

# **Easysoft ODBC-Xero driver User's Guide**

This manual documents version 1.1.n of the Easysoft ODBC-Xero 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-Xero driver . . . . .	5
Installing on Linux or UNIX . . . . .	5
Uninstalling on Linux or UNIX . . . . .	13
Installing on Windows . . . . .	14
Uninstalling on Windows . . . . .	16
Connecting to Xero . . . . .	17
Connecting from Linux or UNIX . . . . .	17
Connecting from Windows . . . . .	18
Xero app . . . . .	19
Xero authentication . . . . .	19
Connection attributes . . . . .	20
DSN-less connections . . . . .	23
Logging . . . . .	24
ODBC Driver Manager logging on Linux or UNIX . . . . .	24
Easysoft ODBC-Xero driver logging on Linux and UNIX . . . . .	24
ODBC Driver Manager logging on Windows . . . . .	24
Easysoft ODBC-Xero driver logging on Windows . . . . .	26
Finding out what product version you have on Windows . . . . .	27
Client applications . . . . .	28
Microsoft Access . . . . .	29
Linking a table . . . . .	29
Importing a table . . . . .	29
Microsoft Excel . . . . .	30
Data Connection Wizard . . . . .	30
Microsoft Query . . . . .	30
PowerPivot . . . . .	30
Microsoft Power BI . . . . .	31
Microsoft SQL Server . . . . .	32
Oracle . . . . .	34
Connecting Xero to Oracle on Windows . . . . .	34
Connecting Xero to Oracle on Linux and UNIX . . . . .	35
LibreOffice . . . . .	38
Go . . . . .	39
Node.js . . . . .	40
Perl . . . . .	42
PHP . . . . .	45
Python . . . . .	47
R . . . . .	49
About the Easysoft ODBC-Xero driver . . . . .	51
ODBC API and scalar functions . . . . .	52
API functions . . . . .	52
Scalar functions . . . . .	54
Data type mapping . . . . .	57
Finding out more about data types on Windows . . . . .	58
SQL support . . . . .	59
Index . . . . .	60

## Getting started

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

The following section shows you how to install the Easysoft ODBC-Xero driver and configure the ODBC data source that stores the connection details for your Xero organisation. You're then ready to work with Xero data in your application.

- [Installing the Easysoft ODBC-Xero driver](#)
- [Connecting to Xero](#)
- [Logging](#)

# Installing the Easysoft ODBC-Xero driver

Install the Easysoft ODBC-Xero driver on the computer where the application you want to connect to Xero 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-Xero 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 Xero 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-Xero 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.

- 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-Xero driver in /usr/local/easysoft.
  - Install the Easysoft ODBC-Xero 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-Xero 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-Xero 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-Xero 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-Xero driver.
- The unixODBC Driver Manager.

You need an ODBC Driver Manager to use the Easysoft ODBC-Xero 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-Xero 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-Xero 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-xero-1.1.0-linux-x86-64-ubuntu164.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-xero-1.1.0-linux-x86-64-ubuntu164
```

### 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-Xero driver and data sources during the installation.
- Most applications and interfaces that support ODBC are compatible with unixODBC. The Easysoft ODBC-Xero 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-Xero 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-Xero 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-Xero driver installation script:

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

If the Easysoft ODBC-Xero driver is already registered with unixODBC, a warning displays that lists the drivers unixODBC knows about. If you're installing the Easysoft ODBC-Xero 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-Xero driver data source. If unixODBC is installed and you registered the Easysoft ODBC-Xero 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-Xero driver license:

```
Would you like to request a Easysoft ODBC-Xero 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-Xero 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-Xero 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-Xero 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-Xero 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-Xero 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-Xero 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-Xero driver entry needs to be removed from the odbcinst.ini file. To check whether the Easysoft ODBC-Xero driver is configured under unixODBC, use odbcinst -q -d. If the command output contains [Easysoft ODBC-Xero], uninstall the driver from unixODBC by using:

```
odbcinst -u -d -n Easysoft ODBC-Xero
```

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-Xero 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 Easysoft ODBC-Xero.
2. Remove the install.info for the Easysoft ODBC-Xero 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-Xero 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-Xero 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-Xero 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-Xero 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-Xero 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-Xero 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](https://www.easysoft.com/support/licensing/trial_license.html) or [https://www.easysoft.com/support/licensing/full\\_license.html](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-Xero driver, the installation is complete.

## Repairing the installation

The installer can repair a broken Easysoft ODBC-Xero driver installation. For example, you can use the installer to restore missing Easysoft ODBC-Xero 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-Xero driver in the list, and then choose **Repair**.

## Uninstalling on Windows

This section explains how to remove the Easysoft ODBC-Xero driver from your system.

### Removing Easysoft ODBC-Xero driver data sources

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

1. In the Windows **Taskbar**, enter Add or remove programs in the Windows **Search** box.
2. Select Easysoft ODBC-Xero 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 Xero

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-Xero 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 Xero data:

- Create a SYSTEM data source, which is available to anyone who logs on to the computer where the Easysoft ODBC-Xero 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-Xero driver is installed.

By default, the Easysoft ODBC-Xero driver installation creates a sample SYSTEM data source named XERO\_SAMPLE. If you're using the unixODBC included in the Easysoft ODBC-Xero 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 Xero, 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-Xero driver is installed into unixODBC, it places a Easysoft ODBC-Xero entry into the `odbcinst.ini` file. You should always have `Driver = Easysoft ODBC-Xero` in your Easysoft ODBC-Xero driver data sources therefore.

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

- Your Xero organisation (Organisation).
- The Client ID for the Xero app you created for this ODBC data source. (Client\_Id).
- The Client Secret for the Xero app you created for this ODBC data source. (Client\_Secret).

For example:

```
[Xero]
Driver           = Easysoft ODBC-Xero
Organisation     = Demo Company (UK)
Client_Id        = myuser@mydomain
Client_Secret    = mypassword
```

The Easysoft ODBC-Xero 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-Xero driver data sources. To test the Easysoft ODBC-Xero 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 Accounts;
```

–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-Xero driver, and then choose **Finish**.
6. Complete the Easysoft ODBC-Xero driver configuration dialog box.  
To find out how to do this, refer to the Connection attributes section.
7. To test the connection to Xero, choose **Test**.  
Note that this doesn't test that the Easysoft ODBC-Xero driver is licensed. If you haven't yet [licensed](#) the Easysoft ODBC-Xero driver, this ODBC data source won't work with your application, even if the **Test** button succeeds.

## Xero app

To use the Easysoft ODBC-Xero driver, you first need to create a Xero app in the Xero Developer Platform web site.

Create a Xero app, which the Xero OAuth mechanism will use to authorise and keep track of your ODBC connection. We recommend that you create a Xero app for each Xero ODBC data source that you create.

To do this:

1. Add a Xero OAuth application (<https://developer.xero.com/app/manage>), by choosing **New App**.
2. In the **Add a new app** screen, enter a meaningful name in the **App name** box.
3. In the **OAuth 2.0 grant type** section, choose **Auth code Web app**.
4. Enter your organisation's web site URL in the **Company or application URL** box.
5. In the OAuth 2.0 redirect URL box, enter:

```
http://localhost:9998
```

6. Choose the **Create app** button.
7. In the **App details** page for the newly created app, copy the **Client id**. You will need this to create your data source.
8. Generate and copy a **Client secret**. You will need this to create your data source.

## Xero authentication

To use the Easysoft ODBC-Xero driver to work with your Xero data, you need to obtain a refresh token.

To obtain a refresh token on Windows, choose the **Authenticate and obtain token** button in the configuration dialog box when setting up your data source. This launches your default browser.

Sign into your Xero account and then grant the Easysoft ODBC-Xero driver the access permissions it needs when prompted. Then follow the onscreen instructions.

To obtain a refresh token on Linux or UNIX, change to the `<installation_directory>/easysoft/xero_oauth2` directory and then run `xero_oauth2`. In a web browser on the same machine, go to the URL returned by the program. Sign into your Xero account and then grant the Easysoft ODBC-Xero driver the access permissions it needs when prompted. Then follow the onscreen instructions. Paste the refresh token returned by following this process into your ODBC data source in `odbc.ini`. The refresh token is the value for the `Refresh-Token` attribute.

## Connection attributes

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

## Setting on Linux and UNIX

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

- Your Xero organisation (Organisation).
- The Client ID for the Xero app you created for this ODBC data source. (`Client_Id`).
- The Client Secret for the Xero app you created for this ODBC data source. (`Client_Secret`).

For example:

```
[Xero]
Driver           = Easysoft ODBC-Xero
Organisation     = Demo Company (UK)
Client_Id        = myuser@mydomain
Client_Secret    = mypassword
```

These optional attributes may be set in `odbc.ini`.

- Description
- Filter\_Local
- Logfile
- Logging
- OAuthTokenLocation
- Proxy
- Proxy\_Pass
- Proxy\_User
- Refresh-Token
- ShareOAuthToken
- Uri

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

## Setting on Windows

The Easysoft ODBC-Xero driver data source configuration dialog box, accessible when you create or edit an Easysoft ODBC-Xero driver data source in **ODBC Data Source Administrator** contains a number of attribute 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.
Client ID	The Client ID for the Xero app you created for this ODBC data source.
Client Secret	The Client Secret for the Xero app you created for this ODBC data source.
Organisation	The organisation to connect to. For example, Demo Company (UK).
OAuthfile	<p>The path to a file where updated refresh tokens can be stored. For example, C:\Temp\XeroRT. Make sure that the file is in directory that your application users can read and write to. If the file does not exist, the driver will create it.</p> <p>This attribute is provided as a workaround when permissions prevent your application storing updated refresh tokens in the registry (Windows) or in /etc/odbc.ini (Linux and UNIX). The error generated in this situation contains "Failed to update Refresh Token", although you may need to turn on Easysoft ODBC-Xero driver logging and examine the log file to see this message. To retrieve the initial refresh token, use the <b>Authenticate and Obtain Token</b> button (Windows) or the xero_oauth2 program (Linux).</p>
Share Refresh	Enable this option if you want multiple data sources to Easysoft ODBC-Xero driver share the same Xero refresh token.
Proxy	<p>If you use a proxy server connect to Xero, use this attribute to specify this server's details. Use this format:</p> <p><code>http://address:port</code></p> <p>where <i>address</i> is the host name or IP address of the proxy server and <i>port</i> is the proxy server port. For example:</p> <p><a href="http://squid.example.com:8080">http://squid.example.com:8080</a></p>
Proxy User	<p>If your proxy server has authentication turned on, use this attribute to supply a user name that can connect to the proxy server.</p> <p>The Easysoft ODBC-Xero driver supports the Basic and Digest proxy authentication schemes.</p>
Proxy Password	The password for the proxy user.
Logging	Whether to turn on Easysoft ODBC-Xero driver logging. Normally, you'll only do this if so directed by the Easysoft support team.

## 22 Connection attributes

Name	Value
Log File	<p>The file name and path of the file you want the driver to write log information to. For example:</p> <p>C:\Windows\Temp\Easysoft.log</p> <p>If the file doesn't exist, the Easysoft ODBC-Xero driver creates it.</p>
Local Filtering	<p>When turned on, the Easysoft ODBC-Xero driver increases its preferred batch size for queries, which alters the number of Xero API calls it makes. Xero may create batches that are larger or smaller than the requested size to maximise performance.</p> <p>There is no one correct setting for <b>Local Filtering</b>, it depends on how your application executes its queries. You may find that this attribute increases query performance for one application and decreases performance for another.</p> <p>By default, <b>Local Filtering</b> is turned off.</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-Xero driver connection string for Windows:

```
"DRIVER={Easysoft XERO ODBC Driver};Organisation='Demo Company  
(UK)';Client_ID=abcdef1234;Client_Secret=12345xyz"
```

Here's an example Easysoft ODBC-Xero driver connection string for Linux and UNIX:

```
"DRIVER={Easysoft ODBC-Xero};Organisation='Demo Company  
(UK)';Client_ID=abcdef1234;Client_Secret=12345xyz"
```

For a list of the other attributes you can set in the connection string, refer to [this section](#).

## Logging

If you report an issue to us, we may ask you to turn on ODBC Driver Manager or Easysoft ODBC-Xero 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-Xero 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-Xero driver logging is specific to the Easysoft driver and is of most use when making a support call.

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

```
[XERO_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.

## Easysoft ODBC-Xero driver 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. Do one of the following:
  - If you configured a system data source, choose the **System DSN** tab.
  - Or-
  - If you configured a system data source, choose the **System DSN** tab.
4. Choose your Easysoft ODBC-Xero driver data source from the list, and then choose **Configure**.
5. In the Easysoft ODBC-Xero driver data source configuration dialog box, turn on **Driver Logging**.
6. Enter a log file name and path in the space provided. For example:

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

## Finding out what product version you have on Windows

If you have an issue with the Easysoft ODBC-Xero 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-Xero driver in the list.

The product version displays below.

## Client applications

How to work with Xero 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-Xero driver](#) on same computer as Access.
2. [Configure an ODBC data source](#).
3. Choose one of the following ways to work with your Xero 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-Xero 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-Xero 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-Xero driver](#) on same computer as Excel.
2. [Configure an ODBC data source](#).
3. Choose one of the following ways to work with your Xero data in Excel.

### Data Connection Wizard

1. Choose **Data > Get Data > From Other Sources > From ODBC**.
2. Choose your Easysoft ODBC-Xero 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 Xero 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 Xero 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-Xero driver data source from the list, and then choose **OK**.
6. Choose **Next**.
7. Choose how to import your Xero data and then choose **Finish**.
8. Choose **Close** to return the data to Excel.

## Microsoft Power BI

1. [Install the Easysoft ODBC-Xero 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 Xero 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.

<b>Note</b>	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 Xero data is now available to use in Power BI visualisations.

## Microsoft SQL Server

1. [Install the Easysoft ODBC-Xero 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 "Xero".
6. From the **Provider** list, choose **Microsoft OLE DB Provider for ODBC drivers**.
7. In the **Data source** box, enter the name of your Xero 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-Xero driver, it will not have the latest version of the System Path environment variable. The Easysoft ODBC-Xero 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-Xero driver.
  - If you made a mistake when specifying the Easysoft ODBC-Xero 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-Xero 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 = 'Xero'
```

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

```
SELECT * FROM [Xero]..DBO.MyTable
```

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([Xero], 'SELECT * FROM MyTable')
```

```
-- 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 MyTable (MyCol1, MyCol2, MyCol3, MyCol4, MyCol5)
VALUES ('MyValue1' , 'MyValue2' , 'MyValue3' , 'MyValue4' ,
'MyValue5'))
AT Xero

UPDATE OPENQUERY ([Xero], 'SELECT MyCol1 FROM MyTable WHERE MyCol1 =
'MyValue1''') SET MyCol1='MyNewValue'
DELETE OPENQUERY (Xero, 'SELECT MyCol1 FROM MyTable WHERE MyCol1 =
'MyValue1''')
```

SQL Server sends pass-through queries as uninterpreted query strings to the Xero. 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-Xero driver](#) on same computer as Oracle.
2. [Configure an ODBC data source](#).
3. Follow the instructions for your Oracle platform.

## Connecting Xero 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 initxero.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 = "Xero"
```

Replace Xero with the name of your Easysoft ODBC-Xero 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=xero)
(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:

```
xero =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = oracle_host)(PORT = 1521))
(CONNECT_DATA =
(SID = xero)
)
(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 Xero. For example:

```
CREATE PUBLIC DATABASE LINK xeroLink
CONNECT TO "dbuser" IDENTIFIED BY "dbpassword"
USING 'xero';
```

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

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

```
SELECT "MyCol1" FROM "MyTable"@xeroLink;

DECLARE
    num_rows integer;
BEGIN
    num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@xeroLink
('INSERT INTO MyTable (MyCol1, MyCol2, MyCol3, MyCol4, MyCol5) VALUES
('MyValue1'', 'MyValue2'', 'MyValue3'', 'MyValue4'', 'MyValue5''));
END;
/

DECLARE
    num_rows integer;
BEGIN
    num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@xeroLink
('UPDATE "MyTable" SET "MyCol1" = 'MyNewValue' WHERE "MyCol1" = 'MyValue1'');
END;
/

DECLARE
    num_rows integer;
BEGIN
    num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@xeroLink
('DELETE from "MyTable" WHERE MyCol1 = 'MyValue1'');
END;
/
```

## Notes

- If you have problems connecting to Xero 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 initXero.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 Xero 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 initxero.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 = "Xero"
```

Replace Xero with the name of your Easysoft ODBC-Xero 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=xero)
(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:

```
xero =
(DESCRIPTION =
(AADDRESS = (PROTOCOL = TCP)(HOST = oracle_host)(PORT = 1521))
(CONNECT_DATA =
(SID = xero)
)
(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 Xero. For example:

```
CREATE PUBLIC DATABASE LINK xeroLink
CONNECT TO "dbuser" IDENTIFIED BY "dbpassword"
USING 'xero';
```

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

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

```
SELECT "MyCol1" FROM "MyTable"@xeroLink;

DECLARE
    num_rows integer;
BEGIN
    num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@xeroLink
('INSERT INTO MyTable (MyCol1, MyCol2, MyCol3, MyCol4, MyCol5) VALUES
('MyValue1'', 'MyValue2'', 'MyValue3'', 'MyValue4'', 'MyValue5''));
END;
/

DECLARE
    num_rows integer;
BEGIN
    num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@xeroLink
('UPDATE "MyTable" SET "MyCol1" = 'MyNewValue' WHERE "MyCol1" = 'MyValue1'');
END;
/

DECLARE
    num_rows integer;
BEGIN
    num_rows:=DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@xeroLink
('DELETE from "MyTable" WHERE MyCol1 = 'MyValue1'');
END;
/
```

## Notes

- If you have problems connecting to Xero 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 initXero.ora and then start or restart the Oracle listener. If the trace directory does not exist, create it.

## LibreOffice

1. [Install the Easysoft ODBC-Xero 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-Xero 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 Xero 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=Xero")
    if err != nil {
        log.Fatal(err)
    }

    var (
        name string
    )

    rows, err := db.Query("SELECT MyCol1 FROM MyTable")
    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-Xero 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 Xero data:

```
const odbc = require('odbc');
// Replace Xero with the name of your Easysoft ODBC-Xero driver
// data source.
const connection = odbc.connect('DSN=Xero', (error, connection) => {
  connection.query('SELECT MyCol1 FROM MyTable', (error, result) => {
    if (error) { console.error(error) }
    console.log(result);
  });
});
```

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

```
const odbc = require('odbc');
const connection = odbc.connect('DSN=Xero', (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=Xero', (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 Xero data:

```
const odbc = require("odbc");
const connection = odbc.connect("DSN=Xero", (error, connection) => {
  connection.query("INSERT INTO
MyTable (
  MyCol1,
  MyCol2,
  MyCol3,
  MyCol4,
  MyCol5
```



```
)
VALUES
(
  'MyValue1',
  'MyValue2',
  'MyValue3',
  'MyValue4',
  'MyValue5'
)", (error, result) => {
  if (error) { console.error(error) }
  console.log(result);
});
});

const odbc = require("odbc");
const connection = odbc.connect("DSN=Xero", (error, connection) => {
  connection.query("UPDATE MyTable SET MyCol1 = 'MyNewValue' WHERE MyCol1 = 'MyValue1'", (error, result) => {
    if (error) { console.error(error) }
    console.log(result);
  });
});

const odbc = require("odbc");
const connection = odbc.connect("DSN=Xero", (error, connection) => {
  connection.query("DELETE FROM MyTable WHERE MyCol1 = 'MyValue1'", (error, result) => {
    if (error) { console.error(error) }
    console.log(result);
  });
});
});
```

## Perl

1. [Install the Easysoft ODBC-Xero 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-Xero driver to connect to Xero.

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

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

my $sql = "SELECT MyCol1 FROM MyTable";

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-Xero driver data source:

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

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:Xero');

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 Xero data:

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

my $sth = $dbh->prepare(q/INSERT INTO MyTable (MyCol1, MyCol2, MyCol3, MyCol4,
MyCol5) VALUES (?, ?, ?, ?, ?)/)
    or die "Can't prepare statement: $DBI::errstr";

$sth->execute('MyValue1', 'MyValue2', 'MyValue3', 'MyValue4', 'MyValue5');

$dbh->disconnect if ($dbh);

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

my $sth = $dbh->prepare('UPDATE MyTable SET MyCol1 = \'MyNewValue\' WHERE MyCol1
= ?')
    or die "Can't prepare statement: $DBI::errstr";

$sth->execute('MyValue1');

$dbh->disconnect if ($dbh);

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

my $sth = $dbh->prepare('DELETE FROM MyTable WHERE MyCol1 = ?')
    or die "Can't prepare statement: $DBI::errstr";
```

```
$sth->execute('MyValue1');  
  
$dbh->disconnect if ($dbh);
```

### Further information

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

## PHP

1. [Install the Easysoft ODBC-Xero 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 Xero data:

```
<?php
// Replace Xero with the name of your Easysoft ODBC-Xero driver data source.
// If your database requires a user name and password, supply them in the
odbc_connect_call.
$con = odbc_connect("Xero", "", "");
$stmt = odbc_exec($con, "SELECT * FROM MyTable");
// You may need to change the capitalisation of MyCol1 to all upper case or all
lower case.
while ($row = odbc_fetch_array($stmt)) {
    echo "MyCol1 = ", $row["MyCol1"], "\n";
}
odbc_close($con);
?>
```

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

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

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

```
<?php
$conx = odbc_connect("Xero", "", "");
$stmt = odbc_prepare($conx, "INSERT INTO MyTable (MyCol1, MyCol2, MyCol3,
MyCol4, MyCol5) VALUES (?, ?, ?, ?, ?)");
$success = odbc_execute($stmt, array('MyValue1', 'MyValue2', 'MyValue3',
```

```
'MyValue4', 'MyValue5')));
    odbc_close($cnx);
?>

<?php
    $cnx = odbc_connect("Xero", "", "");
    $stmt = odbc_prepare($cnx, "UPDATE MyTable SET MyCol1 = 'MyNewValue' WHERE
MyCol1 = ?");
    $success = odbc_execute($stmt, array('MyValue1'));
    odbc_close($cnx);
?>

<?php
    $cnx = odbc_connect("Xero", "", "");
    $stmt = odbc_prepare($cnx, "DELETE FROM MyTable WHERE MyCol1 = ?");
    $success = odbc_execute($stmt, array('MyValue1'));
    odbc_close($cnx);
?>
```

### Further information

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

# Python

1. [Install the Easysoft ODBC-Xero 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 Xero data:

```
import pyodbc

# Replace Xero with the name of your Easysoft ODBC-Xero driver data source.
cnxn = pyodbc.connect("DSN=Xero")
cursor = cnxn.cursor()
sql = "SELECT MyCol1 FROM MyTable"
cursor.execute(sql)
rows = cursor.fetchall()
# You may need to change the capitalisation of MyCol1 to all upper case or all
lower case.
for row in rows:
    print(row.MyCol1)
exit()
```

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

```
import pyodbc

# Replace Xero with the name of your Easysoft ODBC-Xero driver data source.
cnxn = pyodbc.connect("DSN=Xero")
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 Xero with the name of your Easysoft ODBC-Xero driver data source.
```

```
cnxn = pyodbc.connect("DSN=Xero")
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 Xero data:

```
import pyodbc

cnxn = pyodbc.connect("DSN=Xero")
cursor = cnxn.cursor()
sql = "INSERT INTO MyTable (MyCol1, MyCol2, MyCol3, MyCol4, MyCol5) VALUES (?, ?, ?, ?, ?)"
cursor.execute(sql, 'MyValue1', 'MyValue2', 'MyValue3', 'MyValue4', 'MyValue5')
cursor.commit()
exit()
```

```
import pyodbc

cnxn = pyodbc.connect("DSN=Xero")
cursor = cnxn.cursor()
sql = "UPDATE MyTable SET MyCol1 = 'MyNewValue' WHERE MyCol1 = ?"
cursor.execute(sql, 'MyValue1')
cursor.commit()
exit()
```

```
import pyodbc

cnxn = pyodbc.connect("DSN=Xero")
cursor = cnxn.cursor()
sql = "DELETE FROM MyTable WHERE MyCol1 = ?"
cursor.execute(sql, 'MyValue1')
cursor.commit()
exit()
```

## Further information

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



## R

1. [Install the Easysoft ODBC-Xero 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 Xero data:

```
library("RODBC")
# Replace Xero with the name of your Easysoft ODBC-Xero driver data source.
ch <- odbcConnect("Xero")
sqlQuery(ch, paste("SELECT MyCol1 FROM MyTable"))
odbcClose(ch)
quit()
```

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

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

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

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

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

```
library("RODBC")
ch <- odbcConnect("Xero")
sqlQuery(ch, paste("INSERT INTO MyTable (MyCol1, MyCol2, MyCol3, MyCol4, MyCol5)
VALUES ('MyValue1', 'MyValue2', 'MyValue3', 'MyValue4', 'MyValue5')"))
odbcClose(ch)
quit()

library("RODBC")
```

```
ch <- odbcConnect("Xero")
sqlQuery(ch, paste("UPDATE MyTable SET MyCol1 = 'MyNewValue' WHERE MyCol1 =
'MyValue1'"))
odbcClose(ch)
quit()

library("RODBC")
ch <- odbcConnect("Xero")
sqlQuery(ch, paste("DELETE FROM MyTable WHERE MyCol1 = 'MyValue1'"))
odbcClose(ch)
quit()
```

---

## About the Easysoft ODBC-Xero driver

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

In this section:

- [ODBC API and scalar functions](#)
- [Data type mapping](#)
- [SQL support](#)

## ODBC API and scalar functions

### API functions

Use this table to find out what ODBC API functions the Easysoft ODBC-Xero 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	Not supported
SQLColumns	Supported
SQLConnect	Supported
SQLCopyDesc	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
SQLFreeEnv	Supported
SQLFreeHandle	Supported
SQLFreeStmt	Supported

Function	Status
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
SQLSetParam	Supported
SQLSetPos	Supported
SQLSetScrollOptions	Supported

Function	Status
SQLSetStmtOption	Supported
SQLSetStmtAttr	Supported
SQLStatistics	Supported
SQLTablePrivileges	Not supported
SQLTables	Supported
SQLTransact	Supported

## Scalar functions

The Easysoft ODBC-Xero driver supports a number of scalar functions:

- [String functions](#)
- [Numeric functions](#)
- [Time, date, and interval functions](#)
- [System functions](#)
- [Conversion functions](#)

Use either the SQL-92 or ODBC syntax with scalar functions. For example:

```

SELECT
    Invoice_Id,
    Customer_Name,
    EXTRACT(YEAR FROM Due_Date) as "Year"
FROM
    Invoice

SELECT
    Invoice_Id,
    Customer_Name,
    {fn EXTRACT(YEAR FROM Due_Date)} as "Year"
FROM
    Invoice

```

## String functions

The Easysoft ODBC-Xero driver supports these [string](#) functions:

- `ASCII(string_exp)`
- `BIT_LENGTH(string_exp)`
- `CHAR(code)`
- `CHAR_LENGTH(string_exp)`
- `CHARACTER_LENGTH(string_exp)`
- `CONCAT(string_exp1, string_exp2)`
- `DIFFERENCE(string_exp1, string_exp2)`
- `INSERT(string_exp1, start, length, string_exp2)`
- `LCASE(string_exp)`
- `LEFT(string_exp, count)`
- `LENGTH(string_exp)`
- `LOCATE(string_exp1, string_exp2[start])`
- `LTRIM(string_exp)`

- OCTET\_LENGTH(*string\_exp*)
- POSITION(*char\_exp* IN *char\_exp*)
- REPEAT(*string\_exp*, *count*)
- REPLACE(*string\_exp1*, *string\_exp2*, *string\_exp3*)
- RIGHT(*string\_exp*, *count*)
- RTRIM(*string\_exp*)
- SOUNDEX(*string\_exp*)
- SPACE(*count*)
- UCASE(*string\_exp*)

## Numeric functions

The Easysoft ODBC-Xero driver supports these [numeric](#) functions:

- ABS(*numeric\_exp*)
- ACOS(*float\_exp*)
- ASIN(*float\_exp*)
- ATAN(*float\_exp*)
- CEILING(*numeric\_exp*)
- COS(*float\_exp*)
- COT(*float\_exp*)
- DEGREES(*numeric\_exp*)
- EXP(*float\_exp*)
- FLOOR(*numeric\_exp*)
- LOG(*float\_exp*)
- LOG10(*float\_exp*)
- MOD(*integer\_exp1*, *integer\_exp2*)
- PI()
- POWER(*numeric\_exp*, *integer\_exp*)
- RADIANS(*numeric\_exp*)
- RAND([*integer\_exp*])
- ROUND(*numeric\_exp*, *integer\_exp*)
- SIGN(*numeric\_exp*)
- SIN(*float\_exp*)
- SQRT(*float\_exp*)
- TAN(*float\_exp*)
- TRUNCATE(*numeric\_exp*, *integer\_exp*)

## Time, date, and interval functions

The Easysoft ODBC-Xero driver supports these [time, date, and interval](#) functions:

- CURRENT\_DATE()
- CURRENT\_TIME[(*time-precision*)]
- CURRENT\_TIMESTAMP[(*timestamp-precision*)]
- DAYNAME(*date\_exp*)
- DAYOFMONTH(*date\_exp*)
- DAYOFWEEK(*date\_exp*)
- DAYOFYEAR(*date\_exp*)
- EXTRACT(*extract-field* FROM *extract-source*)
- HOUR(*time\_exp*)
- MINUTE(*time\_exp*)
- MONTH(*date\_exp*)
- MONTHNAME(*date\_exp*)

- `NOW()`
- `QUARTER(date_exp)`
- `SECOND(time_exp)`
- `TIMESTAMPADD(interval, integer_exp, timestamp_exp)`
- `TIMESTAMPDIFF(interval, timestamp_exp1, timestamp_exp2)`
- `WEEK(date_exp)`
- `YEAR(date_exp)`

## System functions

The Easysoft ODBC-Xero driver supports these [system](#) functions:

- `DATABASE()`
- `USER()`

## Conversion functions

The Easysoft ODBC-Xero driver supports both the [SQL-92 CAST](#) function and the [ODBC CONVERT](#) function for conversion between compatible data types.



## Data type mapping

The Easysoft ODBC-Xero driver maps Xero data types to ODBC data types in this way:

Xero data type	ODBC data type
LONG NATIONAL CHARACTER VARYING	SQL_WLONGVARCHAR
NATIONAL CHARACTER VARYING	SQL_WVARCHAR
NATIONAL CHARACTER	SQL_WCHAR
BIT	SQL_BIT
TINYINT	SQL_TINYINT
BIGINT	SQL_BIGINT
LONG VARBINARY	SQL_LONGVARBINARY
VARBINARY	`SQL_VARBINARY`
BINARY	SQL_BINARY
LONG CHARACTER VARYING	SQL_LONGVARCHAR
CHARACTER	SQL_CHAR
NUMERIC	SQL_NUMERIC
DECIMAL	SQL_DECIMAL
INTEGER	SQL_INTEGER
INTEGER AUTONUMBER	SQL_INTEGER
SMALLINT	SQL_SMALLINT
FLOAT	SQL_FLOAT
REAL	SQL_REAL
DOUBLE PRECISION	SQL_DOUBLE
CHARACTER VARYING	SQL_VARCHAR
DATE	SQL_TYPE_DATE
TIME	SQL_TYPE_TIME
TIMESTAMP	SQL_TYPE_TIMESTAMP
INTERVAL YEAR	SQL_INTERVAL_YEAR
INTERVAL MONTH	SQL_INTERVAL_MONTH
INTERVAL DAY	SQL_INTERVAL_DAY
INTERVAL HOUR	SQL_INTERVAL_HOUR
INTERVAL MINUTE	SQL_INTERVAL_MINUTE
INTERVAL SECOND	SQL_INTERVAL_SECOND
INTERVAL DAY TO HOUR	SQL_INTERVAL_DAY_TO_HOUR
INTERVAL DAY TO MINUTE	SQL_INTERVAL_DAY_TO_MINUTE
INTERVAL DAY TO SECOND	SQL_INTERVAL_DAY_TO_SECOND
INTERVAL HOUR TO MINUTE	SQL_INTERVAL_HOUR_TO_MINUTE

Xero data type	ODBC data type
INTERVAL HOUR TO SECOND	SQL_INTERVAL_HOUR_TO_SECOND
INTERVAL MINUTE TO SECOND	SQL_INTERVAL_MINUTE_TO_SECOND

## 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-Xero driver is installed.
3. Double-click **odbcte32.exe**.
4. Select **Con > Full Connect**.
5. Choose your Easysoft ODBC-Xero 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**.

## SQL support

The Easysoft ODBC-Xero driver supports these SQL statements, clauses, and operators:

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

# Index

## A

### Access

- importing Xero data, [29](#)
- linking Xero data, [29](#)

## B

### Base

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

## C

- Client ID attribute, [21](#)
- Client Secret attribute, [21](#)
- connecting to Xero, [17](#)
- connection string attributes, [23](#)

## D

- data source attributes, [20](#)
- data sources
  - removing
    - on Windows, [16](#)
- data type mapping, [57](#)
- Description attribute, [21](#)
- DG4ODBC
  - working with Xero data, [34](#)
- DSN attribute, [21](#)
- DSN-less connections, [23](#)

## E

- Easysoft ODBC-Xero driver
  - adding ODBC data sources
    - on Linux or UNIX, [17](#)
    - on Windows, [18](#)
  - connecting to Xero, [17](#)
  - data source attributes, [20](#)
  - data type mapping, [57](#)
  - DSN-less connections, [23](#)
  - installing
    - on Linux or UNIX, [5](#)
    - on Windows, [14](#)
  - logging, [24](#)
  - ODBC API support, [52](#)
  - scalar function support, [54](#)
  - SQL support, [59](#)
  - uninstalling
    - on Linux or UNIX, [13](#)
    - on Windows, [16](#)

## Excel

- importing Xero data with Data Connection Wizard, [30](#)
- importing Xero data with PowerPivot, [30](#)
- importing Xero data with Query, [30](#)

## G

### Go

- working with Xero data, [39](#)

## I

- installing the Easysoft ODBC-Xero driver, [5](#)

## L

### LibreOffice

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

### linked server

- working with Xero data, [32](#)

### Local Filtering attribute, [22](#)

### Log File attribute, [22](#)

### log files, [24](#)

### Logging attribute, [21](#)

## N

### Node.js

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

## O

### OAuthfile attribute, [21](#)

### ODBC API function support, [52](#)

### ODBC connection string attributes, [23](#)

### ODBC data sources

- adding
  - on Linux or UNIX, [17](#)
  - on Windows, [18](#)
- removing
  - on Windows, [16](#)

### Oracle

- working with Xero data, [34](#)

### Organisation attribute, [21](#)

## P

### Perl

- working with Xero data, [42](#)

### PHP

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

### Power BI

- importing Xero data, [31](#)

### PowerPivot

- importing Xero data, [30](#)

### Proxy attribute, [21](#)

### Proxy Password attribute, [21](#)

### Proxy User attribute, [21](#)

### Python

- working with Xero data, [47](#)

## R

### R

- working with Xero data, [49](#)

## S

scalar function support, [54](#)

Share Refresh attribute, [21](#)

## SQL Server

working with Xero data, [32](#)

SQL support, [59](#)

## T

trace files, [24](#)

## U

### uninstalling

on Linux or UNIX, [13](#)

on Windows, [16](#)