoft ODBC-InterBase Driver data sources

Easysoft ODBC-InterBase Driver data sources are not removed when you uninstall the Easysoft ODBC-InterBase Driver. You don't therefore need to recreate your Easysoft ODBC-InterBase Driver data sources if you reinstall or upgrade. If you don't want to keep your Easysoft ODBC-InterBase Driver data sources, use Microsoft **ODBC Data Source Administrator** to remove them, **before** uninstalling the Easysoft ODBC-InterBase Driver:

1. In the Windows **Taskbar**, enter Run in the Windows **Search** box.
2. In the Windows **Run** dialog box, enter:

```
odbcad32.exe
```

3. Locate your data source in either the **User** or **System** tab.
4. Select the data source from the list, and then choose **Remove**.

If the **Remove** button isn't available, close **ODBC Data Source Administrator**, and then, in the Windows **Run** dialog box, enter:

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

Repeat the previous two steps.

Removing the Easysoft ODBC-InterBase Driver

1. In the Windows **Taskbar**, enter Add or remove programs in the Windows **Search** box.
2. Select Easysoft ODBC-InterBase Driver in the list, and then choose **Uninstall**.

Note

Easysoft product licenses are stored 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.

Connecting to InterBase

Applications that support ODBC interface with an ODBC Driver Manager, which is included with the operating system, and also the Easysoft ODBC driver distribution on some platforms. One of the jobs that the ODBC Driver Manager does is to manage ODBC data sources. A data source specifies which ODBC driver to load, which data store to connect to, and how to connect to it.

Before setting up a data source, you must have [successfully installed the Easysoft ODBC-InterBase Driver](#).

- [Connecting from Linux or UNIX](#)
- [Connecting from Windows](#)

Connecting from Linux or UNIX

Creating an ODBC data source

There are two ways to create a data source to your InterBase data:

- Create a SYSTEM data source, which is available to anyone who logs on to the computer where the Easysoft ODBC-InterBase Driver is installed.
– Or –
- Create a USER data source, which is only available to the user who is currently logged on to the computer where the Easysoft ODBC-InterBase Driver is installed.

By default, the Easysoft ODBC-InterBase Driver installation creates a sample SYSTEM data source named IB_SAMPLE. If you're using the unixODBC included in the Easysoft ODBC-InterBase 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 InterBase, USER data sources must be created and edited in `$HOME/.odbc.ini`.

Notes

- 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.

The Driver attribute identifies the ODBC driver in the `odbcinst.ini` file to use for a data source. When the Easysoft ODBC-InterBase Driver is installed into unixODBC, it places a INTERBASE entry into the `odbcinst.ini` file. You should always have `Driver = INTERBASE` in your Easysoft ODBC-InterBase Driver data sources therefore.

To configure a Easysoft ODBC-InterBase Driver data source, in your `odbc.ini` file, you need to specify:

- The database to connect to (Database).
- A user who has permission to access this database (User).
- The password for this user (Password).

For example:

```
[IB_SAMPLE]
Driver          = INTERBASE
Database        = /opt/interbase/examples/employee.gdb
User            = SYSDBA
Password        = masterkey
```

The Easysoft ODBC-InterBase Driver must be able to find the following shared objects:

- `libodbcinst.so`
By default, this is located in `/usr/local/easysoft/unixODBC/lib/`.
- `libeslicshr.so`
By default, this is located in `/usr/local/easysoft/lib/`.
- `libessupp.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.

The `isql` query tool lets you test your Easysoft ODBC-InterBase Driver data sources. To test the Easysoft ODBC-InterBase Driver connection:

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

```
SQL> SELECT * FROM Employees;
```

–Or–

4. Enter help to return a list of tables:

```
SQL> help
```

Connecting from Windows

Creating an ODBC data source

1. In the Windows **Taskbar Search** box, enter “Run”.
2. Do one of the following:
 - If your application is 64-bit, in the **Run** dialog box, enter:

```
odbcad32.exe
```

–Or–

- If your application is 32-bit, in the **Run** dialog box, enter:

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

Note

If you're not sure whether your application is 32-bit or 64-bit, start your application, then in Windows **Task Manager** check whether your application's process name contains (32-bit). For example, the process name for the 32-bit version of Excel is Microsoft Excel (32-bit); the process name for the 64-bit

version of Excel is Microsoft Excel. On older versions of Windows, 32-bit applications contain *32 in the process name rather than (32-bit). For applications such as Oracle or SQL Server that run as a service, check the *Background processes* list rather than the **Apps** list in **Task Manager**. If you're running a programming language from within a Windows command-line shell (for example, Command or PowerShell), in your shell, run the .exe file for the programming language. For example, run perl, php, python, or node. In **Task Manager**, expand the process list for **Windows Command Processor** or **Windows PowerShell**, as appropriate, and check whether the process for your programming language contains (32-bit).

3. Do one of the following:
 - To create a data source that only the user you're currently logged in as can access, choose the **User** tab.
If your application is a Windows service (for example, SQL Server or Oracle) creating a user data source won't work, unless the service is running as the same user you're logged in as.
 - To create a data source that all users on this computer can access, choose the **System** tab.
4. Choose **Add**.
5. In the list of ODBC drivers, select Easysoft ODBC-InterBase Driver, and then choose **Finish**.
6. Complete the Easysoft ODBC-InterBase Driver configuration dialog box.
To find out how to do this, refer to the Connection attributes section.
7. To test the connection to InterBase, choose **Test**.
Note that this doesn't test that the Easysoft ODBC-InterBase Driver is licensed. If you haven't yet [licensed](#) the Easysoft ODBC-InterBase Driver, this ODBC data source won't work with your application, even if the **Test** button succeeds.

Connection attributes

- [Setting on Linux and UNIX](#)
- [Setting on Windows](#)

Setting on Linux and UNIX

Your Easysoft ODBC-InterBase Driver data source in odbc.ini must contain these attributes:

- Database
- User
- Password

For example:

```
[IB_SAMPLE]
Driver       = INTERBASE
Database     = /opt/interbase/examples/employee.gdb
User        = SYSDBA
Password     = masterkey
```

These optional attributes may be set in odbc.ini.

- Charset
- CommitSelect
- Description
- Dialect
- DQuote
- ExecProc

- FixPrecision
- FlushCommit
- HideVarchar
- Nowait
- NullSchema
- OldMetaData
- PadVarchar
- ReadOnly
- Role
- SimpleUnicode
- TxnMode
- Uidoverride
- WCharDefaultC
- WCMB
- With_Schema
- WithDefault

For more information about these attributes, refer to the table in the next section.

Setting on Windows

The Easysoft ODBC-InterBase Driver data source configuration dialog box, accessible when you create or edit an Easysoft ODBC-InterBase Driver data source in **ODBC Data Source Administrator** contains these fields:

Name	Value
DSN	The name of the data source. You'll need to specify this in your application. For example, your application may prompt you to choose this from a list of DSNs.
Description	Some applications display this to help users identify a particular data source.

Name	Value
Database	<p>The name of the InterBase database to connect to.</p> <p>For example:</p> <p>C:\Users\All Users\Embarcadero\InterBase\gds_db\examples\database\employee.g db connects to a database on the same Windows computer as the Easysoft ODBC-InterBase Driver.</p> <p>-Or-</p> <p>/opt/interbase/examples/employee.gdb connects to a database on the same UNIX computer as the Easysoft ODBC-InterBase Driver.</p> <p>As this type of local connection is made in the process context of the client application, it can fail if the application user doesn't have permission to access the database. (For example, if the database connection is being made by a web server.)</p> <p>The following example value connects to a database on a different Windows computer to the Easysoft ODBC-InterBase Driver:</p> <p>192.168.0.10:C:\Users\All Users\Embarcadero\InterBase\gds_db\examples\database\employee.g db.</p> <p>In this type of connection, the client process uses TCP/IP to connect to a server process, which then connects to the server database. This avoids many of the problems associated with permissions that affect local database connections. The InterBase server may be located either on a remote computer, or on the same computer as the client.</p> <p>The following example connects by using Named Pipes and NetBIOS:</p> <p>\\192.168.0.11\C:\Users\All Users\Embarcadero\InterBase\gds_db\examples\database\employee.g db</p>
User Name	The InterBase user name, if required.
Password	The password for User Name.
Role	The InterBase role for this user.

Name	Value
With Schema	<p>Whether owner and schema information should be returned from metadata calls.</p> <p>The InterBase metadata store contains a value for the owner or schema of a table.</p> <p>So, for example, if a user named myuser ran:</p> <pre>CREATE TABLE TEST1 (I INTEGER NOT NULL PRIMARY KEY)</pre> <p>a table called TEST1 would be created and its schema would be set to myuser.</p> <p>However, InterBase doesn't support using this information in SQL statements, so for instance:</p> <pre>SELECT * FROM MYUSER.TEST1</pre> <p>fails with a syntax error.</p> <p>To avoid these errors, the Easysoft ODBC-InterBase Driver doesn't return schema information from metadata calls (for example SQLTables).</p> <p>Turning on this attribute will cause most applications to fail.</p>
No Wait	<p>Whether the Easysoft ODBC-InterBase Driver reports update conflicts.</p> <p>InterBase uses a transaction model that avoids deadlocks between conflicting updates from concurrent applications.</p> <p>This is done by pausing the transaction within a secondary application until the active transaction within a primary application has finished and then executing that secondary transaction.</p> <p>By default, the Easysoft ODBC-InterBase Driver also acts in this manner. Turning on this attribute returns an error message reporting the update conflict.</p>
Dialect	<p>The SQL dialect passed to the server when processing SQL queries.</p> <p>Refer to the InterBase documentation for the difference between dialects.</p> <p>Refer to the Leave Quotes attribute for information about how changing the Dialect attribute's value may affect the way that InterBase converts certain SQL strings.</p>

Name	Value
Pre IB6 MetaData	<p data-bbox="454 163 1490 239">Whether the Easysoft ODBC-InterBase Driver use the precision field in metadata queries.</p> <p data-bbox="454 280 1490 432">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) has a precision of 10 and a scale of 2.</p> <p data-bbox="454 472 1490 624">However, in earlier versions of InterBase, 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.</p> <p data-bbox="454 665 1490 781">The Easysoft ODBC-InterBase Driver converts the length value into a precision value, but in this case it would not be 10, but 18, this being the maximum precision that could be stored in the field.</p> <p data-bbox="454 822 1490 938">In most cases, the Easysoft ODBC-InterBase Driver can, by default, determine whether the metadata contains a precision value. However, some earlier versions of InterBase don't have the precision value.</p> <p data-bbox="454 978 1490 1055">If you turn on this attribute, the Easysoft ODBC-InterBase Driver doesn't use the precision field in metadata queries.</p> <p data-bbox="454 1095 1490 1202">This attribute is turned off by default, which is correct in most cases. Turn on the attribute, if you have problems with the SQLColumns ODBC API.</p>

Name	Value
Execute Proc	<p>Whether to call stored procedures with a SELECT or an EXECUTE statement.</p> <p>Turn on this attribute if you're calling an executable procedure. Otherwise, leave this attribute set to its default value (turned off).</p> <p>There are two types of InterBase procedure:</p> <ul style="list-style-type: none"> • 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. <p>Execute Proc lets applications call either type of procedure by using the standard ODBC escape sequence. For example:</p> <pre>{call myselectproc('myarg1', 'myarg2')}`</pre> <p>-Or-</p> <pre>{call myexecutableproc('myarg')}</pre> <p>When Execute Proc is turned off (the default state), the Easysoft ODBC-InterBase Driver converts procedure calls to use the SELECT statement method. For example:</p> <pre>SELECT * FROM myselectproc('myarg1', 'myarg2')`</pre> <p>The called procedure then generates a result set that can be accessed in the same way as any other result set.</p> <p>When Execute Proc is turned on, the Easysoft ODBC-InterBase Driver converts procedure calls to use the EXECUTE PROCEDURE method. For example:</p> <pre>EXECUTE PROCEDURE myexecutableproc('myarg')`</pre> <p>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.</p> <p>The Easysoft ODBC-InterBase Driver 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.</p> <p>If an application uses a direct InterBase stored procedure call (rather than the ODBC escape sequence), the Easysoft ODBC-InterBase Driver adapts to match that usage when Execute Proc is turned on.</p>

Name	Value
Leave Quotes	<p>Whether the Easysoft ODBC-InterBase Driver strip double quotes from SQL strings if SQL Dialect is set to 1 or 2.</p> <p>Although SQL-92 allows both single and double quotes in SQL strings, they have different meanings.</p> <p>In this SQL statement:</p> <pre>SELECT * FROM mytable WHERE "mycolumn" = 'myvalue'</pre> <p>the result set contains all the rows where a column with the name mycolumn contains the value myvalue.</p> <p>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.</p> <p>InterBase treats quotes in this way when using InterBase dialect 3. Double and single quotes are as being the same when using dialect 1 or dialect 2, so that the expression:</p> <pre>WHERE "mycolumn" = 'myvalue'</pre> <p>is identical to:</p> <pre>WHERE 'COLUMN NAME' = 'COLUMN VALUE'</pre> <p>which always evaluates to false.</p> <p>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.</p> <p>To work around this, if SQL Dialect is set to 1 or 2 then the Easysoft ODBC-InterBase Driver strip double quotes from any SQL string by default.</p> <p>This default behaviour can be turned off by turning on Leave Quotes.</p>
UID Override	<p>Overrides the user and password passed to either SQLDriverConnect or SQLConnect with values stored in the ODBC data source.</p> <p>You may need to turn on this attribute if using Crystal Reports.</p>
Read Only	<p>Restricts a database connection to read-only access.</p> <p>If you turn on this attribute, only SELECT statements are permitted. No UPDATE, INSERT, or DELETE statements are permitted.</p>
Character Set	<p>The default InterBase character to use for the connection.</p> <p>Refer to the InterBase documentation for a list of valid character sets. For example UNICODE_FSS.</p>

Name	Value
Commit Select	<p>Turn on this attribute if you're calling a SELECT procedure that executes an INSERT, UPDATE, or DELETE statement.</p> <p>When Commit Select is off (the default state), the Easysoft ODBC-InterBase Driver automatically commits INSERT, UPDATE, or DELETE statements when in autocommit 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). Turning on Commit Select forces the Easysoft ODBC-InterBase Driver to call a COMMIT statement after each SELECT statement call when autocommit mode is on. This ensures that updates made by SELECT procedures are committed.</p>
With Default	<p>Returns the default column type from the SQLColumns statement.</p> <p>However, this may not provide usable information, as the BLOB that comes back is the internal BLR representation of the default value.</p>
Txn Isolation	<p>Causes the database connection to use a different default isolation method for transactions.</p> <p>0 = SQL_TXN_REPEATABLE_READ</p> <p>The database default; in InterBase terms, either isc_tpb_read or isc_tpb_write, depending on the mode.</p> <p>2 = SQL_TXN_READ_COMMITTED</p> <p>In InterBase terms, isc_tpb_read_committed and isc_tpb_rec_version.</p> <p>8 = SQL_TXN_SERIALIZABLE</p> <p>In InterBase terms, isc_tpb_consistency.</p> <p>Any non-zero value specified with this attribute overrides any isolation method specified in the calling application.</p> <div data-bbox="451 1406 1495 1619"> <p>Note The default value for the Easysoft ODBC-InterBase Driver is 2 (SQL_TXN_READ_COMMITTED). This is not the same as the database default, which is 0 (SQL_TXN_REPEATABLE_READ).</p> </div> <p>The default isolation method can also be set by using the ODBC API. This attribute provides a way of setting it without altering the calling client application.</p>
Flush Commit	<p>Whether to use a COMMIT statement instead of a COMMIT RETAIN statement when writing to a database.</p> <p>Turning on this attribute forces a COMMIT after a SELECT statement if required. (For example, a SELECT statement from a stored procedure that does an update.) Note that although this ensures that transactions are restarted, the visibility of altered rows in other connections can be affected.</p>

Name	Value
Pad Varchar	Return VARCHAR columns in the same format as CHAR columns (namely, they are not right space truncated).
Null Schema	<p>This option alters the way the Easysoft ODBC-InterBase Driver returns schema information in metadata calls.</p> <p>According to the ODBC specification:</p> <p>For ODBC drivers that support schema (owner) information for tables, the Schema field in ODBC metadata calls (SQLTables, SQLColumns, and so on) should contain a empty string, if the database does not provide schema information.</p> <p>-Or-</p> <p>For ODBC drivers that do not support the use of a schema, a NULL should be returned in those fields.</p> <p>This option lets you choose which of these two ODBC driver behaviours your application requires. If Null Schema option is turned on, a NULL is returned, otherwise an empty string is returned.</p> <p>You may need to turn this option on if you're using the Easysoft ODBC-InterBase Driver with a SQL Server linked server.</p>
Fix Precision	<p>Control the precision information that the Easysoft ODBC-InterBase Driver returns for NUMERIC fields.</p> <p>By default, the Easysoft ODBC-InterBase Driver returns the precision and scale of fixed precision data types. For example, NUMERIC(10, 3).</p> <p>However, while InterBase supports such a definition, it actually allows larger numbers to be stored in the field, effectively ignoring the precision value. In some cases, if the value in the database exceeds the precision reported by the Easysoft ODBC-InterBase Driver, the application fails with an error. As a workaround for this, if you turn on this option, the Easysoft ODBC-InterBase Driver reports a precision that's the maximum possible precision that could be stored in the field, overriding the actual stored precision.</p>

Name	Value
Simple UNICODE	<p>Alters the method used to convert between Unicode and ANSI data.</p> <p>By default, the Easysoft ODBC-InterBase Driver uses the operating system conversion routine to change the InterBase Unicode encoding (UTF8) to the ODBC Unicode encoding (UCS-2). However, it is possible to store single character values in a InterBase database, that are not a valid UTF8 character sequence. In this case, the operating system routine can't do the conversion, and doesn't give the expected result.</p> <p>If you turn on this option, the Easysoft ODBC-InterBase Driver uses a built-in conversion mechanism that allows non UTF-8 data to be correctly converted to UCS-2.</p> <p>This simple conversion mechanism can't correctly convert true UTF8 data, so this option must be used on a case by case basis.</p>
Default SQLCHAR	<p>Select the DEFAULT data type for wide character conversion.</p> <p>The ODBC specification allows applications to request that data is returned in a number of forms, and for each SQL data type 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.</p> <p>However, some Microsoft applications don't 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.</p> <p>Microsoft Access only behaves 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.</p> <p>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 applications. If this is the case, create a separate ODBC data source with this option set for use only with Access.</p>
WCMB Mode	<p>The code page to use when mapping a character string to a wide character (Unicode) string.</p> <p>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.</p>
Hide LONGVARCHAR	<p>Report LONGVARCHAR columns as VARCHAR(32000). Turn on this option, if you get "out of memory" errors when using SQL Server and the Easysoft ODBC-InterBase Driver to work with LONGVARCHAR data.</p>

DSN-less connections

Some applications allow you to make an ODBC connection without configuring a data source. To do this, you supply a connection string that contains the ODBC driver name and other driver-specific attribute-value pairs.

Here's an example Easysoft ODBC-InterBase Driver connection string:

```
Driver={Easysoft Interbase ODBC};DATABASE=C:\\Users\\All  
Users\\Embarcadero\\InterBase\\gds_db\\examples\\database\\employee.gdb;UID=SYSDBA;  
PWD=masterkey;
```

Logging

If you report an issue to us, we may ask you to turn on ODBC Driver Manager or Easysoft ODBC-InterBase Driver logging, to help us diagnose the cause of the issue.

To turn on logging, refer to the following sections.

Note If your application is a service (for example, Oracle or SQL Server), you may need to restart the service before enabling logging takes effect. To do this on Linux or UNIX, use `service`, `systemctl`, or a vendor-supplied script. To do this on Windows, use the Windows **Services** app.

ODBC Driver Manager logging on Linux or UNIX

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

```
Trace = Yes
TraceFile = /path/filename
```

For example:

```
[ODBC]
Trace = Yes
TraceFile = /tmp/sql.log
```

Ensure that the user who's running the application to log has write permission to `TraceFile` (and to the directory containing it), otherwise no logging information will be produced.

Easysoft ODBC-InterBase Driver logging on Linux and UNIX

Driver manager trace files show all the ODBC calls an application makes, including their arguments and return values. Easysoft ODBC-InterBase Driver logging is specific to the Easysoft driver and is of most use when making a support call.

To turn on Easysoft ODBC-InterBase Driver logging, edit your ODBC data source in `odbc.ini`. For example:

```
[IB_SAMPLE]
.
.
Logging = Yes
LogFile = /tmp/easysoft-odbc-driver.log
```

The value shown in the example specifies a log file named `/tmp/easysoft-odbc-driver.log`. Ensure that the user who's running the application to log has write permission to the log file (and to the directory containing it), otherwise no logging information will be produced.

ODBC Driver Manager logging on Windows

1. In the Windows **Taskbar Search** box, enter "Run".
2. Do one of the following:
 - If your application is 64-bit, in the **Run** dialog box, enter:

```
odbcad32.exe
```

-Or-

- If your application is 32-bit, in the **Run** dialog box, enter:

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

Note

If you're not sure whether your application is 32-bit or 64-bit, start your application, then in Windows **Task Manager** check whether your application's process name contains (32-bit). For example, the process name for the 32-bit version of Excel is Microsoft Excel (32-bit); the process name for the 64-bit version of Excel is Microsoft Excel. On older versions of Windows, 32-bit applications contain *32 in the process name rather than (32-bit). For applications such as Oracle or SQL Server that run as a service, check the *Background processes* list rather than the **Apps** list in **Task Manager**. If you're running a programming language from within a Windows command-line shell (for example, Command or PowerShell), in your shell, run the .exe file for the programming language. For example, run perl, php, python, or node. In **Task Manager**, expand the process list for **Windows Command Processor** or **Windows PowerShell**, as appropriate, and check whether the process for your programming language contains (32-bit).

3. Choose the **Tracing** tab.
4. Select **Machine-Wide tracing for all identities**.
5. Enter a log file name and path in the space provided. For example:

```
C:\Windows\Temp\SQL.log
```

6. Choose **Start Tracing Now**.

Note

With SQL Server, you may get two Driver Manager log files, we need both. The first log file is in the folder that you specify in **ODBC Data Source Administrator**. The second file's location is defined by SQL Server. Two possible locations are the top-level folder (for example, C:\SQL.log) or the SQL Server temporary folder (for example, C:\Users\MSSQL\$SQLEXPRESS\AppData\Local\Temp\SQL.log). If the Driver Manager log file isn't in these folders, search for it on the drive where SQL Server is installed.

Finding out what product version you have on Windows

If you have an issue with the Easysoft ODBC-InterBase Driver, we may ask you to tell us what your product version is. To find this out:

1. In the Windows **Taskbar**, enter “Add or remove programs” in the Windows **Search** box.
2. Select Easysoft ODBC-InterBase Driver in the list.

The product version displays below.

Client applications

How to work with InterBase data in some example applications and programming languages:

- [Microsoft Access](#)
- [Microsoft Excel](#)
- [Microsoft Power BI](#)
- [SQL Server](#)
- [Oracle](#)
- [LibreOffice](#)
- [Go](#)
- [Node.js](#)
- [Perl](#)
- [PHP](#)
- [Python](#)
- [R](#)

Microsoft Access

1. [Install the Easysoft ODBC-InterBase Driver](#) on same computer as Access.
2. [Configure an ODBC data source](#).
3. Choose one of the following ways to work with your InterBase data in Access.

Linking a table

1. Open your Microsoft Access database.
2. Choose **External Data**.
3. In the **New Data Source** list, choose **From Other Sources > ODBC Database**.
4. In the **Get External Data** screen, choose **Link to the data source by creating a linked table**, and choose **OK**.
5. In the **Select Data Source** dialog box, choose the **Machine Data Source** tab.
6. Choose your Easysoft ODBC-InterBase Driver ODBC data source from the **Machine Data Source** list, and then choose **OK**.
7. In the **Link Tables** dialog box, choose the tables that you want to link to, and then choose **OK**.

Importing a table

1. Open your Microsoft Access database.
2. Choose **External Data**.
3. In the **New Data Source** list, choose **From Other Sources > ODBC Database**.
4. In the **Get External Data** screen, choose **Import the source data into a new table in the current database**, and choose **OK**.
5. In the **Select Data Source** dialog box, choose the **Machine Data Source** tab.
6. Choose your Easysoft ODBC-InterBase Driver ODBC data source from the **Machine Data Source** list, and then choose **OK**.
7. In the **Import Objects** dialog box, choose the tables you want to import, and then choose **OK**.

Microsoft Excel

1. [Install the Easysoft ODBC-InterBase Driver](#) on same computer as Excel.
2. [Configure an ODBC data source](#).
3. Choose one of the following ways to work with your InterBase data in Excel.

Data Connection Wizard

1. Choose **Data > Get Data > From Other Sources > From ODBC**.
2. Choose your Easysoft ODBC-InterBase Driver data source from the list, and then choose **OK**.
3. Enter the user name and password for your data store if applicable, otherwise, enter any text string to get past this stage. Choose **Next**.
4. Choose the table that contains the data you want to retrieve, and then choose **Load**.

Microsoft Query

1. Choose **Data > Get Data > From Other Sources > From Microsoft Query**.
2. In the **Choose Data Source** dialog box, choose your InterBase data source from the list, and then choose **OK**.
3. In the **Query Wizard**, choose the columns that contain the data you want to retrieve, and then click **Next**.
4. If you want to return a subset of the data, use the **Filter Data** screen to filter the results of your query (this is the equivalent of a SQL WHERE clause), and then choose **Next**.
5. If you want to change the sort order of your data, use the **Sort Order** screen to sort the results of your query (this is the equivalent of a SQL ORDER BY clause), and then choose **Next**. Choose **Finish** to return your InterBase data to Excel.

PowerPivot

1. On the **PowerPivot** tab, choose **Manage**.
2. In the **PowerPivot** window, choose **Get External Data > From Other Sources**.
3. In the **Connect to a Data Source** list, choose **Others (OLEDB/ODBC)**.
4. In the **Specify a Connection** screen, enter a name for your connection in the space provided. Then choose **Build**.
5. In the **Data Link Properties** box, choose your Easysoft ODBC-InterBase Driver data source from the list, and then choose **OK**.
6. Choose **Next**.
7. Choose how to import your InterBase data and then choose **Finish**.
8. Choose **Close** to return the data to Excel.

Microsoft Power BI

1. [Install the Easysoft ODBC-InterBase Driver](#) on same computer as Power BI Desktop.
2. [Configure an ODBC data source](#).
3. In Power BI Desktop, choose **Get data from another source**.
4. In the **Get Data** dialog box, choose **ODBC**, and then choose **Connect**.
5. In the **From ODBC** dialog box, choose your InterBase data source, and then choose **OK**.
6. Enter your database user name and password when prompted.

If you make a mistake when entering the user name and password, cancel the connection process. Then in Power BI Desktop **Options and Settings**, edit the data source. Specify the correct user name or password in the data source credentials dialog box. Otherwise, Power BI Desktop will continue to use the cached incorrect credentials.

Note	If you do not normally need to enter a user name and password, enter some dummy strings in the spaces provided.
-------------	---

7. In the **Navigator** dialog box, choose the tables you want to analyse in Power BI Desktop, and then choose **Load**.

Your InterBase data is now available to use in Power BI visualisations.

Microsoft SQL Server

1. [Install the Easysoft ODBC-InterBase Driver](#) on same computer as SQL Server.
2. [Configure an ODBC data source](#).
3. In Microsoft SQL Server Management Studio, connect to the SQL Server instance you want to create the linked server against.

You need to log on with an account that is a member of the SQL Server sysadmin fixed server role to create a linked server.

4. Right-click **Server Objects**. From the pop-up menu choose **New > Linked Server**.
5. In the **Linked server** box, enter "InterBase".
6. From the **Provider** list, choose **Microsoft OLE DB Provider for ODBC drivers**.
7. In the **Data source** box, enter the name of your InterBase data source, and then choose **OK**.

SQL Server verifies the linked server by testing the connection.

- If you get the error "Specified driver could not be loaded due to system error 126: The specified module could not be found," choose **Yes** when prompted whether to keep the linked server. You need to restart your SQL Server instance before you can use the linked server. If SQL Server was already running when you installed the Easysoft ODBC-InterBase Driver, it will not have the latest version of the System Path environment variable. The Easysoft ODBC-InterBase Driver Setup program adds entries for the driver to the System Path. Restarting the instance makes these changes available to SQL Server, allowing it to load the Easysoft ODBC-InterBase Driver.
 - If you made a mistake when specifying the Easysoft ODBC-InterBase Driver, you get the error "Data source name not found and no default driver specified." If you get this error, choose **No** when prompted whether to keep the linked server and edit the value in the **Data source** box.
8. You can query your Easysoft ODBC-InterBase Driver data either by using a:
 - Four part table name in a distributed query.

A four part table name has the format:

```
server_name.[database_name].[schema_name].table_name
```

For data stores where there is no database or schema, Easysoft ODBC drivers return a "dummy" value for both identifiers, because some ODBC applications expect there to be a database and a schema. To find out the identifier names, run:

```
EXEC sp_tables_ex @table_server = 'InterBase'
```

If present, include these identifiers in your SQL statements. If not present, omit them. For example:

```
SELECT * FROM [InterBase]..DBO.Customers
```

The capitalisation of the table name must be the same as it is in the result set returned by sp_tables_ex.

- Pass-through query in an OPENQUERY function. For example:

```
SELECT * FROM OPENQUERY([InterBase], 'SELECT * FROM Customers')
```

```
-- If you get an "RPC not enabled for this server" message, right-click your
linked server and choose Properties.
```

```
-- In Server Options, set both RPC and RPC Out to `True`.
```

```
EXEC ('INSERT INTO Customers (Surname, GivenName, City, Phone, CompanyName)
VALUES ('Devlin' , 'Michaels' , 'Kingston' , '2015558966' ,
'PowerGroup'))
AT InterBase

UPDATE OPENQUERY ([InterBase], 'SELECT Surname FROM Customers WHERE
CompanyName = 'PowerGroup') SET Surname='Jones'
DELETE OPENQUERY (InterBase, 'SELECT Surname FROM Customers WHERE CompanyName
= 'PowerGroup')
```

SQL Server sends pass-through queries as uninterpreted query strings to the InterBase. This means that SQL Server does not apply any kind of logic to the query or try to estimate what that query will do.

Oracle

1. [Install the Easysoft ODBC-InterBase Driver](#) on same computer as Oracle.
2. [Configure an ODBC data source](#).
3. Follow the instructions for your Oracle platform.

Connecting InterBase to Oracle on Windows

1. Create a DG4ODBC init file on your Oracle machine. To do this, change to the %ORACLE_HOME%\hs\admin directory. Create a copy of the file initdg4odbc.ora. Name the new file initInterbase.ora.

Note In these instructions, replace %ORACLE_HOME% with the location of your Oracle HOME directory. For example, C:\app\product\21c\homes\OraDB21Home1.

2. Ensure these parameters and values are present in your init file:

```
HS_FDS_CONNECT_INFO = "InterBase"
```

Replace InterBase with the name of your Easysoft ODBC-InterBase Driver data source.

3. Comment out the line that enables DG4ODBC tracing. For example:

```
#HS_FDS_TRACE_LEVEL = <trace_level>
```

4. Add an entry to %ORACLE_HOME%\network\admin\listener.ora that creates a SID_NAME for DG4ODBC. For example:

```
SID_LIST_LISTENER =
( SID_LIST =
( SID_DESC =
( SID_NAME = Interbase )
( ORACLE_HOME = %ORACLE_HOME% )
( PROGRAM = dg4odbc )
)
)
```

5. Add a DG4ODBC entry to %ORACLE_HOME%\network\admin\tnsnames.ora that specifies the SID_NAME created in the previous step. For example:

```
Interbase =
( DESCRIPTION =
( ADDRESS = ( PROTOCOL = TCP ) ( HOST = oracle_host ) ( PORT = 1521 ) )
( CONNECT_DATA =
( SID = Interbase )
)
( HS = OK )
)
```

Replace oracle_host with the host name of your Oracle machine.

6. Start (or restart) the Oracle Listener:

```
cd %ORACLE_HOME%\bin
lsnrctl stop
```



```
lsnrctl start
```

7. Connect to your Oracle database in SQL*Plus.
8. In SQL*Plus, create a database link for InterBase. For example:

```
CREATE PUBLIC DATABASE LINK InterbaseLink
CONNECT TO "dbuser" IDENTIFIED BY "dbpassword"
USING 'Interbase';
```

Replace dbuser and dbpassword with your backend user name and password, if applicable.

9. Try querying and updating your InterBase data. For example:

```
SELECT "Surname" FROM "Customers"@InterbaseLink;

DECLARE
    num_rows integer;
BEGIN
    num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@InterbaseLink
('INSERT INTO Customers (Surname, GivenName, City, Phone, CompanyName) VALUES
('Devlin', 'Michaels', 'Kingston', '2015558966', 'PowerGroup')');
END;
/

DECLARE
    num_rows integer;
BEGIN
    num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@InterbaseLink
('UPDATE "Customers" SET "Surname" = ''Jones'' WHERE "CompanyName" =
''PowerGroup''');
END;
/

DECLARE
    num_rows integer;
BEGIN
    num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@InterbaseLink
('DELETE from "Customers" WHERE CompanyName = ''PowerGroup''');
END;
/
```

Notes

- If you have problems connecting to InterBase from Oracle, enable DG4ODBC tracing and check the trace files written to the %ORACLE_HOME%\hs\trace directory. To enable DG4ODBC tracing, add the line HS_FDS_TRACE_LEVEL = DEBUG to initInterBase.ora and then start or restart the Oracle listener. If the trace directory does not exist, create it.
- If you enable ODBC Driver Manager tracing, but do not get a log file in the location you specify, try looking in the top-level folder (for example, C:\SQL.log). Alternatively, in **ODBC Data Source Administrator**, change the trace file location to the Windows TEMP directory.

Connecting InterBase to Oracle on Linux and UNIX

42 Connecting InterBase to Oracle on Linux and UNIX

1. Create a DG4ODBC init file on your Oracle machine. To do this, change to the \$ORACLE_HOME\hs\admin directory. Create a copy of the file initdg4odbc.ora. Name the new file initInterbase.ora.

Note In these instructions, replace \$ORACLE_HOME with the location of your Oracle HOME directory. For example, /u01/app/oracle/product/21c/dbhome_1.

2. Ensure these parameters and values are present in your init file:

```
HS_FDS_CONNECT_INFO = "InterBase"
```

Replace InterBase with the name of your Easysoft ODBC-InterBase Driver data source.

3. Comment out the line that enables DG4ODBC tracing. For example:

```
#HS_FDS_TRACE_LEVEL = <trace_level>
```

4. Add an entry to \$ORACLE_HOME/network/admin/listener.ora that creates a SID_NAME for DG4ODBC. For example:

```
SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC=
(SID_NAME=Interbase)
(ORACLE_HOME=$ORACLE_HOME)
(PROGRAM=dg4odbc)
(ENVS=LD_LIBRARY_PATH = /usr/local/easysoft/unixODBC/lib:
/usr/local/easysoft/lib)
)
)
```

Replace oracle_home_directory with the value of \$ORACLE_HOME. For example, /u01/app/oracle/product/21c/dbhome_1.

5. Add a DG4ODBC entry to \$ORACLE_HOME/network/admin/tnsnames.ora that specifies the SID_NAME created in the previous step. For example:

```
Interbase =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = oracle_host)(PORT = 1521))
(CONNECT_DATA =
(SID = Interbase)
)
(HS = OK)
)
```

Replace oracle_host with the host name of your Oracle machine.

6. Start (or restart) the Oracle Listener:

```
cd $ORACLE_HOME/bin
./lsnrctl stop
./lsnrctl start
```

7. Connect to your Oracle database in SQL*Plus.

8. In SQL*Plus, create a database link for InterBase. For example:

```
CREATE PUBLIC DATABASE LINK InterbaseLink
  CONNECT TO "dbuser" IDENTIFIED BY "dbpassword"
  USING 'Interbase';
```

Replace dbuser and dbpassword with your backend user name and password, if applicable.

9. Try querying and updating your InterBase data. For example:

```
SELECT "Surname" FROM "Customers"@InterbaseLink;

DECLARE
  num_rows integer;
BEGIN
  num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@InterbaseLink
  ('INSERT INTO Customers (Surname, GivenName, City, Phone, CompanyName) VALUES
  ('Devlin', 'Michaels', 'Kingston', '2015558966', 'PowerGroup')');
END;
/

DECLARE
  num_rows integer;
BEGIN
  num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@InterbaseLink
  ('UPDATE "Customers" SET "Surname" = ''Jones'' WHERE "CompanyName" =
  ''PowerGroup''');
END;
/

DECLARE
  num_rows integer;
BEGIN
  num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@InterbaseLink
  ('DELETE from "Customers" WHERE CompanyName = ''PowerGroup''');
END;
/
```

Notes

- If you have problems connecting to InterBase from Oracle, enable DG4ODBC tracing and check the trace files written to the \$ORACLE_HOME/hs/trace directory. To enable DG4ODBC tracing, add the line HS_FDS_TRACE_LEVEL = DEBUG to initInterBase.ora and then start or restart the Oracle listener. If the trace directory does not exist, create it.

LibreOffice

1. [Install the Easysoft ODBC-InterBase Driver](#) on same computer as LibreOffice.
2. [Configure an ODBC data source](#).
3. Choose **File > New > Database**.
4. Choose **Connect to an existing database**.
5. Choose **ODBC** in the list, and then choose **Next**.
6. Choose **Browse**, double-click your data source, and then choose **Next**.
7. If your database requires a database user name, enter it in the **User name** box. If this user needs to supply a password choose the **Password required** check box.
8. Choose **Finish**.
9. Save the database when prompted.

The database opens in a new Base window. From here you can access your data.

10. In the left pane of the database window, choose the **Tables** icon to display a hierarchy of tables. Enter the database password if prompted, and then choose **OK**.
11. To retrieve the data in a table, in the **Tables** pane, double-click a table.
12. Choose the **Queries** icon to create a query.

Use any of the methods listed in the **Tasks** pane to create a query.

Go

1. [Install the Easysoft ODBC-InterBase Driver](#) on same computer as Go.
2. [Configure an ODBC data source](#).
3. Install the `odbc` package for Go:

```
go mod init test
go get github.com/alexbrainman/odbc
```

4. Create and then use Go to run this script, which retrieves some InterBase data:

```
package main

import (
    _ "github.com/alexbrainman/odbc"
    "database/sql"
    "log"
)

func main() {
    // Replace the DSN value with the name of your ODBC data source.
    db, err := sql.Open("odbc",
        "DSN=InterBase")
    if err != nil {
        log.Fatal(err)
    }

    var (
        name string
    )

    rows, err := db.Query("SELECT Surname FROM Customers")
    if err != nil {
        log.Fatal(err)
    }
    defer rows.Close()
    for rows.Next() {
        err := rows.Scan(&name)
        if err != nil {
            log.Fatal(err)
        }
        log.Println(name)
    }
    err = rows.Err()
    if err != nil {
        log.Fatal(err)
    }

    defer db.Close()
}
```

Node.js

1. [Install the Easysoft ODBC-InterBase Driver](#) on same computer as Node.js.
2. [Configure an ODBC data source.](#)
3. Install the `odbc` module for Node.js:

```
npm install odbc
```

4. Create and then use Node.js to run this script, which retrieves some InterBase data:

```
const odbc = require('odbc');
// Replace InterBase with the name of your Easysoft ODBC-InterBase Driver
// data source.
const connection = odbc.connect('DSN=InterBase', (error, connection) => {
  connection.query('SELECT Surname FROM Customers', (error, result) => {
    if (error) { console.error(error) }
    console.log(result);
  });
});
```

5. This script retrieves the tables and views in your Easysoft ODBC-InterBase Driver data source:

```
const odbc = require('odbc');
const connection = odbc.connect('DSN=InterBase', (error, connection) => {
  connection.tables(null, null, null, null, (error, result) => {
    if (error) { return; }
    const util = require('util');
    console.log(util.inspect(result, {maxLength: null, depth:null}))
  });
});
```

6. This script retrieves the names of the columns in these tables and views:

```
const odbc = require('odbc');
const connection = odbc.connect('DSN=InterBase', (error, connection) => {
  connection.columns(null, null, null, null, (error, result) => {
    if (error) { return; }
    const util = require('util');
    console.log(util.inspect(result, {maxLength: null, depth:null}))
  });
});
```

7. These scripts insert, update, and then delete some InterBase data:

```
const odbc = require("odbc");
const connection = odbc.connect("DSN=InterBase", (error, connection) => {
  connection.query("INSERT INTO
Customers (
  Surname,
  GivenName,
  City,
  Phone,
  CompanyName
```

```

    )
VALUES
    (
        'Devlin',
        'Michaels',
        'Kingston',
        '2015558966',
        'PowerGroup'
    ), (error, result) => {
        if (error) { console.error(error) }
        console.log(result);
    });
});

const odbc = require("odbc");
const connection = odbc.connect("DSN=InterBase", (error, connection) => {
    connection.query("UPDATE Customers SET Surname = 'Jones' WHERE CompanyName =
'PowerGroup'", (error, result) => {
        if (error) { console.error(error) }
        console.log(result);
    });
});

const odbc = require("odbc");
const connection = odbc.connect("DSN=InterBase", (error, connection) => {
    connection.query("DELETE FROM Customers WHERE CompanyName = 'PowerGroup'",
(error, result) => {
        if (error) { console.error(error) }
        console.log(result);
    });
});
});

```

Perl

1. [Install the Easysoft ODBC-InterBase Driver](#) on same computer as Perl.
2. [Configure an ODBC data source](#).
3. Check whether your Perl distribution supports ODBC:

```
perl -e 'use DBD::ODBC;'
```

4. Do one of the following:
 - If you get no output, your Perl distribution supports ODBC. Skip to the next step.
 - If you get:

```
Can't locate DBD/ODBC.pm
```

you need to [install DBD::ODBC](#) before you can use the Easysoft ODBC-InterBase Driver to connect to InterBase.

5. Create and then use Perl to run this script, which retrieves some InterBase data:

```
use strict;
use DBI;
# Replace InterBase with the name of your Easysoft ODBC-InterBase Driver data
source.
my $dbh = DBI-> connect('dbi:ODBC:InterBase');

my $sql = "SELECT Surname FROM Customers";

my $sth = $dbh->prepare($sql)
    or die "Can't prepare statement: $DBI::errstr";

$sth->execute();

my($Col);

# Fetch and display the result set values.
while(($Col) = $sth->fetchrow()){
    print("$Col\n");
}

$dbh->disconnect if ($dbh);
```

6. This script retrieves the tables and views in your Easysoft ODBC-InterBase Driver data source:

```
use strict;
use DBI;
my $dbh = DBI-> connect('dbi:ODBC:InterBase');

my $sth = $dbh->table_info()
    or die "Can't prepare statement: $DBI::errstr";

my @row;

while (@row = $sth->fetchrow_array) {
```



```

        print join(", ", @row), "\n";
    }

    $dbh->disconnect if ($dbh);

```

7. This script retrieves the names of the columns in these tables and views:

```

use strict;
use DBI;
my $dbh = DBI-> connect('dbi:ODBC:InterBase');

my $sth = $dbh->column_info('', '', '', '')
    or die "Can't prepare statement: $DBI::errstr";

my @row;
while (@row = $sth->fetchrow_array) {
    print join(", ", @row), "\n";
}

$dbh->disconnect if ($dbh);

```

8. These scripts insert, update, and then delete some InterBase data:

```

use strict;
use DBI;
my $dbh = DBI-> connect('dbi:ODBC:InterBase');

my $sth = $dbh->prepare(q/INSERT INTO Customers (Surname, GivenName, City, Phone,
CompanyName) VALUES (?, ?, ?, ?, ?)/)
    or die "Can't prepare statement: $DBI::errstr";

$sth->execute('Devlin', 'Michaels', 'Kingston', '2015558966', 'PowerGroup');

$dbh->disconnect if ($dbh);

use strict;
use DBI;
my $dbh = DBI-> connect('dbi:ODBC:InterBase');

my $sth = $dbh->prepare('UPDATE Customers SET Surname = \'Jones\' WHERE
CompanyName = ?')
    or die "Can't prepare statement: $DBI::errstr";

$sth->execute('PowerGroup');

$dbh->disconnect if ($dbh);

use strict;
use DBI;
my $dbh = DBI-> connect('dbi:ODBC:InterBase');

my $sth = $dbh->prepare('DELETE FROM Customers WHERE CompanyName = ?')

```

```
        or die "Can't prepare statement: $DBI::errstr";

$sth->execute('PowerGroup');

$dbh->disconnect if ($dbh);
```

Further information

- [Perl DBI DBD::ODBC tutorial: Drivers, data sources, and connection](#)

PHP

1. [Install the Easysoft ODBC-InterBase Driver](#) on same computer as PHP.
2. [Configure an ODBC data source](#).
3. Check whether your PHP distribution supports ODBC. In php.ini, make sure there is no comment character (;) before the extension_dir and extension=odbc settings (;extension_dir=directory becomes extension_dir=directory and ;extension=odbc becomes extension=odbc).
4. Create and then use PHP to run this script, which retrieves some InterBase data:

```
<?php
// Replace InterBase with the name of your Easysoft ODBC-InterBase Driver data
source.
// If your database requires a user name and password, supply them in the
odbc_connect_call.
$con = odbc_connect("InterBase", "", "");
$stmt = odbc_exec($con, "SELECT * FROM Customers");
// You may need to change the capitalisation of Surname to all upper case or
all lower case.
while ($row = odbc_fetch_array($stmt)) {
    echo "Surname = ", $row["Surname"], "\n";
}
odbc_close($con);
?>
```

5. This script retrieves the tables and views in your Easysoft ODBC-InterBase Driver data source:

```
<?php
$con = odbc_connect("InterBase", "", "");
$tables = odbc_tables($con);
while (($row = odbc_fetch_array($tables))) {
    print_r($row);
}
odbc_close($con);
?>
```

6. This script retrieves the names of the columns in these tables and views:

```
<?php
$con = odbc_connect("InterBase", "", "");
$columns = odbc_columns($con);
while (($row = odbc_fetch_array($columns))) {
    print_r($row);
}
odbc_close($con);
?>
```

7. These scripts insert, update, and then delete some InterBase data:

```
<?php
$cnx = odbc_connect("InterBase", "", "");
$stmt = odbc_prepare($cnx, "INSERT INTO Customers (Surname, GivenName, City,
Phone, CompanyName) VALUES (?, ?, ?, ?, ?)");
```

```
$success = odbc_execute($stmt, array('Devlin', 'Michaels', 'Kingston',  
'2015558966', 'PowerGroup'));  
odbc_close($cnx);  
?>  
  
<?php  
$cnx = odbc_connect("InterBase", "", "");  
$stmt = odbc_prepare($cnx, "UPDATE Customers SET Surname = 'Jones' WHERE  
CompanyName = ?");  
$success = odbc_execute($stmt, array('PowerGroup'));  
odbc_close($cnx);  
?>  
  
<?php  
$cnx = odbc_connect("InterBase", "", "");  
$stmt = odbc_prepare($cnx, "DELETE FROM Customers WHERE CompanyName = ?");  
$success = odbc_execute($stmt, array('PowerGroup'));  
odbc_close($cnx);  
?>
```

Further information

- [Easysoft PHP tutorials and code samples](#)

Python

1. [Install the Easysoft ODBC-InterBase Driver](#) on same computer as Python.
2. [Configure an ODBC data source](#).
3. Check whether your Python distribution supports ODBC.

```
pip list
```

If you don't have pip installed:

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python get-pip.py
```

4. Do one of the following:
 - If the output contains pyodbc, your Python distribution supports ODBC. Skip to the next step.
 - If the output does not contain pyodbc, use pip to install this module:

```
pip install pyodbc
```

5. Create and then use Python to run this script, which retrieves some InterBase data:

```
import pyodbc

# Replace InterBase with the name of your Easysoft ODBC-InterBase Driver data
source.
cnxn = pyodbc.connect("DSN=InterBase")
cursor = cnxn.cursor()
sql = "SELECT Surname FROM Customers"
cursor.execute(sql)
rows = cursor.fetchall()
# You may need to change the capitalisation of Surname to all upper case or all
lower case.
for row in rows:
    print(row.Surname)
exit()
```

6. This script retrieves the tables and views in your Easysoft ODBC-InterBase Driver data source:

```
import pyodbc

# Replace InterBase with the name of your Easysoft ODBC-InterBase Driver data
source.
cnxn = pyodbc.connect("DSN=InterBase")
cursor = cnxn.cursor()
cursor.tables()
rows = cursor.fetchall()
for row in rows:
    print(row.table_name)
exit()
```

7. This script retrieves the names of the columns in these tables and views:

```
import pyodbc
```

```
# Replace InterBase with the name of your Easysoft ODBC-InterBase Driver data
source.
cnxn = pyodbc.connect("DSN=InterBase")
cursor = cnxn.cursor()
cursor.columns()
rows = cursor.fetchall()
for row in rows:
    print(row.table_name, row.column_name)
exit()
```

8. These scripts insert, update, and then delete some InterBase data:

```
import pyodbc

cnxn = pyodbc.connect("DSN=InterBase")
cursor = cnxn.cursor()
sql = "INSERT INTO Customers (Surname, GivenName, City, Phone, CompanyName)
VALUES (?, ?, ?, ?, ?)"
cursor.execute(sql, 'Devlin', 'Michaels', 'Kingston', '2015558966', 'PowerGroup')
cursor.commit()
exit()
```

```
import pyodbc

cnxn = pyodbc.connect("DSN=InterBase")
cursor = cnxn.cursor()
sql = "UPDATE Customers SET Surname = 'Jones' WHERE CompanyName = ?"
cursor.execute(sql, 'PowerGroup')
cursor.commit()
exit()
```

```
import pyodbc

cnxn = pyodbc.connect("DSN=InterBase")
cursor = cnxn.cursor()
sql = "DELETE FROM Customers WHERE CompanyName = ?"
cursor.execute(sql, 'PowerGroup')
cursor.commit()
exit()
```

Further information

- [Easysoft Python tutorials and code samples](#)

R

1. [Install the Easysoft ODBC-InterBase Driver](#) on same computer as R.
2. [Configure an ODBC data source](#).
3. In R Console, check whether your R distribution supports ODBC.

```
library("RODBC")
```

4. Do one of the following:
 - If you get no output, you have the ODBC library for R. Skip to the next step.
 - If you get an "there is no package" error, install the ODBC library for R:

```
install.packages("RODBC")
```

5. Create and then use R to run this script, which retrieves some InterBase data:

```
library("RODBC")
# Replace InterBase with the name of your Easysoft ODBC-InterBase Driver data
source.
ch <- odbcConnect("InterBase")
sqlQuery(ch, paste("SELECT Surname FROM Customers"))
odbcClose(ch)
quit()
```

6. This script retrieves the tables and views in your Easysoft ODBC-InterBase Driver data source:

```
library("RODBC")
# Replace InterBase with the name of your Easysoft ODBC-InterBase Driver data
source.
ch <- odbcConnect("InterBase")
sqlTables(ch)
odbcClose(ch)
quit()
```

7. This script retrieves the names of the columns in the specified table or view:

```
library("RODBC")
# Replace InterBase with the name of your Easysoft ODBC-InterBase Driver data
source.
ch <- odbcConnect("InterBase")
# You may need to change the capitalisation of Customers to all upper case or all
lower case.
sqlColumns(ch, sqtable="Customers")
odbcClose(ch)
quit()
```

8. These scripts insert, update, and then delete some InterBase data:

```
library("RODBC")
ch <- odbcConnect("InterBase")
sqlQuery(ch, paste("INSERT INTO Customers (Surname, GivenName, City, Phone,
CompanyName) VALUES ('Devlin', 'Michaels', 'Kingston', '2015558966',
'PowerGroup')"))
```

```
odbcClose(ch)
quit()

library("RODBC")
ch <- odbcConnect("InterBase")
sqlQuery(ch, paste("UPDATE Customers SET Surname = 'Jones' WHERE CompanyName =
'PowerGroup'"))
odbcClose(ch)
quit()

library("RODBC")
ch <- odbcConnect("InterBase")
sqlQuery(ch, paste("DELETE FROM Customers WHERE CompanyName = 'PowerGroup'"))
odbcClose(ch)
quit()
```

About the Easysoft ODBC-InterBase Driver

The Easysoft ODBC-InterBase Driver provides real-time access to InterBase data from any application that supports ODBC.

- [ODBC API and scalar functions](#)
- [Data type mapping](#)
- [Cursor support](#)
- [Multiple transactions on a single connection](#)
- [Runtime SQL dialect support](#)
- [Rollback retaining](#)
- [SQL support](#)

ODBC API and scalar functions

API functions

Use this table to find out what ODBC API functions the Easysoft ODBC-InterBase Driver supports:

Function	Status
SQLAllocConnect	Supported
SQLAllocEnv	Supported
SQLAllocHandle	Supported
SQLAllocStmt	Supported
SQLBindCol	Supported
SQLBindParameter	Supported
SQLBrowseConnect	Not supported
SQLBulkOperations	Not supported
SQLCancel	Supported
SQLCloseCursor	Supported
SQLColAttribute	Supported
SQLColAttributes	Supported
SQLColumnPrivileges	Supported
SQLColumns	Supported
SQLConnect	Supported
SQLCopyDesc	Not supported
SQLDataSources	Supported
SQLDescribeCol	Supported
SQLDescribeParam	Supported
SQLDisconnect	Supported
SQLDriverConnect	Supported
SQLDrivers	Supported
SQLEndTran	Supported
SQLError	Supported
SQLExecDirect	Supported
SQLExecute	Supported
SQLExtendedFetch	Supported
SQLFetch	Supported
SQLFetchScroll	Supported
SQLForeignKeys	Supported
SQLFreeConnect	Supported

Function	Status
SQLFreeEnv	Supported
SQLFreeHandle	Supported
SQLFreeStmt	Supported
SQLGetConnectAtt	Supported
SQLGetConnectOption	Supported
SQLGetCursorName	Supported
SQLGetData	Supported
SQLGetDescField	Supported
SQLGetDescRec	Supported
SQLGetDiagField	Supported
SQLGetDiagRec	Supported
SQLGetEnvAttr	Supported
SQLGetFunctions	Supported
SQLGetInfo	Supported
SQLGetStmtAttr	Supported
SQLGetStmtOption	Supported
SQLGetTypeInfo	Supported
SQLMoreResults	Supported
SQLNativeSql	Supported
SQLNumParams	Supported
SQLNumResultCols	Supported
SQLParamData	Supported
SQLParamOptions	Supported
SQLPrepare	Supported
SQLPrimaryKeys	Supported
SQLProcedureColumns	Supported
SQLProcedures	Supported
SQLPutData	Supported
SQLRowCount	Supported
SQLSetConnectAttr	Supported
SQLSetConnectOption	Supported
SQLSetCursorName	Supported
SQLSetDescField	Supported
SQLSetDescRec	Supported
SQLSetEnvAttr	Supported

Function	Status
SQLSetParam	Supported
SQLSetPos	Not supported
SQLSetScrollOptions	Not supported
SQLSetStmtOption	Supported
SQLSetStmtOption	Supported
SQLSetStmtAttr	Supported
SQLStatistics	Supported
SQLTablePrivileges	Supported
SQLTables	Supported
SQLTransact	Supported

Data type mapping

The Easysoft ODBC-InterBase Driver maps InterBase data types to ODBC data types in this way:

InterBase data type	ODBC data type
BLOB	SQL_LONGVARBINARY
BLOB SUB_TYPE TEXT	SQL_LONGVARCHAR
CHARACTER	SQL_CHAR
CHARACTER VARYING	SQL_VARCHAR
DATE	SQL_TYPE_DATE
DECIMAL	SQL_DECIMAL
DOUBLE PRECISION	SQL_DOUBLE
FLOAT	SQL_FLOAT
INTEGER	SQL_INTEGER
NATIONAL CHARACTER	SQL_WCHAR
NATIONAL CHARACTER VARYING	SQL_WVARCHAR
NUMERIC	SQL_NUMERIC
SMALLINT	SQL_SMALLINT
TIME	SQL_TYPE_TIME
TIMESTAMP	SQL_TYPE_TIMESTAMP

Finding out more about data types on Windows

If you need more information about a data types, for example, the precision and scale, use Microsoft's ODBC Test to do this.

1. Download the version of ODBC Test that matches your application's architecture from:
<https://www.easysoft.com/ftp/pub/utils/windows/odbc-test/>
2. Copy both files to a folder on the machine where Easysoft ODBC-InterBase Driver is installed.
3. Double-click **odbcte32.exe**.
4. Select **Con > Full Connect**.
5. Choose your Easysoft ODBC-InterBase Driver data source from the list.
6. Choose **Catalog > SQLGetTypeInfo**.
7. Either choose **SQL_ALL_TYPES=0 (1.0)** or a specific data type from the **DataType** list.
8. Choose **Results > Get Data All**.

Cursor support

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 works with ADO snapshots, but not with dynasets.

Multiple transactions on a single connection

Multiple transactions on a single connection aren't supported in ODBC 3.x and therefore can't be supported by the Easysoft ODBC-InterBase Driver.

Runtime SQL dialect support

Currently, the required SQL dialect can be set for a connection, but can't be set at runtime, as it's not a statement level property.

Rollback retaining

The Easysoft ODBC-InterBase Driver supports the `isc_rollback_retaining` function if it's present in the InterBase client library.

SQL support

The Easysoft ODBC-InterBase Driver supports these SQL statements, clauses, and operators:

- SELECT
- SELECT DISTINCT
- WHERE
- ORDER BY
- AND
- OR
- NOT
- INSERT INTO
- NULL
- UPDATE
- DELETE
- FIRST
- MIN
- MAX
- COUNT
- SUM
- AVG
- LIKE
- WILDCARDS
- IN
- BETWEEN
- ALIASES
- JOINS
- UNION
- GROUP BY
- HAVING
- EXISTS
- CASE

Example queries

- To fetch all records from a table, use the asterisk symbol (*) in your queries. For example:

```
SELECT * FROM Customers
```

- To only fetch records whose values are different, use DISTINCT. For example:

```
-- Which different sales regions are there?  
SELECT DISTINCT Region AS Different_Regions FROM SalesOrders  
-- How many different sales regions are there?  
SELECT COUNT(DISTINCT Region) AS Different_Regions FROM SalesOrders
```

- To filter records, use WHERE. For example:

```
SELECT
    OrderDate,
    SalesRepresentative
FROM
    SalesOrders
WHERE
    Region = 'Eastern'

SELECT
    OrderDate,
    SalesRepresentative
FROM
    SalesOrders
WHERE
    Region = 'Eastern'
    OR Region = 'Western'

SELECT
    OrderDate,
    SalesRepresentative
FROM
    SalesOrders
WHERE
    Region = 'Eastern'
    AND EXTRACT (YEAR FROM OrderDate) = 2025
```

You can also supply a WHERE clause value as a parameter. For example, to do this in [Python](#):

```
cursor.execute("SELECT
    OrderDate,
    SalesRepresentative
FROM
    SalesOrders
WHERE
    Region = ?", ['Eastern'])
```

68 Example queries

- To fetch records that don't match the WHERE clause pattern use NOT. For example:

```
SELECT
    OrderDate,
    SalesRepresentative
FROM
    SalesOrders
WHERE
    NOT Region = 'Eastern'
```

- To sort the result set in either ascending or descending order, use ORDER BY. For example:

```
SELECT
    *
FROM
    SalesOrders
ORDER BY
    OrderDate ASC

SELECT
    *
FROM
    Contacts
ORDER BY
    (
        CASE
            WHEN Surname IS NULL THEN Title
            ELSE Surname
        END
    );
```

- To group a result set into summary rows, use GROUP BY. For example:

```
SELECT
    COUNT(Id) As "Number",
    ProductID
FROM
    SalesOrderItems
GROUP BY
    ProductID
```

```
SELECT
    COUNT(Id) As "Number",
    ProductID
FROM
    SalesOrderItems
GROUP BY
    ProductID
HAVING
    COUNT(Id) > 100;
```

- To do calculations based on result set values, use the SQL aggregate functions MIN(), MAX(), COUNT(), SUM(), and AVG(). For example:

```
SELECT Max(Quantity) FROM SalesOrderItems
SELECT Sum(Quantity) FROM SalesOrderItems
```

- To convert between compatible data types, use CAST. For example:

```
SELECT CAST(Quantity AS Char(100))FROM SalesOrderItems
```

- To fetch records that contain column values between a given range, use BETWEEN. For example:

```
SELECT ProductID FROM SalesOrderItems WHERE Quantity BETWEEN 10 AND 20
```

70 Example queries

- To combine the result set of two or more SELECT statements, use UNION. For example:

```
SELECT City FROM Contacts
UNION
SELECT City FROM Customers
```

- To combine rows from two or more tables, use JOIN. For example:

```
SELECT SalesOrders.ID, Customers.Surname, SalesOrders.OrderDate
FROM SalesOrders
INNER JOIN Customers ON SalesOrders.CustomerID=Customers.ID;
```

- To fetch records that contain column values matching a search pattern, use LIKE. For example:

```
SELECT Surname, GivenName FROM Customers WHERE CompanyName LIKE 'R%'
SELECT Surname, GivenName FROM Customers WHERE CompanyName LIKE '_he'
```

- To search for columns without a value (NULL) or with a value (non NULL), use either IS NULL or IS NOT NULL. For example:

```
SELECT * FROM Customers WHERE CompanyName IS NULL
```

- To specify multiple values in a WHERE clause, you can use IN as an alternative to OR. For example:

```
SELECT
    OrderDate,
    SalesRepresentative
FROM
    SalesOrders
WHERE
    Region = 'Eastern'
    OR Region = 'Western'
    OR Region = 'Central'
```

can be replaced with:

```
SELECT
    OrderDate,
    SalesRepresentative
FROM
    SalesOrders
WHERE
    Region IN ('Eastern', 'Western', 'Central')
```

- To set the maximum number of records to return, use FIRST. For example:

```
SELECT FIRST 10 * FROM Customers
```

- To test for the existence of records in a subquery, use EXISTS. For example:

```
SELECT
    Name
FROM
    Products
WHERE
    EXISTS (
        SELECT
            *
        FROM
            SalesOrderItems
        WHERE
            Products.ID = SalesOrderItems.ProductID
            AND Quantity < 20
    )
```

Example inserts, updates, and deletes

- To insert a InterBase record, use INSERT INTO. For example:

```
INSERT INTO
    Customers (
        Surname,
        GivenName,
        City,
        Phone,
        CompanyName
    )
VALUES
    (
        'Devlin',
        'Michaels',
        'Kingston',
        '2015558966',
        'PowerGroup'
    )
```

- Here's a SQL Server linked server example:

```
EXEC ('INSERT INTO Customers (Surname, GivenName, City, Phone, CompanyName)
VALUES ('Devlin' , 'Michaels' , 'Kingston' , '2015558966' ,
'PowerGroup'))
```


- Here's an Oracle linked table example:

```
DECLARE
    num_rows integer;
BEGIN
    num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@InterbaseLink
    ('INSERT INTO Customers (Surname, GivenName, City, Phone, CompanyName) VALUES
    ('Devlin', 'Michaels', 'Kingston', '2015558966', 'PowerGroup')');
END;
/
```

- The Easysoft ODBC-InterBase Driver also supports parameterized inserts. Here's an example of doing this in [Perl](#):

```
my $sth = $dbh->prepare(q/INSERT INTO Customers (Surname, GivenName, City, Phone,
CompanyName) VALUES (?, ?, ?, ?, ?)/)
    or die "Can't prepare statement: $DBI::errstr";

$sth->execute('Devlin', 'Michaels', 'Kingston', '2015558966', 'PowerGroup');
```

- To update a InterBase record, use UPDATE. For example:

```
UPDATE Customers
SET
    Surname = 'Jones'
WHERE
    Account_Id = 'PowerGroup'
```

The Easysoft ODBC-InterBase Driver also supports parameterized updates. Here's an example of doing this in [Perl](#):

```
my $sth = $dbh->prepare('UPDATE Customers SET Surname = \'Jones\' WHERE
CompanyName = ?')
    or die "Can't prepare statement: $DBI::errstr";

$sth->execute('PowerGroup');
```

- To delete a InterBase record, use DELETE. For example:

```
-- Delete (mark inactive) a bank account  
DELETE FROM Customers WHERE CompanyName = 'PowerGroup'
```

The Easysoft ODBC-InterBase Driver also supports parameterized deletes. Here's an example of doing this in [Python](#):

```
sql = "DELETE FROM Customers WHERE CompanyName = ?"  
cursor.execute(sql, 'PowerGroup')
```

Index

A

Access

- importing InterBase data, [35](#)
- linking InterBase data, [35](#)

B

Base

- working with InterBase data, [44](#)

C

- Character Set attribute, [25](#)
- Commit Select attribute, [26](#)
- connecting to InterBase, [17](#)
- connection string attributes, [29](#)
- cursor support, [62](#)

D

- data source attributes, [19](#)
- data sources
 - removing
 - on Windows, [16](#)
- data type mapping, [61](#)
- Database attribute, [21](#)
- Default SQLCHAR attribute, [28](#)
- Description attribute, [20](#)
- DG4ODBC
 - working with InterBase data, [40](#)
- Dialect attribute, [22](#)
- DSN attribute, [20](#)
- DSN-less connections, [29](#)

E

Easysoft ODBC-InterBase Driver

- adding ODBC data sources
 - on Linux or UNIX, [17](#)
 - on Windows, [18](#)
- connecting to InterBase, [17](#)
- data source attributes, [19](#)
- data type mapping, [61](#)
- DSN-less connections, [29](#)
- installing
 - on Linux or UNIX, [5](#)
 - on Windows, [14](#)
- InterBase client, [4](#)
- logging, [30](#)
- ODBC API support, [58](#)
- SQL support, [66](#)
- uninstalling
 - on Linux or UNIX, [13](#)
 - on Windows, [16](#)

Excel

- importing InterBase data with Data Connection Wizard, [36](#)

- importing InterBase data with PowerPivot, [36](#)
- importing InterBase data with Query, [36](#)

Execute Proc attribute, [24](#)

F

- Fix Precision attribute, [27](#)
- Flush Commit attribute, [26](#)

G

Go

- working with InterBase data, [45](#)

H

- Hide LONGVARCHAR attribute, [28](#)

I

- installing the Easysoft ODBC-InterBase Driver, [5](#)
- InterBase client, [4](#)

L

- Leave Quotes attribute, [25](#)

LibreOffice

- working with InterBase data, [44](#)

linked server

- working with InterBase data, [38](#)

log files, [30](#)

N

- No Wait attribute, [22](#)

Node.js

- working with InterBase data, [46](#)

Null Schema attribute, [27](#)

O

- ODBC API function support, [58](#)
- ODBC connection string attributes, [29](#)
- ODBC data sources
 - adding
 - on Linux or UNIX, [17](#)
 - on Windows, [18](#)
 - removing
 - on Windows, [16](#)

Oracle

- working with InterBase data, [40](#)

P

- Pad Varchar attribute, [27](#)
- Password attribute, [21](#)
- Perl
 - working with InterBase data, [48](#)

PHP

- working with InterBase data, [51](#)

Power BI

- importing InterBase data, [37](#)

PowerPivot

- importing InterBase data, [36](#)

- Pre IB6 MetaData attribute, [23](#)

- Python

- working with InterBase data, [53](#)

- R

- R

- working with InterBase data, [55](#)

- Read Only attribute, [25](#)

- Role attribute, [21](#)

- rollback retaining, [65](#)

- S

- Simple UNICODE attribute, [28](#)

- SQL dialect, [64](#)

- SQL Server

- working with InterBase data, [38](#)

- SQL support, [66](#)

- T

- trace files, [30](#)

- transactions

- multiple, [63](#)

- Txn Isolation attribute, [26](#)

- U

- UID Override attribute, [25](#)

- uninstalling

- on Linux or UNIX, [13](#)

- on Windows, [16](#)

- User Name attribute, [21](#)

- W

- WCMB Mode attribute, [28](#)

- With Default attribute, [26](#)

- With Schema attribute, [22](#)