

# **Easysoft ODBC-ODBC Bridge User's Guide**

This manual documents version 2.6.n of the Easysoft ODBC-ODBC Bridge.

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
Overview . . . . .	5
Installing the Easysoft ODBC-ODBC Bridge . . . . .	6
Installing on Windows . . . . .	6
Uninstalling on Windows . . . . .	8
Installing on Linux or UNIX . . . . .	8
Uninstalling on Linux or UNIX . . . . .	18
Configuring the Easysoft ODBC-ODBC Bridge . . . . .	21
The Statistics page . . . . .	21
The Configuration page . . . . .	23
The Security page . . . . .	29
The Information page . . . . .	30
Configuring the Easysoft ODBC-ODBC Bridge server on Windows . . . . .	30
Configuring the Easysoft ODBC-ODBC Bridge server on Linux and UNIX . . . . .	33
Connecting to your ODBC data source . . . . .	36
Configure an ODBC data source for your target database . . . . .	36
Configure an Easysoft ODBC-ODBC Bridge client data source . . . . .	36
Connection attributes . . . . .	38
Client support for fallback Easysoft ODBC-ODBC Bridge servers . . . . .	43
Testing the Easysoft ODBC-ODBC Bridge server . . . . .	46
Worked examples . . . . .	47
Tracing . . . . .	53
Reporting and statistics . . . . .	54
Introduction . . . . .	54
Changing the refresh frequency . . . . .	55
DSN statistics . . . . .	55
Host statistics . . . . .	56
Termination statistics . . . . .	56
Log of failing SQL . . . . .	56
Auditing . . . . .	57
Event logging on Windows . . . . .	59
Browsing system data sources in the web administrator . . . . .	60
Viewing data sources, tables, columns, and data . . . . .	60
The Licenses page . . . . .	60
About the Easysoft ODBC-ODBC Bridge . . . . .	61
ODBC versions supported . . . . .	62
Unsupported ODBC 3.5 functionality . . . . .	63
Modifications to the ODBC API . . . . .	64
SQLBrowseConnect . . . . .	64
Understanding ODBC diagnostic messages . . . . .	65
Implementing ODBC diagnostics . . . . .	66
Status returns . . . . .	66
Obtaining diagnostics . . . . .	66
Diagnostic fields . . . . .	68
Index . . . . .	69

## Getting started

- [Overview](#)
- [Installing the Easysoft ODBC-ODBC Bridge](#)
- [Configuring the Easysoft ODBC-ODBC Bridge](#)
- [Connecting to your ODBC data source](#)
- [Testing the Easysoft ODBC-ODBC Bridge server](#)
- [Tracing](#)
- [Reporting and statistics](#)

---

## Overview

The Easysoft ODBC-ODBC Bridge lets applications that support ODBC connect to a datastore that's on a different platform or has a different architecture. For example, an application on Linux needs to connect to a database for which there's only a Windows ODBC driver; a 64-bit application needs to connect to a database for which there's only a 32-bit ODBC driver.

# Installing the Easysoft ODBC-ODBC Bridge

The Easysoft ODBC-ODBC Bridge installer lets you install both the client and server components of the product.

Install the Easysoft ODBC-ODBC Bridge client on the same computer as your application. Install the Easysoft ODBC-ODBC Bridge server on the same computer as your target ODBC driver. If the application and ODBC driver are located on the same computer, install both the client and server on this computer.

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

## Installing on Windows

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

1. [Download the Easysoft ODBC-ODBC Bridge 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-ODBC Bridge 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-ODBC Bridge 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-ODBC Bridge 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-ODBC Bridge. 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-ODBC Bridge:

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-ODBC Bridge, the installation is complete.

## Repairing the installation

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

## Uninstalling on Windows

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

### Removing the Easysoft ODBC-ODBC Bridge

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

<b>Note</b>	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.
-------------	---

## Installing on Linux or UNIX

The installation can be done by anyone with root access.

1. [Download the Easysoft ODBC-ODBC Bridge 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 ODBC-ODBC Bridge 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](#)
- [Testing the Easysoft ODBC-ODBC Bridge server](#)
- [Testing the Easysoft ODBC-ODBC Bridge client](#)
- [Setting dynamic linker search paths](#)
- [Post installation steps for non-root installations](#)

## Preinstallation requirements

To install the Easysoft ODBC-ODBC Bridge 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-ODBC Bridge in /usr/local/easysoft.
  - Install the Easysoft ODBC-ODBC Bridge 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-ODBC Bridge 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-ODBC Bridge 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-ODBC Bridge 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-ODBC Bridge.
- The unixODBC Driver Manager.

You need an ODBC Driver Manager to use the Easysoft ODBC-ODBC Bridge 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-ODBC Bridge 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-ODBC Bridge 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-odbc-bridge-2.6.5-linux-x86-64.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-odbc-bridge-2.6.5-linux-x86-64
```

## 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-ODBC Bridge and data sources during the installation.
- Most applications and interfaces that support ODBC are compatible with unixODBC. The Easysoft ODBC-ODBC Bridge 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-ODBC Bridge 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-ODBC Bridge distribution is not built with the default options in unixODBC's configure line.

Option	Description
<code>--prefix=/etc</code>	This means the default SYSTEM <code>odbc.ini</code> file where SYSTEM data sources are located is <code>/etc/odbc.ini</code> .
<code>--enable-drivers=no</code>	This means other ODBC drivers that come with unixODBC are not installed.
<code>--enable-iconv=no</code>	This means unixODBC does not look for <code>libiconv</code> . Warnings about not finding an <code>iconv</code> library were confusing our customers.
<code>--enable-stats=no</code>	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-ODBC Bridge installation script:

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

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

```
Would you like to request a Easysoft ODBC-ODBC Bridge 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-ODBC Bridge), 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-ODBC Bridge 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-ODBC Bridge 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-ODBC Bridge 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.
```

## Testing the Easysoft ODBC-ODBC Bridge server

The Easysoft ODBC-ODBC Bridge client distribution includes the `oobping` program, which lets you test the connection to an Easysoft ODBC-ODBC Bridge server. There are two versions of `oobping` in `/usr/local/easysoft/bin`. `oobpings` is statically linked and should not require any libraries other than `libc` (this means you don't have to set any dynamic linker paths). `oobpingd` is a dynamically linked `oobping` (therefore smaller in size) that requires additional libraries in `/usr/local/easysoft/lib` and `/usr/local/easysoft/oob/client`. Generally speaking, you should use `oobpings`.

For full instructions on using `oobping` refer to [Testing the Easysoft ODBC-ODBC Bridge server](#). To display a summary of usage options, run `oobpings` with no arguments.

A short summary is:

1. `cd /usr/local/easysoft/bin`
2. To test an Easysoft ODBC-ODBC Bridge server is installed, running, and listening on a specific port use:

```
./oobpings -h my_server -p 8888
```

The `-p 8888` is optional, as, by default, the Easysoft ODBC-ODBC Bridge server uses port 8888.

3. To test operating system authentication:

```
./oobpings -h my_server -p 8888 -u my_user -p p455w0rd
```

4. To do a full test of an Easysoft ODBC-ODBC Bridge client DSN defined in your `odbc.ini` file use:

```
./oobpings -d "DSN=my_odbc_dsn;"
```



5. You can also use oobping to test DSN-less connections by putting all the connection attributes in the -d argument. For example:

```
./oobpings -d
'LogonUser=my_os_user;LogonAuth=p455w0rd;TargetDSN=pubs;UID=my_db_user;PWD=p455w0rd;serverPort=my_server:8888;'
```

## Testing the Easysoft ODBC-ODBC Bridge client

The Easysoft ODBC-ODBC Bridge client is a shared object containing entry points for the full ODBC 3.5 API, most of the ODBC 2.0 API (and a few others as well).

The Easysoft ODBC-ODBC Bridge client only becomes functional when an application is linked with it directly or through an ODBC Driver Manager. You can use unixODBC's command line isql program to test the Easysoft ODBC-ODBC Bridge client and send SQL to your database. Easysoft ODBC-ODBC Bridge comes with an isql wrapper program called isql.sh that sets up the dynamic linker before running isql.

The /usr/local/easysoft/oob/examples directory contains a number of test applications and make files that let you try the Easysoft ODBC-ODBC Bridge out.

Refer to the INSTALL.txt file in the examples directory.

## Setting dynamic linker search paths

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

On operating systems where the dynamic linker has a file specifying locations for shared objects (Linux, FreeBSD), the installation attempts to add paths under the path you provided at the start of the install to the end of this list; no further action should be required.

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

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

This always works and overrides 2 below.

2. At build time, a run path is inserted into the executable or shared objects. On System V systems, Easysoft mostly distributes Easysoft ODBC-ODBC Bridge client shared objects with an embedded run path that the dynamic linker can use to locate Easysoft ODBC-ODBC Bridge client shared object dependencies.

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

Environment variable	Platform
LD_LIBRARY_PATH	System V based operating systems and SCO, Linux, Solaris, FreeBSD, Irix, QNX.
DYLD_LIBRARY_PATH	macOS
LIBPATH	AIX
SHLIB_PATH	HP-UX



Environment variable	Platform
LD_RUN_PATH	Many platforms use this in addition to those listed above.

To use the Easysoft ODBC-ODBC Bridge client ODBC driver you need to add:

```
<InstallDir>/easysoft/oob/client:<InstallDir>/easysoft/lib
```

where <InstallDir> is the directory in which you chose to install the Easysoft ODBC-ODBC Bridge. If you accepted the default location, this is /usr/local.

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

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/easysoft/oob/client:/usr/local/easysoft/lib
export LD_LIBRARY_PATH
```

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

If unixODBC has been installed then you will also need to add <InstallDir>/easysoft/unixODBC/lib to the dynamic linker search path.

## Uninstalling on Linux or UNIX

There is no automated way to remove the Easysoft ODBC-ODBC Bridge 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-ODBC Bridge 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-ODBC Bridge entry needs to be removed from the odbcinst.ini file. To check whether the Easysoft ODBC-ODBC Bridge is configured under unixODBC, use odbcinst -q -d. If the command output contains [OOB], uninstall the driver from unixODBC by using:

```
odbcinst -u -d -n OOB
```

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-ODBC Bridge 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 OOB.
2. Remove the install.info for the Easysoft ODBC-ODBC Bridge from the /usr/local/easysoft directory.
3. To remove the service entries, make a backup of the /etc/services configuration file.
4. Open /etc/services and search for a line similar to:

```
esoobserver 8888/tcp # Easysoft ODBC-ODBC Bridge
```

If more than one Easysoft ODBC-ODBC Bridge service has been created, there will be more than one line in the services file.

Each such line should have a comment referencing the Easysoft ODBC-ODBC Bridge.

esoobserver is the default name for an Easysoft ODBC-ODBC Bridge service and any additional Easysoft ODBC-ODBC Bridge services will display the alternative names given to them.

Note down the names of the services you remove at this stage, so that if there are problems you can look them up in your servicesbackup file and re-introduce them.

**Note** You may want to compare /etc/services with /etc/services.pre\_OOB (which is installed with the Easysoft ODBC-ODBC Bridge) for details of entries that require removal.

5. You must remove all services that were configured for use with the Easysoft ODBC-ODBC Bridge. Delete all lines pertaining to all Easysoft ODBC-ODBC Bridge servers and save the file.
6. To remove from inetd, make a backup of the /etc/inetd.conf configuration file.
7. Open /etc/inetd.conf in your editor. Look for a line in the file similar to the following:

```
esoobserver stream tcp nowait root /bin/sh /bin/sh  
/usr/local/easysoft/oob/server/SERVER
```

where `esoobserver` is the name as specified in the services file, so there should be one entry in `inetd.conf` for every entry in the services file.

**Note** You may want to compare `/etc/inetd.conf` with `/etc/inetd.conf.pre_OOB` (which is installed with the Easysoft ODBC-ODBC Bridge) for details of entries that require removal.

8. Remove the lines pertaining to all Easysoft ODBC-ODBC Bridge servers and save the file.
9. Use `ps` to find the Process ID (PID) of the `inetd` process and send it a hangup signal:

```
kill -HUP pid
```

10. To remove from `xinetd`, refer to these notes:

By default, `xinetd` uses the configuration file `/etc/xinetd.conf`, which can either contain the names of the services and what to do with them, or (on some Red Hat machines, for example) an `includedir` setting that points to a directory where those services are defined (one per file).

You should have been made aware of which method your machine uses from the original installation procedure.

In the former case, the Easysoft ODBC-ODBC Bridge install adds a service entry to `/etc/xinetd.conf` and in the latter case, it creates a new service file called `esoobserver` containing the Easysoft ODBC-ODBC Bridge service settings.

11. There are therefore two ways to delete the Easysoft ODBC-ODBC Bridge service from `xinetd`.

- Make a backup of the `/etc/xinetd.conf` configuration file.

Open `/etc/xinetd.conf` in your editor. Look for a line in the file similar to the following:

```
service esoobserver
```

where `esoobserver` is the name as specified in the services file, so there should be one entry in `xinetd.conf` for every entry in the services file.

-Or-

- Delete the `esoobserver` file in the directory to which the `includedir` setting in `/etc/xinetd.conf` points.

Remove all the lines referring to the Easysoft ODBC-ODBC Bridge server and save the file.

12. Use `ps` to find the Process ID (PID) of the `xinetd` process and send it a Hangup signal:

```
kill -USR1 pid
```

-Or-

```
kill -USR2 pid
```

13. To remove from the dynamic linker, you need to notify the dynamic linker that the shared objects are no longer available.

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

14. If you have the file `/etc/ld.so.conf` file, make a backup copy. For example:

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

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

```
installdir/easysoft/oob/client
```

16. If you're not using any other Easysoft software then you can remove the path to the common Easysoft shared objects:

```
installdir/easysoft/lib
```

17. If you're no longer using unixODBC then you can also remove the reference:

```
installdir/easysoft/unixODBC
```

18. Run `/sbin/ldconfig` so that the dynamic linker re-reads the file and will no longer search the removed paths.

## Configuring the Easysoft ODBC-ODBC Bridge

Use the web administrator to configure the Easysoft ODBC-ODBC Bridge.

To access the web administrator, got to:

`http://serverhost:8031/`

where *serverhost* is the host name computer on which the Easysoft ODBC-ODBC Bridge server is running.

The web administrator contains these pages:

- [Statistics](#)
- [Configuration](#)
- [Security](#)
- [Information](#)

### The Statistics page

The **Statistics** page displays runtime statistics for the latest run of the server and lets you access other pages.

This page contains the following fields:

Field	Description
Server up time	The time in days, hours, minutes, and seconds since the Easysoft ODBC-ODBC Bridge server was started.
Server CPU time (s)	<p>This field is only visible if the ShowProcessTime flag is turned on.</p> <p>One or more values will be shown:</p> <ul style="list-style-type: none"> <li>• If only one value is shown, it is the total CPU time consumed by the Easysoft ODBC-ODBC Bridge server.</li> <li>• If two times are shown, the first is user time and the second is kernel time.</li> </ul> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p><b>Note</b></p> <p>When the Easysoft ODBC-ODBC Bridge server is running multi-threaded, any of the CPU times shown include CPU time consumed by the ODBC driver manager, any ODBC drivers loaded and any child processes or threads. The process time is only updated when the Easysoft ODBC-ODBC Bridge server is idle. If the Easysoft ODBC-ODBC Bridge server is extremely busy servicing incoming connections, the process time will not be updated.</p> </div>
Total Connections	<p>The total number of connections (or attempted connections) to the Easysoft ODBC-ODBC Bridge server.</p> <p>This includes connections dropped due to no license or insufficient license slots, port scanners, or anyone using telnet to access the Easysoft ODBC-ODBC Bridge server port.</p>

Field	Description
Total Threads/Processes	<p>The total number of threads or processes that the Easysoft ODBC-ODBC Bridge server has created during its execution.</p> <p>Connections denied access because of an access control rule or MaxThreadCount or MaxClientConnect being exceeded are not included, because the Easysoft ODBC-ODBC Bridge server does not start a thread or process for these.</p>
Active Threads/Processes	<p>The total number of active threads or processes the Easysoft ODBC-ODBC Bridge server has created to handle connections.</p> <p>This number may exceed the actual active count as the Easysoft ODBC-ODBC Bridge server only looks for exited threads and processes when five seconds has elapsed without any connections. (This is done to give preference to incoming connections.)</p> <p>Note that if MaxThreadCount or MaxClientConnect is set to anything other than 0, the Easysoft ODBC-ODBC Bridge server has to reap exited threads and processes every time a new connection arrives.</p> <p>Also note that this is not a limit and may therefore exceed your maximum licensed connection slots.</p>
Peak concurrent Threads/Processes	The highest value ever seen in the Active Threads/Processes count.
Last Connection time	The time the last connection occurred.
Last Disconnect time	The time the last disconnection occurred.
Number of different client hosts	<p>The number of different client machines that have connected to the Easysoft ODBC-ODBC Bridge server (where a client machine is identified by its IP address).</p> <p>Click on this link to get a list of IP addresses or machine names. Machine names are only displayed if you have ReverseLookup enabled.</p>

Field	Description
Changing the refresh frequency	<p>The Web Administrator uses a set of template files into which the dynamic data is inserted before sending it back to your browser.</p> <p>The template file for the <b>Statistics</b> screen is index.html, which is located in the admin subdirectory under wherever you installed the Easysoft ODBC-ODBC Bridge server.</p> <p>Open index.html in a text editor, and near the top you'll find:  meta http-equiv="refresh" content="60"; URL=/index.html</p> <p>Change 60 (the refresh time in seconds) to your new refresh time.</p> <div> <p><b>Note</b></p> <p>Note that setting the refresh time to a very low value will increase the workload on the Easysoft ODBC-ODBC Bridge server process which handles HTTP requests. As this may reduce the response time to the Easysoft ODBC-ODBC Bridge server thread, times much less than 60 seconds are not recommended.</p> </div>

## The Configuration page

To change settings in this page, use the user name that you specified for this purpose during Easysoft ODBC-ODBC Bridge setup. (Unless you chose to turn off web administrator authentication, in which case no user name is needed.)

This page contains the following fields:

Field	Description
Port	<p>The port on which the Easysoft ODBC-ODBC Bridge server listens for incoming client connections.</p> <p>The default port number is 8888, but it may be any port number not in use on your Easysoft ODBC-ODBC Bridge server machine. If you change the port, you'll need to restart the Easysoft ODBC-ODBC Bridge server before the change comes into effect.</p>
IPAddress	<p>The IP address the Easysoft ODBC-ODBC Bridge server listens on. This is only relevant if you have multiple network interface cards (NICs) and only want the Easysoft ODBC-ODBC Bridge server to listen on one. The default is for the server to bind to port 8888 on INADDR_ANY, which means it is listening on all local interfaces.</p>

Field	Description
HTTPPort	<p>The port on which the Easysoft ODBC-ODBC Bridge server listens for HTTP (web administrator) requests.</p> <p>The default port number is 8031, but it may be any port number not in use on your Easysoft ODBC-ODBC Bridge server machine.</p> <p>If the Flags bitmask has the second bit set (HTTP_Server), the Easysoft ODBC-ODBC Bridge server starts listening on the specified port for HTTP requests in addition to acting in its normal role, serving client requests.</p> <p>On Windows, the web administrator is a separate service that's started automatically by the Windows <b>Service Manager</b>. The web administrator listens on the specified port for HTTP requests. You use the URL <code>http://computer-name: HTTPPort</code> where <i>computer-name</i> is the name (or IP address) of the computer running the Easysoft ODBC-ODBC Bridge server and <i>HTTPPort</i> is the port number to communicate with the Easysoft ODBC-ODBC Bridge server from your browser.</p> <p>On Windows, the web administrator needs to be restarted in <b>Service Manager</b> for the new port to take effect.</p>
Timeout	<p>The inactivity timeout in seconds (the default is 7200, two hours). (Currently, this option is only applicable to the Windows Easysoft ODBC-ODBC Bridge server.)</p> <p>The Easysoft ODBC-ODBC Bridge server starts a new thread (or process) for each client that connects and if there has been no communication in Timeout seconds, the thread or process exits.</p> <p>This ensures clients that fail to close down properly don't cause increasing resource usage on the server.</p> <p>To turn off the timeout, set it to 0. Turn off this field if you're using persistent connections from PHP, mod_perl, and so on.</p>
RetryCount	The number of times the Easysoft ODBC-ODBC Bridge server attempts to create a thread or process to handle a connection, or the number of times the server attempts to obtain a license slot for a new connection.
RetryPause	The time in seconds between each retry attempt.
HTTPAdmin	<p>The user name of the person allowed to make changes to the Easysoft ODBC-ODBC Bridge server by using the web administrator.</p> <p>This must be a valid user name in the operating system the server is running on. If set to the disabled, web server authentication is not required.</p> <p>The value is case-sensitive.</p>



Field	Description
CertFile	The path to the file containing the certificate used to secure the connection between Easysoft ODBC-ODBC Bridge client and server. For example, /home/myuser.cert.pem. You need to restart the Easysoft ODBC-ODBC Bridge server before this setting takes effect.
HTTPNetWorkAccess	The network or IP address of the clients that are allowed to use the web administrator. For example, 194.131.236.4 only allows one computer but if the netmask was 255.255.255.0 then setting 194.131.236 would allow anyone on that network. The default is all browser clients have access to the web administrator although they may be required to enter a username and password if HTTPAdmin is set to anything other than disabled.
MaxBookmarkSize	The size of the largest bookmark the Easysoft ODBC-ODBC Bridge can handle (it defaults to 32 bytes). ODBC 2.x uses fixed length bookmarks of 4 bytes. In ODBC 3.x, bookmarks are all variable in size. If you find an ODBC driver that needs more than 32 bytes for a bookmark, please contact the Easysoft support team, otherwise the default should be fine.
Path	The Easysoft ODBC-ODBC Bridge installation path.
Logging	<p>A bitmask telling the Easysoft ODBC-ODBC Bridge server what sorts of event to record in the log file.</p> <p>This should only be used as directed by Easysoft support and slows the Easysoft ODBC-ODBC Bridge down considerably if set. You may specify the number as decimal or hexadecimal. For example, 2047 or 0x7ff.</p>
LogDir	<p>The directory where Easysoft ODBC-ODBC Bridge log files are created.</p> <p>It defaults to <i>drive:\Program Files\Easysoft\Easysoft ODBC-ODBC Bridge\Logs\</i> on Windows and <i>/tmp</i> on Linux and UNIX.</p>
MaxThreadCount	<p>The maximum number of threads or processes the Easysoft ODBC-ODBC Bridge server allows at any one time.</p> <p>One thread or process is created for every ODBC connection. If MaxThreadCount is set to 0, there is no limit. The default is 100.</p> <p>Use this parameter to prevent too many simultaneous connections swamping your server.</p>
MaxClientConnect	The maximum number of concurrent connections from a single client (where a client is defined as the IP address of the computer where the Easysoft ODBC-ODBC Bridge client is running). If MaxClientConnect is set to 0, there is no limit. The default is 0. You can use this parameter to prevent users from swamping your server with ODBC connections.
NetRCVLOWAT	The size of the receive low-water mark in bytes.
NetRCVBUF	The size of the receive queue buffer for the socket in bytes.
NetSNDBUF	The size of the send queue buffer in bytes.
NetConnectRetryCount	This setting is deprecated.

Field	Description
ListenBacklog	Defines the maximum length the queue of pending connections may grow to. It is a value passed to the BSD sockets call listen(). If a connection request arrives when the queue is full, the client will probably get an error of ECONNREFUSED and the Easysoft ODBC-ODBC Bridge server will be totally unaware of the connection attempt. If this happens, the Easysoft ODBC-ODBC Bridge client will use ConnectAttempts to make multiple connection attempts. You should not usually need to change this value as the default is sufficient for all but the very busiest of Easysoft ODBC-ODBC Bridge servers and having to set a high value is an indication that the server computer is underpowered. In addition, on Windows operating systems, the backlog queue is in non-pageable system memory. For some versions of Windows, changing this value has no effect since it is hard-wired at 5.
Affinity	If the Easysoft ODBC-ODBC Bridge server is running on a multiple CPU Windows server computer, lets you bind the Easysoft ODBC-ODBC Bridge server to particular CPUs on your computer.

Field	Description
Flags	<p>A bitmask of operational flags.</p> <p>The bitmasks are split into check boxes, one for each bit:</p> <ul style="list-style-type: none"> <li>• <b>Authentication_Disabled (0x1)</b> If set, Easysoft ODBC-ODBC Bridge server authentication is turned off . Setting this parameter should be considered as a security risk. However, in controlled environments where you do not need to authenticate the connections this can save a considerable amount of time during connections. The default is off.</li> <li>• <b>HTTP_Server (0x2)</b> If set, the Easysoft ODBC-ODBC Bridge server listens for HTTP connections to the web administrator. The default is on.</li> <li>• <b>MultiProcess (0x4)</b> If set, the Easysoft ODBC-ODBC Bridge server starts a new process rather than a new thread for each incoming connection. Use this setting for an ODBC driver that's not thread-safe or is leaking memory. On Linux and UNIX, the MultiProcess flag can't be updated, as the server is not multi-threaded and therefore always starts new processes. The default is off.</li> <li>• <b>HideSensitive (0x10)</b> If set, the web administrator hides sensitive parameters on the <b>Configuration</b> page. The HTTPAdmin and Authentication_Disabled parameters are hidden when HideSensitive is set. All parameters are always shown in full on the <b>Change Configurable Parameters</b> page as this page is password protected. The default is off.</li> <li>• <b>ReverseLookup (0x20)</b> If set the Easysoft ODBC-ODBC Bridge server calls gethostbyaddr() on the connecting client's IP address to obtain the client's machine name. On machines where DNS is not set up properly, this can cause problems and in any case, adds time to the connection. ReverseLookup currently only affects the number of different clients shown on the <b>Statistics</b> page, where "unknown" will be displayed instead of the machine name for the machine names of connecting clients if ReverseLookup is off. The default is off.</li> <li>• <b>AuditODBCAccess (0x40)</b> If set the Easysoft ODBC-ODBC Bridge server audits all connections to a log file, which can be examined in the web administrator. The audit file is written to the LogDir directory as esoob_access.log. The default is off.</li> <li>• <b>ShowProcessTime (0x100)</b> If set user and kernel CPU times for the main Easysoft ODBC-ODBC Bridge server process are displayed on the <b>Statistics</b> page. Values are updated every five seconds. The default is off.</li> <li>• <b>AutoUpdateConfiguration (0x200)</b> This setting is deprecated.</li> <li>• <b>HTTPAuthAll (0x400)</b> If set, all pages in the web administrator are protected by HTTP authentication, and you'll be prompted for the HTTPAdmin username and password before any pages are returned to you. Note that setting HTTPAuthAll has no effect if HTTPAdmin is set to disabled.</li> <li>• <b>ConnectionPooling (0x800)</b> If set, the Easysoft ODBC-ODBC Bridge server automatically sets the SQL_ATTR_CONNECTION_POOLING attribute to SQL_CP_ONE_PER_DRIVER. This turns on connection pooling in the Driver Manager for the ODBC driver. Note that ConnectionPooling is meaningless when the Easysoft ODBC-ODBC Bridge server is running in MultiProcess mode.</li> </ul>

**Note**

On Windows, the Easysoft ODBC-ODBC Bridge server parameters in the **Configuration** page apply to both the 32-bit and 64-bit Easysoft ODBC-ODBC Bridge server, apart from Port. The Port setting defines the port on which the 32-bit Easysoft ODBC-ODBC Bridge server listens for incoming client connections. If you change the Port value, the 32-bit Easysoft ODBC-ODBC Bridge server will listen on the new port; the 64-bit server will continue to listen on the default port, 8888. (When the 32-bit and 64-bit Easysoft ODBC-ODBC Bridge servers are listening on different ports, it is possible for both servers to run at the same time.) If you do not want the 64-bit Easysoft ODBC-ODBC Bridge server to listen on the default port, you need to change the value of this registry key:  
 HKEY\_LOCAL\_COMPUTER\SOFTWARE\Wow6432Node{PRODUCT}\Configuration\System\Settings\Port64

## Support for multiple network interface cards

By default, the Easysoft ODBC-ODBC Bridge server listens on all network interfaces. If the server computer has network interface cards (NICs) on multiple networks, the Easysoft ODBC-ODBC Bridge server can be associated with a particular NIC by specifying an IP address.

This is necessary if the server is running on multiple networks and access to the Easysoft ODBC-ODBC Bridge server is only required from one network.

The **IPAddress** parameter on the **Configuration** page is usually an empty field, signifying that the Easysoft ODBC-ODBC Bridge server is listening on all local interfaces.

Change this field to the IP address of a particular NIC to restrict access to a particular network.

## Support for specific processors

On multiple processor Windows computers, you can allocate particular processors on which to run the Easysoft ODBC-ODBC Bridge server. This is known as process affinity.

If your server has multiple CPUs, the **Configuration** page shows each configured processor as CPU\_<n> (where <n> is a number between zero and the number of available processors minus one).

Initially, the Easysoft ODBC-ODBC Bridge server is available to run on all processors. Deselecting CPU\_<n> check boxes, changes the Easysoft ODBC-ODBC Bridge server affinity so that it's restricted to a subset of the configured processors.

Note that the Easysoft ODBC-ODBC Bridge server only sets process affinity when starting up, so any changes made will only affect the server after it has been restarted.

## Connection pooling

Automatic connection pooling in the Easysoft ODBC-ODBC Bridge server process can be turned on in the **Configuration** page.

If connection pooling is turned on, the Easysoft ODBC-ODBC Bridge server sets the SQL\_ATTR\_CONNECTION\_POOLING attribute to SQL\_CP\_ONE\_PER\_DRIVER to turn on connection pooling in the Driver Manager for the ODBC driver.

If connection pooling is turned on, connections through the ODBC driver to the database are held open by the Driver Manager for use the next time the application uses the same ODBC data source.

This is only of any real benefit if the Easysoft ODBC-ODBC Bridge server is running in multi-threaded mode (the default), as the pooled connections are per process.

Turning on connection pooling in the Easysoft ODBC-ODBC Bridge server can vastly reduce connection times, especially through ODBC drivers connecting to remote databases.

## Exporting server configurations

Easysoft ODBC-ODBC Bridge server configuration is stored in the registry (in Windows) or in a plain text file (on Linux and UNIX systems).

It is simple to safeguard this configuration or copy it to another Easysoft ODBC-ODBC Bridge server on Linux and UNIX, but normally more difficult on Windows.

In Windows, at the bottom of the **Configuration** page there is an **Export** button.

Clicking **Export** writes the entire Easysoft ODBC-ODBC Bridge server configuration out to the file *LogDir\oob.reg*. **LogDir** is an Easysoft ODBC-ODBC Bridge server parameter that's defined on the **Configuration** page.

This file is in RegEdit 4 format, so it can be copied to other servers where double-clicking on it will install that Easysoft ODBC-ODBC Bridge server configuration into the registry.

## The Security page

This page lets you view or change the computers that are allowed to connect to the Easysoft ODBC-ODBC Bridge.

Easysoft ODBC-ODBC Bridge security splits IP addresses up into two lists: allowed and denied. When a computer attempts to connect to the Easysoft ODBC-ODBC Bridge server, access is only granted if:

- The allowed list is empty and the IP address is not in the denied list.

-Or-

- The IP address is in the allowed list.

Addresses must be entered using the IP dot notation. Entries that consist of fewer than four fields represent all addresses that match the defined fields. For example:

163.141.23. (note the trailing dot) matches all IP addresses from:

163.141.23.0

to

163.141.23.255

## Access control per DSN

The Easysoft ODBC-ODBC Bridge provides these ways to control access to ODBC data sources:

- **Authentication** Clients must provide a valid user name password for the host operating system.
- **Access control lists** Restrict access to the data source by client IP.

The **DSN Access Control** section of the **Security** page lets you build access control lists that define which computers and users can access which DSNs.

The **DSN Allowed Access** list and **Denied Access** list define the hosts and users that may access a particular DSN.

The server consults the **Deny Access** list first, and access is denied if a client address or user name matches an entry in this list.

If there is no match for the client or user name in the **Denied Access** list, the server consults the **Allowed Access** list, and if the **Allowed Access** list is empty the client or user is allowed access.

If there are any entries in the **Allowed Access** list, the client or user is only allowed access if it matches an entry in this list.

The entries in either list must consist of either IP addresses or user names. IP addresses must be full dotted quad notation. For example, 194.131.236.4. Entries may be followed by an optional / and a network mask. For example, 192.168.5.0/255.255.255.0 specifies the private class C address 192.168.5. You can use the address/mask notation to specify any network address. If you omit the network mask it defaults to 255.255.255.255, that is, an exact match for the specified IP address.

Any user name added to either list must match the LogonUser attribute used by the Easysoft ODBC-ODBC Bridge client to log the Easysoft ODBC-ODBC Bridge server thread or process into that account before allowing access to a DSN.

<b>Note</b>	As a special case (to avoid having to name every user when adding an IP address to the DSN allow list) the deny list is checked for the current user, and allows access to the DSN if no matching entry is found.
-------------	---

## Securing the web administrator from clients outside your network

The **HTTPNetworkAccess** parameter can be used to restrict access to the web administrator by setting it to the network or IP address of clients that are to be allowed access.

The IP address may be followed by an optional / and a network mask. For example, 194.131.236.4 only allows one computer to access the web administrator. 194.131.236/255.255.255.0 allows anyone on that network.

Browser clients not permitted access receive an HTTP 403 Forbidden error. By default, all browser clients have access to the web administrator, although they may be required to enter a user name or password if **HTTPAdmin** is set to anything other than disabled.

## Preventing data sources from being browsed in the administrator

The Easysoft ODBC-ODBC Bridge is not only capable of displaying a list of system data sources, but can also allow those data sources to be browsed.

This is a powerful, but potentially dangerous feature, as anyone with a browser and access to the web administrator might be able to see data in your database.

Although you can protect all of the web administrator pages with HTTP authentication, the **AllowDBBrowse** parameter allows the user to define a more precise configuration.

By default **AllowDBBrowse** is not set, which means that data sources shown on the **Data Sources** page are listed but not browsable. If **AllowDBBrowse** is turned on, the data sources become hyperlinks allowing you to browse down through the DSNs, tables, columns, and data.

## The Information page

The page contains:

- A list of links to Easysoft support resources

## Configuring the Easysoft ODBC-ODBC Bridge server on Windows

The Easysoft ODBC-ODBC Bridge server is installed as a Windows service, configured to be started automatically by the Windows **Service Manager** when the computer boots.

<b>Note</b>	The Easysoft ODBC-ODBC Bridge server writes status, diagnostic, and security
-------------	--

information to the Windows application event log. To examine this information, use **Event Viewer** to examine the application event log.

The Easysoft ODBC-ODBC Bridge server runs with administrative privileges, allowing it to become the user specified in the Easysoft ODBC-ODBC Bridge client data source.

When a user connects, the Easysoft ODBC-ODBC Bridge server becomes the user specified in the LogonUser attribute so that it acts with the specified user's permissions.

You can configure the service like any other service. For instance, you can set it up not to start automatically on system boot.

You can also explicitly stop the service.

By default, only the 32-bit Easysoft ODBC-ODBC Bridge server is started automatically. (With the default configuration settings, both 32-bit and 64-bit Easysoft ODBC-ODBC Bridge servers listen for incoming client connections on port 8888, so it is not possible for both servers to run at the same time.) If you want to use the 64-bit Easysoft ODBC-ODBC Bridge server, you need to:

1. Stop the 32-bit Easysoft ODBC-ODBC Bridge server service (Easysoft ODBC-ODBC Bridge server).
2. Start the 64-bit Easysoft ODBC-ODBC Bridge server service (Easysoft ODBC-ODBC Bridge server x64).

To configure the 64-bit Easysoft ODBC-ODBC Bridge server service to start automatically when Windows starts:

1. In **Windows Services**, double-click Easysoft ODBC-ODBC Bridge server.
2. In the Easysoft ODBC-ODBC Bridge server **Properties** dialog box, choose **Stop**. In the **Startup** type list, choose **Manual**.

The 32-bit Easysoft ODBC-ODBC Bridge server service is no longer running and will not start when Windows starts.

3. In **Windows Services**, double-click Easysoft ODBC-ODBC Bridge server x64.
4. In the Easysoft ODBC-ODBC Bridge server **Properties x64** dialog box, choose **Start**. In the **Startup** type list, choose **Automatic**.

The 64-bit Easysoft ODBC-ODBC Bridge server service is now running and will start when Windows starts.

## Setting up recovery actions for the Easysoft ODBC-ODBC Bridge server

If you're running the Easysoft ODBC-ODBC Bridge server as a service in multi-threaded mode (the default) and an exception is caught by the Easysoft ODBC-ODBC Bridge server, it shuts down. The reason for this is that the exception could have occurred anywhere (most commonly, the ODBC driver you are using) and there is no way for the Easysoft ODBC-ODBC Bridge server to correct the problem and no way for it to protect the other threads in the service.

On later versions of Windows, it's possible to set the **Service Manager** to monitor services and restart them for you after a specified time:

1. In the **Services** dialog box, double-click the Easysoft ODBC-ODBC Bridge server entry.
2. In the **Service Configuration** dialog box, select the **Recovery** tab.
3. Choose the recovery action you want in **First failure**, **Second failure**, and **Subsequent failures**.

## User accounts and the server

Windows distinguishes two types of service with respect to their privileges:

- Services running in the system account have privileged access to resources on the local



computer.

- Services set to run as a specific user are given the set of privileges assigned to that user.

The Easysoft ODBC-ODBC Bridge server is designed to be run in the system account. As soon as a client connects, authentication takes place using the LogonUser and LogonAuth attributes. The server calls the LogonUser() and ImpersonateLoggedOnUser() Windows APIs, so that all subsequent actions are attempted with only the privileges of LogonUser.

This is the default configuration of the Easysoft ODBC-ODBC Bridge server. The other option is to set up the server to always run with a specific user's permissions. This is not recommended, but if you do want to do this, the procedure is as follows.

### Important

Running the Easysoft ODBC-ODBC Bridge server under a named account for added security is counterproductive because you open up security holes by doing this. Firstly, anyone with the Easysoft ODBC-ODBC Bridge client can now connect to your data and secondly, to administer the Easysoft ODBC-ODBC Bridge server through HTTP you'll have to turn off HTTP authentication. Turning off HTTP authentication means anyone can change the Easysoft ODBC-ODBC Bridge server settings.

We recommend you run the Easysoft ODBC-ODBC Bridge server in the system account.

1. In Easysoft ODBC-ODBC Bridge web administrator, turn off authentication.
2. In Easysoft ODBC-ODBC Bridge web administrator, turn off HTTP authentication by setting **HTTPAdmin** to disabled.
3. Be sure that the user account you want to use is set up and that you know its password. It may be an existing end-user's account, or one created specifically for the Easysoft ODBC-ODBC Bridge server.

### Important

For the Easysoft ODBC-ODBC Bridge server to function correctly when running as a specific user, that user needs two important access rights: **Log On as a Service**, which is needed for the service to start, and **Act as Part of Operating System**, which is needed so that the service can authenticate connecting users for the Easysoft ODBC-ODBC Bridge server and for the web administrator.

4. In the **Services**, double click the Easysoft ODBC-ODBC Bridge server entry to open the service configuration dialog box.
5. On the **Log On** tab, in the **Log On As** section, choose **This Account**.
6. In the **This Account** box, enter the user name or click ... to browse for a user name.
7. In the **Password** box, enter the user account password.
8. Enter the same password in the **Confirm Password** box.
9. Choose **OK** to close the dialog box and commit your changes.

You may have noticed the **Allow Service to Interact With Desktop** checkbox. This has no effect on the Easysoft ODBC-ODBC Bridge server.

## Easysoft ODBC-ODBC Bridge configuration parameters in the registry

Easysoft ODBC-ODBC Bridge and web administrator configuration settings are stored in the registry. The settings are saved under this registry key:

\HKEY\_LOCAL\_COMPUTER\SOFTWARE\Wow6432Node\ Easysoft ODBC-ODBC Bridge\Configuration\System\Settings

On 32-bit Windows, Easysoft ODBC-ODBC Bridge and web administrator configuration settings are saved under this registry key:



\HKEY\_LOCAL\_COMPUTER\SOFTWARE\Easysoft ODBC-ODBC  
Bridge\Configuration\System\Settings

## Configuring the Easysoft ODBC-ODBC Bridge server on Linux and UNIX

On Linux and UNIX systems, the Easysoft ODBC-ODBC Bridge server is installed as a service under `inetd`, by default, but it may be configured to run in standalone mode.

When the Easysoft ODBC-ODBC Bridge server is installed as an `inetd`-controlled service, entries are added to the `inetd.conf` and `services` files. Examples of these entries might be

```
# inetd.conf entry
esoobserver stream tcp nowait root /bin/sh /bin/sh
/usr/local/easysoft/oob/server/server
```

```
# services entry
esoobserver 8888/tcp # Easysoft ODBC-ODBC Bridge server
```

These are just examples and may differ if you picked a different port, service name, or installation path.

### The mechanics of `inetd` and the server

`inetd` handles incoming network connections and starts the relevant program to service each incoming request.

When `inetd` starts (or is sent a `SIGHUP`), it reads `/etc/inetd.conf` for a list of what services to supply and to configure its handler for each service.

It then starts listening on the ports defined in `/etc/services` for each service configured in `inetd.conf`.

When a new connection is made, `inetd` starts the relevant program and hands over control.

The earlier example applies to the default installation: when a client connects to port 8888, `inetd` uses the information read from `/etc/inetd.conf` to decide which program to run and with what arguments.

Note that `/bin/sh` is repeated. The first time it appears (in the sixth position on the line) it's the name of the executable to run. The second time it appears it is the value to be passed as the zeroeth argument to `/bin/sh`. Normally, this is the same as the name of the executable.

In the default installation, the arguments on the line cause `/bin/sh` to run the Easysoft ODBC-ODBC Bridge server startup script `<InstallDir>/easysoft/oob/server/server`.

The script then sets any necessary environment variables required by the dynamic linker and runs `esoobserver`.

The script passes an argument of `inetd` to the server executable, which notifies the server that it's running under `inetd` rather than at the command line.

The Easysoft ODBC-ODBC Bridge server inherits the socket from `inetd` and begins communicating with the Easysoft ODBC-ODBC Bridge client. `inetd` returns to listening on port 8888 for any new connections.

#### Note

The web administrator feature of the Easysoft ODBC-ODBC Bridge server does not run when the Easysoft ODBC-ODBC Bridge server is configured under `inetd`, because `esoobserver` itself is not run until an Easysoft ODBC-ODBC Bridge client connects.

### Choosing another port or service name

If you have a port conflict or a service name conflict, or for some other reason you want to change the port or service name of the Easysoft ODBC-ODBC Bridge server, you'll need to edit the configuration files manually.

The `/etc/services` file maps service names to ports. The `/etc/inetd.conf` file lists the services that `inetd` will handle requests for.

After making the necessary changes, you'll have to send `SIGHUP` to the `inetd` process to make it re-read the files. This can be done with the command:

```
kill -HUP pid
```

where *pid* is the process ID of `inetd`. You need to be root to edit the configuration files and send `inetd` a `SIGHUP`.

### Configuring the server as standalone

By default, the Easysoft ODBC-ODBC Bridge server is installed as a service with entries in the `/etc/services` and `/etc/inetd.conf` files. (`inetd` is informed of the configuration file changes.)

However, the Easysoft ODBC-ODBC Bridge server can also run standalone. When run standalone, the Easysoft ODBC-ODBC Bridge server is faster at accepting connections than the `inetd` started server. So if you're making a lot of small connections to the Easysoft ODBC-ODBC Bridge server, starting it standalone (without `inetd`) is preferable. Also, the web administrator is only available when the Easysoft ODBC-ODBC Bridge server is running standalone.

Note that the Easysoft ODBC-ODBC Bridge server no longer uses `inetd` to listen on the socket when running standalone, so you cannot run `tcp wrappers`. You can use the facilities within the Easysoft ODBC-ODBC Bridge to set up access control.

To start the Easysoft ODBC-ODBC Bridge server in standalone mode:

1. Comment out the `esoobserver` entry in the `/etc/services` file.
2. Force `inetd` to re-read the files using:

```
kill -HUP inetd
```

3. Start the server with:

```
path/esoobserver standalone
```

This notifies the server that it must listen on the port itself and fork its own child process when a connection is made (the Easysoft ODBC-ODBC Bridge server reads the `port` and `httpport` settings in `path/oobserver.ini` for its configuration).

#### Note

You should start the Easysoft ODBC-ODBC Bridge server as the root user. Otherwise, it will be unable to change to the LogonUser specified in the Easysoft ODBC-ODBC Bridge client data source. However, if you don't start the Easysoft ODBC-ODBC Bridge server as root, no logon authentication will be done and all clients will run on the server as the user who started the Easysoft ODBC-ODBC Bridge server.

You could add a line to one of your `rc` scripts to have the Easysoft ODBC-ODBC Bridge server start up as standalone every time the computer boots.

The file you need to edit depends on your operating system so consult with your system administrator for further information.

When the Easysoft ODBC-ODBC Bridge server is started standalone, it forks a process to handle HTTP requests on port 8890. (You can change this in esoobserver.ini.)

You can then use your browser to go to the web administrator.

The HTTPAdmin configuration parameter controls whether you're prompted for a password before making changes in the web administrator. The default value for HTTPAdmin is disabled, which means that you're not prompted before accessing web administrator pages that let you change configuration settings. If you do want to control access to these web administrator pages, edit the HTTPAdmin value. Specify a user name that exists on the server the Easysoft ODBC-ODBC Bridge server is running on. If you want to do this, the Easysoft ODBC-ODBC Bridge server must be running as root on some computers to validate the user name and password.

## Changing server configurable parameters on Linux and UNIX

If you're running the standalone Easysoft ODBC-ODBC Bridge server, you can use the web administrator to examine and change Easysoft ODBC-ODBC Bridge configuration settings. If your Easysoft ODBC-ODBC Bridge server runs under inetd or you prefer a configuration file to a web-based interface, you'll need to edit the following file to configure the Easysoft ODBC-ODBC Bridge:

<InstallPath>/easysoft/oob/server/esoobserver.ini

If <InstallPath> is anything other than /usr/local/, there will be a symbolic link /usr/local/easysoft to the real easysoft directory.

esoobserver.ini entries have the following format:

```
{Settings}

<attribute> = <value>
```

Note that the bitmask values are given either in decimal or in hexadecimal and the access control lists are separated simply by spaces.

Set flags by entering the sum of the corresponding bitmask values of your required options.

## Connecting to your ODBC data source

### Configure an ODBC data source for your target database

If you have not done so already, create a system ODBC data source that connects to the database you want to access from your application. Create the ODBC data source on the same computer as the one on which you installed the Easysoft ODBC-ODBC Bridge server.

You create system data sources in the **System DSN** tab in **ODBC Data Source Administrator** (Windows) or `/etc/odbc.ini` (Linux and UNIX).

### Configure an Easysoft ODBC-ODBC Bridge client data source

On the computer where your application is installed, create an Easysoft ODBC-ODBC Bridge client data source. Your application then connects to this data source, which transparently forwards all ODBC calls the application makes to the Easysoft ODBC-ODBC Bridge server.

To do this 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:
  - 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-ODBC Bridge, and then choose **Finish**.

6. Complete the Easysoft ODBC-ODBC Bridge configuration dialog box.  
To find out how to do this, refer to the [Connection attributes](#) section.

To do this on Linux and UNIX:

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

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

An Easysoft ODBC-ODBC Bridge client data source can contain these attributes:

- BlockFetchSize
- CertFile
- ConnectAttempts
- Description
- DisguiseWide
- Encrypt
- GetInfoPassthru
- LogonAuth
- LogonUser
- MapExecDirect
- MetadataBlockFetch
- OverrideLength
- ServerPort
- TargetAuth
- TargetDsn
- TargetUser
- UseoobDbAuth

For more information about these attributes, refer to the [Connection attributes](#) section.

## Connection attributes

On Windows, the Easysoft ODBC-ODBC Bridge data source configuration dialog box, accessible when you create or edit an Easysoft ODBC-ODBC Bridge data source in **ODBC Data Source Administrator** contains the fields shown in the table.

On Linux and UNIX, the fields have [different names](#) to the dialog box labels, but have the same functionality.

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.
Servers	<p>The name or IP address of the remote host on which the Easysoft ODBC-ODBC Bridge server is running.</p> <p>The port on this computer where the Easysoft ODBC-ODBC Bridge server (or an agent on its behalf such as <a href="#">inetd</a>) is listening for incoming connections. The default port is 8888. Specify this port unless you know that the Easysoft ODBC-ODBC Bridge server is listening on another port. Separate the host name and port with a colon (:).</p> <p>If the remote system ODBC data source is available on more than one Easysoft ODBC-ODBC Bridge server, you can define a primary Easysoft ODBC-ODBC Bridge server for the ODBC data source and additional fallback Easysoft ODBC-ODBC Bridge servers. For more information about defining multiple Easysoft ODBC-ODBC Bridge servers, refer to <a href="#">Client support for fallback Easysoft ODBC-ODBC Bridge servers</a>.</p>
Username	<p>The name of a user on the computer on which the Easysoft ODBC-ODBC Bridge server is running (if required). The Easysoft ODBC-ODBC Bridge server changes to this user when an incoming connection is made.</p> <p>On Windows, you may need to include the domain name in this attribute. If so, use the format <i>domain/username</i>. For example: admin/John Smith.</p> <p>If you have defined multiple Easysoft ODBC-ODBC Bridge servers for this data source, the user name and password must be valid for each Easysoft ODBC-ODBC Bridge server that you specify</p>
Password	The password for Username.
TargetDSN	The System ODBC data source on the Easysoft ODBC-ODBC Bridge server computer.
TargetUser	The user name for the target database, if required.
TargetAuth	The password for TargetUser.

Name	Value
BlockFetchSize	<p>If you set BlockFetchSize to a value greater than 0, the Easysoft ODBC-ODBC Bridge determines whether to perform an optimization that retrieves multiple rows of data instead of one row at a time. The value you specify with BlockFetchSize is the number of rows to retrieve in one go. The maximum value is 100.</p> <p>This optimization is not done if the application binds columns in the result set. If your ODBC application uses cursors or positioned updates or deletes, you should not set this to greater than 1.</p> <p>By default, BlockFetchSize is turned off.</p>
MetaDataBlockFetch	<p>Turns on block fetching for metadata only (does not affect other result sets).</p> <p>This increases the speed of retrieving metadata, such as lists of tables or columns in a data source.</p> <p>By default, MetaDataBlockFetch is turned on.</p>
Override UID/PWD	<p>When turned on, the Easysoft ODBC-ODBC Bridge ignores the UID and PWD values passed from the application in the connection string and uses the TargetUser and TargetAuth values instead.</p> <p>By default, Override UID/PWD is turned off.</p>
Pass all SQLGetInfo requests on	<p>When turned on, <a href="#">SQLGetInfo</a> requests for the ODBC driver name, version, and ODBC version are sent to the remote ODBC driver instead of returning the Easysoft ODBC-ODBC Bridge client name, version, and ODBC version. This may be useful in situations when you want to know the remote ODBC driver so the application can take alternative actions.</p> <p>By default, Pass all SQLGetInfo requests on turned off.</p>
Disguise wide characters	<p>When turned on, column types described by the ODBC driver as SQL_Wxyz are disguised for applications that do not understand wide characters. For example, some versions of StarOffice need this when connecting to Microsoft SQLserver.</p> <p>By default, Disguise wide characters is off.</p>

Name	Value
Map SQLExecDirect	<p>When turned on, the Easysoft ODBC-ODBC Bridge maps <a href="#">SQLExecDirect</a> calls to <a href="#">SQLPrepare</a> and <a href="#">SQLExecute</a> for ODBC 2.0 applications.</p> <p>Between ODBC 2.0 and 3.0, a change in the allowable state transitions occurred. In ODBC 2.0, an application could call <a href="#">SQLDescribeParam</a> even in the executed state (that is, after the call to SQLPrepare and after the call to SQLExecute). In ODBC 3.0, this is no longer possible. The Easysoft ODBC-ODBC Bridge works around this limitation for ODBC 2.0 applications by mapping SQLExecDirect to SQLPrepare and SQLExecute and caching the parameter descriptions in between.</p> <p>The Easysoft ODBC-ODBC Bridge has always done this mapping for ODBC 2.0 applications. Map SQLExecDirect allows you to turn it off.</p> <p>By default, Map SQLExecDirect is turned on.</p>
Override Length in SQLGetxxx	<p>When turned on, the Easysoft ODBC-ODBC Bridge client overrides the BufferLength argument value in <a href="#">SQLGetInfo</a>, <a href="#">SQLGetConnectAtt</a>, <a href="#">SQLGetStmtAttr</a>, and <a href="#">SQLGetDescField</a> requests for integer information types. The Easysoft ODBC-ODBC Bridge client replaces the BufferLength value supplied by the application with the size of either an SQLINTEGER or an SQLSMALLINT.</p> <p>The ODBC specification says that ODBC drivers should ignore the BufferLength argument supplied by the application for integer information types. However, some ODBC drivers incorrectly expect to receive a buffer length for integer information types from the application. If an application passes a buffer length smaller than the size of an SQLINTEGER or an SQLSMALLINT, these ODBC drivers may return SQL_SUCCESS_WITH_INFO with SQLSTATE 01004 (Data truncated), instead of the integer attribute value. To work around this problem, enable Override Length in SQLGetxxx .</p> <p>By default, Override Length in SQLGetxxx attribute is turned off, which means that the Easysoft ODBC-ODBC Bridge passes the BufferLength value to the target ODBC driver unchanged.</p>
Connection Attempts	<p>The number of times to attempt a connection before failing. Each time a connection attempt fails the Easysoft ODBC-ODBC Bridge client waits (0.1 * connect attempt) seconds before trying again.</p> <p>If you're using multiple server definitions in a single ODBC data source you may find lowering this setting from the default of 5 an advantage as it reduces the time from the client failing to connect to the first server and starting a connection attempt to the second and subsequent servers.</p>



Name	Value
Block Statement Attributes	<p data-bbox="456 161 1490 353">A comma separated list of statement attributes that the Easysoft ODBC-ODBC Bridge client will block in <a href="#">SQLSetStmtAttr</a> calls. Any statement attribute you specify with Block Statement Attributes is not passed to the target ODBC driver. Instead, the Easysoft ODBC-ODBC Bridge client blocks them and returns SQL_SUCCESS.</p> <p data-bbox="456 371 1490 600">To block a statement attribute, specify the ODBC assigned statement attribute value. Look up these values in the ODBC specification and the SQL header files. For example, the values to use for the SQL_ATTR_MAX_ROWS and SQL_ATTR_METADATA_ID attributes are 1 and 10014. To block both these attributes, specify the following Block Statement Attributes value:</p> <div data-bbox="469 631 1477 698">1,10014</div> <p data-bbox="456 712 1490 904">Usually, you should not block any statement attributes as this may prevent your application from working correctly. The Block Statement Attributes setting was introduced to work around a problem with the Allbase ODBC driver, which was setting SQL_ATTR_MAX_ROWS on all statements when it was set on one particular statement.</p> <p data-bbox="456 922 1490 990">By default, the Easysoft ODBC-ODBC Bridge client does not block statement attributes.</p>

Name	Value
Encrypt Communication	<p>Whether to use encryption to protect data passed between the Easysoft ODBC-ODBC Bridge client and server.</p> <p>When turned off, the channel between Easysoft ODBC-ODBC Bridge client and server is not encrypted.</p> <p>When set to Encrypt If Available (1), the channel between Easysoft ODBC-ODBC Bridge client and server is encrypted if the Easysoft ODBC-ODBC Bridge server is capable of this (is a version that supports encryption and has been configured with a valid key and certificate file.) If the Easysoft ODBC-ODBC Bridge server is not capable of encryption, the connection between Easysoft ODBC-ODBC Bridge client and server will still succeed, but the channel between client and server will be unencrypted.</p> <p>When set to Encrypt Required (2), the channel between Easysoft ODBC-ODBC Bridge client and server is encrypted if the Easysoft ODBC-ODBC Bridge server is capable of this. Otherwise, the connection will fail. The Easysoft ODBC-ODBC Bridge client will not attempt to validate the certificate used by the Easysoft ODBC-ODBC Bridge server.</p> <p>When set to Encrypt Required + Validate (3), the channel between Easysoft ODBC-ODBC Bridge client and server is encrypted if the Easysoft ODBC-ODBC Bridge server is capable of this. Otherwise, the connection will fail. In addition, you need to specify the certificate used by the Easysoft ODBC-ODBC Bridge server with the Certificate File attribute. If the Easysoft ODBC-ODBC Bridge client is unable to verify the certificate, the connection will fail.</p> <p>Note that the <b>Test</b> button on the Easysoft ODBC-ODBC Bridge client data source configuration dialog box cannot be used to check whether encryption has been successfully set up. You need to use to connect to the Easysoft ODBC-ODBC Bridge client data source in your application to do this.</p>
Certificate File	<p>The path to a file containing the certificate used by the Easysoft ODBC-ODBC Bridge server to encrypt the connection between it and the Easysoft ODBC-ODBC Bridge client. For example, C:\Cert.pem. Use this attribute if Encrypt Communication is set to Encrypt Required + Validate.</p>
Unquote_Catalog_Fns attribute	<p>When turned on, the Easysoft ODBC-ODBC Bridge client removes the double quotes from around names in the calls to catalog functions such as <a href="#">SQLColumns</a>. This is a workaround for a problem in Applixware, which can incorrectly quote arguments to the catalog functions.</p> <p>By default, Unquote_Catalog_Fns is turned off.</p>

Name	Value
MetaData_ID_Identifier	<p>When turned on, the Easysoft ODBC-ODBC Bridge client calls <a href="#">SQLSetStmtAttr</a> with the SQL_ATTR_METADATA_ID attribute to set it to SQL_TRUE. This causes ODBC 3.x drivers to treat strings in metadata functions as literals. This attribute is useful if you have an application you cannot change that assumes all strings in metadata calls are treated as literal but are going to a database containing tables with special characters in them. (For example, _) Applixware's AXData sometimes needs this attribute defining.</p> <p>By default, MetaData_ID_Identifier is turned off.</p>
DecAsNumeric	<p>When turned on, the Easysoft ODBC-ODBC Bridge client maps an SQL_DECIMAL type in a <a href="#">SQLBindParameter</a> call to an SQL_NUMERIC type. The DecAsNumeric attribute was introduced to work around a problem with Oracle Heterogeneous Services and Microsoft Access. When Oracle attempted to bind a type as a SQL_DECIMAL, the Microsoft Access ODBC driver returned the error "Optional feature not implemented".</p> <p>By default, DecAsNumeric is turned off.</p>

## Client support for fallback Easysoft ODBC-ODBC Bridge servers

By default, the Easysoft ODBC-ODBC Bridge client is passed an ODBC data source that defines a single Easysoft ODBC-ODBC Bridge server with which to connect.

If the client fails to connect to the specified server, it then tries 4 more times by default, pausing (0.1 \* connection attempts) seconds between each attempt. (The number of additional attempts can be configured with the Easysoft ODBC-ODBC Bridge client data source attribute Connect Attempts.)

Each client data source can also define multiple servers and ports. If the client fails to connect to the first server in the list, it retries with each server in the list until either a connection is made or the list is exhausted.

You specify the Easysoft ODBC-ODBC Bridge servers when configuring the client data source. On Windows, you configure client data source in the Easysoft ODBC-ODBC Bridge DSN configuration dialog box.

To add multiple servers by using the Easysoft ODBC-ODBC Bridge DSN dialog box on Windows:

1. In the Easysoft ODBC-ODBC Bridge DSN configuration dialog box, choose the **Server** tab.
2. Click the ... button.
3. In the **Server** box, enter the host name or IP address of the primary Easysoft ODBC-ODBC Bridge server.
4. In the **Port** box, enter the port on which the primary Easysoft ODBC-ODBC Bridge server is listening.

The default port is 8888. Specify this port unless you know that the Easysoft ODBC-ODBC Bridge server is listening on another port.

5. Choose **Add**.
6. In the **Server** box, enter the host name or IP address of a fallback Easysoft ODBC-ODBC Bridge server on which the target system data source is also defined.
7. If the fallback Easysoft ODBC-ODBC Bridge server is listening on a different port to the default, in the **Port** box, enter the port. Otherwise, skip this step.

## 8. Do one of the following:

- To add another fallback Easysoft ODBC-ODBC Bridge server, choose **Add**, then repeat steps 6 and 7.
- If you've finished adding Easysoft ODBC-ODBC Bridge servers, choose **Test** to test the connection process for each server in the list.

If a connection can't be made to a particular server, in the **Servers** list, a red cross displays next to the server. The **Test Log** output shows the reason why the test failed in red text. For help on how to resolve the connection problem, choose **Help**.

<b>Note</b>	You must enter values in the <b>Username</b> and <b>Password</b> boxes in the <b>Server</b> tab before testing, unless Authentication is turned off in the Easysoft ODBC-ODBC Bridge server.
-------------	--

9. Choose **OK**.

The Easysoft ODBC-ODBC Bridge client tries to connect to Easysoft ODBC-ODBC Bridge servers in the order they are defined in the list. To reorder the Easysoft ODBC-ODBC Bridge servers, in the **Servers** list, choose the Easysoft ODBC-ODBC Bridge server whose position you want to change. Then use the **Up** or **Down** buttons to move the Easysoft ODBC-ODBC Bridge server up or down the list.

When you add a new Easysoft ODBC-ODBC Bridge server, it's added to the bottom of the list. If you want to add a server in a different position in the list, select the server above which you want the new server to appear, then choose **Insert**.

On Linux and UNIX, you configure Easysoft ODBC-ODBC Bridge client data sources by editing the `odbc.ini` file.

To add multiple servers by editing `odbc.ini`:

- Specify the primary Easysoft ODBC-ODBC Bridge server and each fallback Easysoft ODBC-ODBC Bridge server with the `ServerPort` attribute. Use the following format:

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

where *primaryserver* is the name or IP address of the primary Easysoft ODBC-ODBC Bridge server on which the target system data source is defined, *port* is the port on which the Easysoft ODBC-ODBC Bridge server is listening (the default is 8888), and *fallbackserver* is the name or IP address of another Easysoft ODBC-ODBC Bridge server on which the target system data source is defined.

For example:

```
[OOB_SAMPLE]
Driver = OOB
TargetDSN = MySystemDSN
LogonUser = myuser
LogonAuth = mypassword
ServerPort = myprimaryserver:8888, myfallbackserver:8888
```

Assuming an example primary Easysoft ODBC-ODBC Bridge server named `myprimaryserver` and a fallback Easysoft ODBC-ODBC Bridge server named `myfallbackserver`, the Easysoft ODBC-ODBC Bridge client will attempt a connection to port 8888 on `myprimaryserver`. If this fails and after Connection Attempts additional attempts a connection has still not been made, the client will try to connect to the backup server `myfallbackserver`.

There may be an advantage in lowering `ConnectionAttempts` from the default of 5 if multiple

server definitions are being used in a single data source, as this will reduce the time taken from the client failing to connect to the first server and starting a new connection attempt to the second and subsequent servers.

## Testing the Easysoft ODBC-ODBC Bridge server

If you're having problems with the connection between your application and the Easysoft ODBC-ODBC Bridge server, you can use oobping to help diagnose and fix any issues.

oobping is a small program distributed with the Easysoft ODBC-ODBC Bridge, and is a valuable tool for checking Easysoft ODBC-ODBC Bridge connectivity and diagnosing connection problems or connection timing issues

On Windows, the oobping.exe program is located in the *install\_dir\Easysoft\Easysoft ODBC-ODBC Bridge* directory.

On UNIX and Linux, there are two versions of oobping located in the *install\_dir/easysoft/bin* directory.

- oobpings (a statically linked version not depending on anything other than libc)
- oobpingd (a dynamically linked version linked against the libesoobclient shared object)

**Note** To use oobpingd, you may need to set and export your LD\_LIBRARY\_PATH or LD\_RUN\_PATH or LIBPATH to include *install\_dir/easysoft/oob/client* and *install\_dir/easysoft/lib*.

oobping has the following command line:

```
oobping [-h host | -d ODBC_connection_string] {-t port} {-u osuser -p ospassword} {-e}
```

where:

- -h *host*

The name or IP address of the machine where the Easysoft ODBC-ODBC Bridge server is located.

- -d *ODBC\_connection\_string*

An ODBC connection string consisting of a list of semi-colon separated attribute=value pairs, as defined by ODBC. For example:

```
DSN=my_system_dsn;UID=myuser;PWD=p455w0rd;
```

If the -u or -p attributes are also specified as well as -d then

LogonUser=my\_os\_user;LogonAuth=p455w0rd; (where my\_os\_user and p455w0rd are the values specified for -u and -p) will be added to the end of the connection string.

**Note** When running oobping from a UNIX or Linux shell, you may have to enclose the ODBC connection string, user name and password with single quotes. Do this to protect any spaces or special characters that the shell might interpret.

When running oobping from a Windows Command Prompt, use double quotes to protect spaces.

- -t *port*

The port on which the Easysoft ODBC-ODBC Bridge server is listening.

- -u *os\_user*

A valid user name on the Easysoft ODBC-ODBC Bridge operating system.

- -p *os\_password*

A password for the user specified with the -u attribute.

- -e

Show the amount of time (in seconds) to complete the requested operation.

## Worked examples

The following examples illustrate the ways in which oobping can be used to investigate connection issues:

- [Example 1: Check the Easysoft ODBC-ODBC Bridge server is running on the correct machine and listening on the correct port](#)
- [Example 2: Check Easysoft ODBC-ODBC Bridge server authentication](#)
- [Example 3: Check connectivity to a specified remote data source](#)
- [Example 4: Checking connection times](#)

### Note

The Easysoft ODBC-ODBC Bridge installation uses oobping to test the connection to the remote Easysoft ODBC-ODBC Bridge server.

### Example 1: Check the Easysoft ODBC-ODBC Bridge server is running on the correct machine and listening on the correct port

oobping connects to port 8888 on the machine myserver, where Easysoft ODBC-ODBC Bridge server version 1.1.0.00 is reported as running.

```
oobping -h myserver -t 8888

Host: myserver, Port: 8888
Attempting connection...OK
Examining Server...
Easysoft ODBC-ODBC Bridge server Version: 1.1.0.00
Easysoft ODBC-ODBC Bridge server Name: Easysoft ODBC-ODBC Bridge
```

### Note

oobping defaults to port 8888, so the -t attribute can be omitted.

If the wrong port is specified or some other service is listening on the specified Easysoft ODBC-ODBC Bridge server port, you will get a variety of errors, such as when pointing oobping at an SMTP server on port 25:

```
oobping -h myserver -t 25

Host: myserver, Port: 25
Failed to receive data
Packet (size=842149920) too big for buffer (size=256)
```

If there's nothing listening on the specified port, you'll get a connection refused error, and you should check that the Easysoft ODBC-ODBC Bridge server is running and is configured to use the specified port. For example:

```
oobping -h myserver -t 8889

Host: myserver, Port: 8889
```

```
Connection refused, connect(), after 5 attempts
```

If you have specified access control rules in the Easysoft ODBC-ODBC Bridge server then you might see an error such as this:

```
oobping -h myserver -t 8888

Host: myserver, Port: 8888
Client denied access due to access control rule.
```

Here the machine you are running oobping on has been denied access to the Easysoft ODBC-ODBC Bridge server and you should check the access control rules on the security page of the Easysoft ODBC-ODBC Bridge web administrator.

### Example 2: Check Easysoft ODBC-ODBC Bridge server authentication

Once you're sure a connection can be made to the Easysoft ODBC-ODBC Bridge server with oobping (as in example 1) you can check Easysoft ODBC-ODBC Bridge server authentication.

The -u and -p arguments to oobping allow you to specify a valid operating system user name and password.

**Note** If you have turned off authentication in the Easysoft ODBC-ODBC Bridge server then any user name and password will work no matter what you enter.

The values specified with -u and -p are equivalent to the Easysoft ODBC-ODBC Bridge client data source attributes LogonUser and LogonAuth.

For example:

```
oobpings -h myserver -t 8888 -u 'my_os_user' -p 'p455w0rd'

Host: myserver, Port: 8888
Attempting connection...OK
Examining Server...
Easysoft ODBC-ODBC Bridge server Version: 1.1.0.00
Easysoft ODBC-ODBC Bridge server Name: Easysoft ODBC-ODBC Bridge
Trying to authenticate...OK
```

If there's something wrong with the user name or password, the output will look something like this:

```
oobpings -h myserver -t 8888 -u 'my_os_user' -p 'p455w0rd'

Host: myserver, Port: 8888
Attempting connection...OK
Examining Server...
Easysoft ODBC-ODBC Bridge server Version: 1.1.0.00
Easysoft ODBC-ODBC Bridge server Name: Easysoft ODBC-ODBC Bridge
Trying to authenticate...Fail
Authentication failure (error number 1326)
```

In this case, the error returned by the remote user name and password authentication service is 1326 (as the server was on Windows, this is a Windows error code).



All the common Windows error codes can be found in the [Easysoft ODBC-ODBC Bridge knowledge base](#).

### Example 3: Check connectivity to a specified remote data source

Once you have completed examples 1 and 2 you should have:

- The name of a server where an Easysoft ODBC-ODBC Bridge server is running.
- The port it's listening on.
- A valid operating system user name and password.
- Proved your connection is allowed by the Easysoft ODBC-ODBC Bridge server access control rules.

You can now check connectivity to the target ODBC data source, which can be done in two ways with oobping.

The simplest way is to add a local DSN to your `odbc.ini` file and then specify the DSN in the `-d` argument as `DSN=dsnname`. This method tests you have defined the local Easysoft ODBC-ODBC Bridge client DSN correctly and put the definition in the right file.

An alternative method is to specify all the connection attributes in the `-d` argument, not just the DSN. For example, suppose as a result of examples 1 and 2 you have the following information:

```
Server      = my_server
Port        = 8888
LogonUser   = my_os_user
LogonAuth   = p455w0rd
```

Now enter the name of the target ODBC data source on `my_server` in the `TargetDSN` attribute.

**Note** Note that the target data source **must** be a remote system data source, as you cannot access user data sources.

If the database also needs login information then the database user name and password are specified using the `TargetUser` and `TargetAuth` attributes.

The corresponding Easysoft ODBC-ODBC Bridge client data source in `odbc.ini` would be:

```
[my_dsn]
ServerPort  = my_server:8888
LogonUser   = my_os_user
LogonAuth   = p455w0rd
TargetDSN   = my_system_dsn
TargetUser   = my_db_user
TargetAuth   = p455w0rd
```

**Note** `Driver=OOB` must be included in the data source to use the `unixODBC` driver manager, but this is not needed when using `oobping`.

You can run `oobping` as follows:

```
oobping -d "DSN=my_dsn;"

Using Connection string :
DSN=my_dsn;
Connected OK
```

```

01000:1:5701:[NetConn: 032bc620][Microsoft][ODBC SQL Server Driver]
[SQL Server]Changed database context to 'pubs'.
01000:2:5703:[NetConn: 032bc620][Microsoft][ODBC SQL Server Driver]
[SQL Server]Changed language setting to us_english.
OutConnectionString:
DSN=my_system_dsn;UID=my_db_user;PWD=p455w0rd;SERVERPORT=myserver:8888;
TARGETDSN=my_system_dsn;LOGONUSER=my_os_user;LOGONAUTH=p455w0rd;
Connected to database: pubs
DBMS Name: Microsoft SQL Server
Driver Name: esoobclient
Driver Version: 01.00.0043
Disconnecting

```

The ODBC connection string "DSN=my\_dsn" was passed from oobping to the Easysoft ODBC-ODBC Bridge client, but the Easysoft ODBC-ODBC Bridge client did not receive sufficient attributes in the connection string.

To find out what it should do, the Easysoft ODBC-ODBC Bridge client looked up the DSN my\_dsn in the odbc.ini file, where it obtained the additional attributes ServerPort, LogonUser, LogonAuth, TargetDSN, TargetUser, and TargetAuth.

The Easysoft ODBC-ODBC Bridge client then connected to the Easysoft ODBC-ODBC Bridge server on my\_server through port 8888, logged you in with LogonUser and LogonAuth values and finally connected by using ODBC to the remote data source my\_system\_dsn.

**Note** Although in this example the ODBC data source pointed to Microsoft SQL Server and informational diagnostics reported the language as "us\_english" and the database as "pubs", not all databases will return informational messages on the connection.

The OutConnectionString is a string returned by the Easysoft ODBC-ODBC Bridge client, which you can use to connect to this data source again and the final messages show the database name, DBMS name, driver name, and driver version.

Instead of specifying just the name of a DSN to oobping and defining the other connection attributes in the odbc.ini file you can specify all the connection attributes in one go and not use an odbc.ini file.

The OutConnectionString shows you what connection string you could have passed to the Easysoft ODBC-ODBC Bridge client to connect without a DSN (if you remove the DSN=my\_dsn;) to produce the same result. For example:

```

oobping -d
"UID=my_db_user;PWD=p455w0rd;SERVERPORT=my_server:8888;TARGETDSN=my_system_dsn;LOGO
NUSER=my_os_user;LOGONAUTH=p455w0rd;"

```

You might be slightly confused as to why TargetUser and TargetAuth is specified in the odbc.ini file, but it's UID and PWD in the connection string.

Strictly speaking, the ODBC defined attributes are UID are PWD and they are passed to the database for authentication.

As far as ODBC connection strings are concerned, UID and PWD and TargetUser and TargetAuth are synonymous in the Easysoft ODBC-ODBC Bridge, but if specified in the odbc.ini file you should always use TargetUser and TargetAuth.

Now you know how to use oobping with connection strings, here are some examples of common problems you might encounter:

```
oobping -d
"UID=my_db_user;PWD=p455w0rd;TargetDSN=test;LogonUser=my_os_user;LogonAuth=p455w0rd
"

Using Connection string :
UID=my_db_user;PWD=p455w0rd;TargetDSN=test;LogonUser=my_os_user;LogonAuth=p455w0rd
IM002:1:0:[Easysoft ODBC (Client)]
Data source not found and no default driver specified
HY000:2:0:[Easysoft ODBC (Client)]
general error: Missing attribute(s): SERVERPORT
```

The initial diagnostic, IM002 is defined by ODBC, but as it is rather vague, the Easysoft ODBC-ODBC Bridge client added a secondary more helpful diagnostic.

Here the Easysoft ODBC-ODBC Bridge connection attribute ServerPort was omitted and so the Easysoft ODBC-ODBC Bridge client did not know which server to connect to.

```
oobping -d

"ServerPort=myserver:8888;UID=my_db_user;PWD=p455w0rd;TargetDSN=test;LogonUser=my_o
s_user;LogonAuth=p455w0rd"

Using Connection string :
ServerPort=myserver:8888;UID=my_db_user;PWD=p455w0rd;TargetDSN=test;LogonUser=my_os
_user;LogonAuth=p455w0rd
28000:1:18456:[][Microsoft][ODBC SQL Server Driver][SQL Server]
Login failed for user 'my_db_user'.
```

This Microsoft SQL Server error suggests the database password p455w0rd is not correct for the database user my\_db\_user or that my\_db\_user is not a valid user, and illustrates the importance of identifying which component is reporting an ODBC error.

You should examine the components displayed in [], where the one furthest to the right is the reporting component and as you move left through the components you are moving closer to the Easysoft ODBC-ODBC Bridge client.

For example, if you specified a TargetDSN that did not exist on the server machine then the driver manager on the remote machine would be the component reporting the error and so the diagnostic error message (for Windows) would be:

```
IM002:1:0:[][Microsoft][ODBC Driver Manager]

Data source name not found and no default driver specified
```

The other important thing to note here is that oobping outputs any ODBC diagnostics as:

```
ODBC State : diagnostic sequence : ODBC native error: text
```

The native error code is specific to the component reporting the error, and can be referenced in the Microsoft SQL Server documentation, which should tell you more accurately in which circumstances this error is reported.

## Example 4: Checking connection times

## 52 Worked examples

---

oobping includes the -e option, which times the requested operation and can be invaluable in diagnosing a slow connection and determining which phase the problem is occurring in.

These operations may also use the -e switch. For example:

```
oobping -e -h my_server -t 8888

Host: my_server, Port: 8888
Attempting connection...OK
Examining Server...
Easysoft ODBC-ODBC Bridge server Version: 1.1.0.00
Easysoft ODBC-ODBC Bridge server Name: Easysoft ODBC-ODBC Bridge
Time for execution: 0.16s
```

If this is repeated with the -u and -p attributes, you can work out the extra time required to perform the operating system login:

```
oobping -e -h my_server -u my_os_user -p p455w0rd

Host: my_server, Port: 8888
Attempting connection...OK
Examining Server...
Easysoft ODBC-ODBC Bridge server Version: 1.1.0.00
Easysoft ODBC-ODBC Bridge server Name: Easysoft ODBC-ODBC Bridge
Trying to authenticate...OK
Time for execution: 0.52s
```

This example clearly demonstrates the extra time required to authenticate a user.

<b>Note</b>	When connecting to a server in part of a Windows domain, the server may need to contact the Primary Domain Controller (PDC). This can take a significant amount of time.
-------------	--

In addition, if you install the Easysoft ODBC-ODBC Bridge server on the same machine as Microsoft SQL Server, the Easysoft ODBC-ODBC Bridge server and SQL Server compete for CPU time. It is often quicker to put the Easysoft ODBC-ODBC Bridge server on a different Windows machine.

## Tracing

There are three ways to trace the ODBC calls an application makes through the driver manager and the Easysoft ODBC-ODBC Bridge client ODBC driver:

- ODBC Driver Manager tracing.
- Easysoft ODBC-ODBC Bridge client ODBC driver tracing.
- An application can turn tracing on by calling the ODBC API [SQLSetConnectAttr\(...,SQL\\_ATTR\\_TRACE,...\)](#). The trace file name may also be specified with the SQLSetConnectAttr attribute SQL\_ATTR\_TRACEFILE.

Starting tracing in the Driver Manager is platform-specific:

To do this on Windows, in **ODBC Data Source Administrator**, click the **Tracing** tab. Enter a trace file name in the space provided. Choose **Machine-wide tracing for all user identities**, and then choose **Start tracing now**.

To do this on Linux and UNIX:, add this section to the start of odbcinst.ini (usually located in /etc/odbcinst.ini):

```
[ODBC]
Trace = Yes
TraceFile = /path/filename
```

For example:

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

The user who is running the application to be traced must have write permission to TraceFile (and to the directory containing it).

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

You can enable Easysoft ODBC-ODBC Bridge client ODBC tracing by SQLSetConnectAttr(...,SQL\_ATTR\_TRACE,...) or by a platform-specific method:

On Windows, update the Registry by running regedit and editing:

```
HKEY_LOCAL_MACHINE\SOFTWARE\EASYSOFT ODBC-ODBC
BRIDGE\CONFIGURATION\SYSTEM\SETTINGS
```

Edit the Logging string value and set it to 0xffffffff, which turns on all tracing.

A trace file named esoobclient.log is written to the LogDir directory (which is defined in the same registry key).

On Linux and UNIX, Easysoft ODBC-ODBC Bridge client ODBC driver tracing can be enabled by adding a {Settings} section to the end of your odbc.ini file. For example:

```
{Settings}
Logging = 0xffffffff
```

The Logging value is a bitmask and the values are listed in the [Easysoft ODBC-ODBC Bridge knowledge base](#).

A trace file named esoobclient\_<PID>.log is created in LogDir, where <PID> is the process ID.

## Reporting and statistics

The Easysoft ODBC-ODBC Bridge provides a number of performance management reports and connection statistics.

- [Introduction](#)
- [Changing the refresh frequency](#)
- [DSN statistics](#)
- [Host statistics](#)
- [Termination statistics](#)
- [Log of failing SQL](#)
- [Auditing](#)
- [Auditing](#)
- [Browsing system data sources in the web administrator](#)

## Introduction

One of the most frequent requests that we receive from enterprise customers is for more reports and statistics, so that they can find out exactly what the Easysoft ODBC-ODBC Bridge server is doing at any time.

The Easysoft ODBC-ODBC Bridge's web administrator provides a number of statistics detailing server up time, total connections, total successful connections, active threads and processes, peak concurrent threads and processes and the time of the last connect or disconnect.

The web administrator's **Statistics** page provides:

- **Server up time** The time in days, hours, minutes, and seconds since the Easysoft ODBC-ODBC Bridge server was started.
- **Server CPU time(s)** One or more values are shown. If only one value is shown, it's the total CPU time consumed by the Easysoft ODBC-ODBC Bridge server. If two times are shown, the first is the user time and the second is the kernel time. Note that the CPU times shown include CPU time consumed by the ODBC Driver Manager, any ODBC drivers and any child processes.  
(**Server CPU time(s)** is only visible if the **ShowProcessTime** option on the **Configuration** page is turned on.)
- **Total Connections** The total number of connections (or attempted connections) to the Easysoft ODBC-ODBC Bridge server. This includes connections dropped due to invalid license keys or insufficient license slots, port scanners, or anyone using telnet to access the Easysoft ODBC-ODBC Bridge server ODBC port.
- **Total Threads/Processes** The total number of threads or processes that the Easysoft ODBC-ODBC Bridge server has created during its execution. Connections denied access because of a server access control rule (but not a DSN access control rule) or because the value of either **MaxThreadCount** or **MaxClientConnect** has been exceeded don't count because the Easysoft ODBC-ODBC Bridge server does not start a thread or process for these.
- **Active Threads/Processes** The total number of active threads or processes the Easysoft ODBC-ODBC Bridge server has created to handle ODBC connections. However, this number may exceed the actual active count, as the Easysoft ODBC-ODBC Bridge server only looks for exited threads and processes when five seconds has elapsed without any connections or every 10 connections (to give preference to incoming connections). Note that if **MaxThreadCount** or **MaxClientConnect** is set to anything other than 0, the Easysoft ODBC-ODBC Bridge server has to reap exited threads and processes every time a new connection arrives. In this case, the active count will always be up to date immediately after an ODBC connection from a client.
- **Peak concurrent Threads/Processes** The maximum value recorded in the **Active Threads/Processes** field.

- **Connections/minute** The server up time (in minutes) divided by **Total Threads/Processes**. This is a very useful measure of how busy the Easysoft ODBC-ODBC Bridge server is. However, if you're using persistent connections from your client application, you should realise that this number is likely to always be low, since connections are being reused by your client.
- **Last Connection time** The time of the last ODBC connection.
- **Last Disconnect time** The time of the last disconnect.
- **Number of different client hosts** The number of different client machines that have connected to the Easysoft ODBC-ODBC Bridge server (where a client machine is identified by its IP address). You can click on this link to get a list of IP addresses or machine names. Machine names are only displayed if you have turned on **ReverseLookup**.
- **Audit File** A link to another page that shows entries from the last audit trail file. As the audit trail file is renamed every day, you'll also get links to older audit trail files. Graphs of connections per hour and connections per minute are available for each audit file. Note that an audit trail file is only produced if **AuditODBCAccess** is turned on.
- **DSN statistics** Shows the first ten DSNs accessed with the Easysoft ODBC-ODBC Bridge, the number of times a connection to this DSN has been opened, the total time (in seconds) connections to this DSN have been open, the connections per minute, and the average time per connection. Note that the total time is concurrent time: if, for example, the same DSN is opened twice concurrently for ten seconds, **Total time** shows 20 seconds.

## Changing the refresh frequency

The web administrator uses a set of template files into which the dynamic data is inserted before sending it back to your browser.

The template file for the **Statistics** page is index.html, which is located in the templates directory wherever you installed the Easysoft ODBC-ODBC Bridge server.

Edit the index.html file and near the top you'll find:

```
meta http-equiv="refresh" content="60"; URL=/index.html
```

Change 60 (the refresh time in seconds) to your preferred setting.

### Note

Setting the refresh time to a very low value increases the workload on the Easysoft ODBC-ODBC Bridge server process that handles HTTP requests. As this may adversely affect the Easysoft ODBC-ODBC Bridge server's response time, times much less than 60 seconds are not recommended.

## DSN statistics

The Easysoft ODBC-ODBC Bridge records each DSN to which it is connected and displays a table of DSN statistics:

- **Number of connections to this DSN** By comparing this number for each DSN you can find out which is the busiest data source on your server.
- **Total time connected to this DSN** This is concurrent time, so two concurrent connections, connected for the same 5 second period registers as 10 seconds total time.
- **Connections per minute to this DSN** The server up time (in minutes) divided by the number of connections to this DSN. This includes connections that were refused by the server for various reasons, for example, access control or authentication.
- **Average time per connection** The total time connected to this DSN divided by the number of connections to this DSN. If you know your client applications connect, execute some SQL, and then disconnect, this is a good measure of how much processing there is for each connection.

This number is likely to be higher for client applications using persistent connections, as there may be long periods of connected time where nothing is happening.

## Host statistics

In the web administrator statistics page, there's a link called **Number of different client hosts**, which takes you to a table of statistics per connecting client:

- **IP Address** The IP address of the client.
- **FQDN** The Fully Qualified Domain Name of connecting clients. This is displayed as "unknown" unless **ReverseLookup** is turned on.
- **Connections** The number of connections per client. If you have more than one connecting client, this allows you to find out which client is putting the greatest load on your server.

## Termination statistics

The contents of the **Statistics** page are written on exit to a file called `esoob_stats_<dd>_<mm>_<yyyy>_<hhmmss>.stats` where `<dd>` is the day number, `<mm>` is the month number, `<yyyy>` is the year number and `<hhmmss>` is the time in hours, minutes, and seconds.

This file is written to the directory defined by **LogDir**. Here's some example output:

```
server up time 0 days, 18 hours, 5 minutes and 23 seconds
server CPU time(s): u=92.66 k=242.72
Total Connections: 58966
Total Threads/Processes: 58966
Active Threads/Processes: 4
Peak concurrent Threads/Processes: 74
Connections/minute: 54
Last Connection time: Wed Jul 04 08:55:13 2001
Last Disconnect time: Wed Jul 04 08:55:12 2001
DSN Connections Total Connections Average
DSN Time(s) per minute Time per
DSN connection(s)
dsn1 3217 4153 2 1.29
dsn2 55566 212747 51 3.83
dsn3 42 8060 0 191.90
dsn4 111 101 0 0.91

Number of different client hosts: 3
192.168.0.1 (test1.easysoft.com)
192.168.0.2 (test2.easysoft.com)
192.168.0.3 (test3.easysoft.com)
```

## Log of failing SQL

The Easysoft ODBC-ODBC Bridge server can keep a log of any failed [SQLPrepare](#), [SQLExecute](#) or [SQLExecDirect](#) functions.

To turn on this feature, select the **LogFailingSQL** parameter on the **Configuration** page. This creates a file called `oob_sql.log` when the first failing SQL occurs.

This file is located in the directory defined by the **LogDir** parameter on the **Configuration** page.



The failing SQL log file has two types of entry:

- An entry showing a piece of SQL that's failed:

```
date time "DSN" SQL
```

For example:

```
12-01-2025 10:32:55 "My_System_DSN" INSERT INTO My_Table VALUES (1,2)
```

The SQL may fail for a number of reasons. For example, the SQL fails to parse correctly or the database engine rejects the SQL (a duplicate key error, for example):

```
31-07-2024 11:24:24 "example" UPDATE Title SET Count=Count+1 WHERE id = 1
01-08-2024 09:08:08 "web" SELECT columnnoexist FROM stores
01-08-2024 09:05:05 "test" UPDATE sales SET stor_id = 8042 WHERE stor_id = "fred"
01-08-2024 09:06:06 "test" SELECT * FROM tablenoexist
```

The DSN name field is extremely useful when identifying problems.

In this example, the "web" and "test" DSNs both point at the same database, but "web" is the live version of the application and "test" is used by software developers.

As the client applications are on different machines, the TargetDSN used by the Easysoft ODBC-ODBC Bridge client is set to "web" on one client and to "test" on the other.

Failing SQL in the log to "test" would not be uncommon as this is under development, but failures for "web" indicate a problem a real user might have experienced running live software.

- An entry logging the first database error retrieved **after** a piece of SQL failed:

```
date time SQLState="xxxxx", NativeError=nnnn, Msg="xxxxxxxxx"
```

For example:

```
12-04-2002 10:32:55 SQLState="23000", NativeError=2627 Msg="[Microsoft][ODBC SQL
server Driver][SQL server] Violation of PRIMARY KEY constraint"
```

The failing SQL file is written to as follows:

1. Each piece of SQL executed with either [SQLPrepare](#), [SQLEExecute](#), or [SQLExecDirect](#) is documented where those APIs fail.
2. An entry is written in the first format shown above if an SQLPrepare, SQLEExecute, or SQLExecDirect function fails.
3. An entry is written in the second format if the application calls [SQLError](#) or [SQLGetDiagRec](#).

When possible, failing SQL is written to one line of the failing SQL log, followed by the error text returned from the database engine on the next line.

However, this is **not** always the case, as the failing SQL statements and the error text can get mixed up if the server is very busy. (For example, if multiple concurrent connections all fail at the same time.)

## Auditing

The Easysoft ODBC-ODBC Bridge server records all activity to an audit trail file if the **AuditODBCAccess** is turned on.

The **Audit File** link on the **Statistics** page displays a page showing:

- The first page from the current audit trail file.
- Links to each recorded audit trail file.
- Two graphs for each audit trail file, one showing connections per minute and another connections per hour over the period of that audit trail file (a new file is opened each day). The graphs show when the Easysoft ODBC-ODBC Bridge server is busiest (so that better decisions can be made for scheduling administration tasks that require taking the server down, for example) and highlight unexpected bursts in activity (for example, a search robot indexing your web site).

## Audit file description

The current audit file is called `esoob_access.log` and is written to the directory specified by the **LogDir**.

Here's an example of a few lines from the audit file:

```
Mon, 02 Jul 2024 15:01:07 GMT CONNECT 192.168.0.1(test1.easysoft.com)
Mon, 02 Jul 2024 15:01:12 GMT DISCONNECT 192.168.0.1(test1.easysoft.com)
Fri, 06 Jul 2024 13:10:46 GMT AUTH_DENIED 192.168.0.1(test1.easysoft.com)
Fri, 06 Jul 2024 14:34:36 GMT REFUSED_BY_DSN_RULE test(Martin Evans)
Fri, 06 Jul 2024 15:18:34 GMT REFUSED_MAXCLIENTCONNECT 192.168.0.3(unknown)
Fri, 06 Jul 2024 15:19:33 GMT REFUSED_MAXCONCURRENT 192.168.0.2(test2.easysoft.com)
Mon, 09 Jul 2024 13:04:02 GMT REFUSED_BY_RULE 192.168.0.1(test1.easysoft.com)
```

The fields in each line of the audit file are:

- **Day** The day of the week
- **Day number** The day of the month
- **Month** The month name
- **Year** The year
- **Time** Hours, minutes, and seconds
- **Timezone** The time zone
- **Event** An event code
- **Event arguments** Details of the client, DSN, or user name

The events and their descriptions are:

- **CONNECT** An Easysoft ODBC-ODBC Bridge client has connected to the Easysoft ODBC-ODBC Bridge server. This doesn't mean the client gained successful access to the ODBC data source, but that the server accepted the connection. If the Easysoft ODBC-ODBC Bridge server's `Authentication_Disabled` parameter was turned off, the client was authenticated by the operating system and also completed any access control test based on its IP address. The last field of the audit file shows the client IP address and also the client FQDN in brackets if `ReverseLookup` is turned on.
- **DISCONNECT** An Easysoft ODBC-ODBC Bridge client disconnected from the Easysoft ODBC-ODBC Bridge server. The audit file does not indicate why the client disconnected. The client may have called [SQLDisconnect](#), been interrupted, or the Easysoft ODBC-ODBC Bridge server could have timed out the connection because it was inactive for the number of seconds set in the `Timeout`.
- **AUTH\_DENIED** This event happens when an Easysoft ODBC-ODBC Bridge client connects to the Easysoft ODBC-ODBC Bridge server, Easysoft ODBC-ODBC Bridge server authentication is turned and the client passes an invalid user name or password (`LogonUser` or `LogonAuth`) pair for the operating system. The Easysoft ODBC-ODBC Bridge server has turned down the connection request.
- **REFUSED\_MAXCLIENTCONNECT** The connection attempt has been turned down. The Easysoft

ODBC-ODBC Bridge client attempting the connection to the Easysoft ODBC-ODBC Bridge server already has too many open connections. The Easysoft ODBC-ODBC Bridge server limits the connections from a particular client to MaxClientConnect.

- REFUSED\_MAXCONCURRENT The connection attempt has been turned down. There are already too many open connections as defined by MaxThreadCount.
- REFUSED\_BY\_RULE The connection attempt has been turned down due to an client access control rule defined in the server on the **Security** page.
- REFUSED\_BY\_DSN\_RULE The connection attempt has been turned down because of a DSN access control rule. These rules are defined on the **Security** page.
- SERVER\_SUSPENDED The service was suspended.
- SERVER\_RESUMED The service was resumed.

## Daily renaming

The audit files are renamed once a day at midnight. At midnight, the current audit trail file (esoob\_access.log) is renamed to esoob\_access\_<dd>\_<mm>\_<yyy>.log, where <dd> is the day number, <mm> is the month number and <yyy> is the year number.

The audit file for each day should be visible in the web administrator.

## Graph generation

The web administrator is capable of analysing a particular audit file and producing graphs of connections per minute and connections per hour for any selected day.

These graphs are accessed from the **Audit File** link on the **Statistics** page.

Connections are displayed as blue bars and warning events are displayed as coloured warning bars on top of the relevant blue bar.

Failed connects (purple bar):

- AUTH\_DENIED

Attempts denied access due to access control lists or server limits (red bar):

- REFUSED\_MAXCLIENTCONNECT
- REFUSED\_MAXCONCURRENT

Authentication failures (orange bar):

- REFUSED\_BY\_RULE
- REFUSED\_BY\_DSN\_RULE

## Event logging on Windows

The Windows Easysoft ODBC-ODBC Bridge server uses the application event log to record status, diagnostic, and security information. Events logged by the Easysoft ODBC-ODBC Bridge server include:

- The Easysoft ODBC-ODBC Bridge server was started, stopped, paused, or resumed.
- An attempt was made to access the Easysoft ODBC-ODBC Bridge server or a data source that was disallowed by an access control rule.
- An attempt was made to access a web administrator page that was disallowed because an invalid user name or password was supplied.
- The Easysoft ODBC-ODBC Bridge server caught an exception.

Much of the information written to the application event log is also written to the Easysoft ODBC-ODBC Bridge server log files (by default, these are stored in the c:\temp directory). Refer to the application event log if:

- The standard Easysoft ODBC-ODBC Bridge auditing mechanism wasn't enabled for the period that you're interested in.
- You prefer to examine log file entries by using **Event Viewer** rather than a text editor.

## Browsing system data sources in the web administrator

The **Data Sources** page displays the ODBC system data sources found by the server and their configured drivers. Note that if **AllowDSNBrowse** is turned off, no data source or driver information will display in this page.

## Viewing data sources, tables, columns, and data

The web administrator lets you browse data sources for lists of tables, table composition, and rows of data in the tables.

In order to browse system data sources on the machine where the Easysoft ODBC-ODBC Bridge server is running you must turn on the AllowDBBrowse configurable parameter on the **Configuration** page.

This converts the list of data source names on the **Data Sources** page into links to further pages, allowing you to browse down through the data source and its tables, columns, and data.

You may be prompted for a database user name and password when first clicking on a DSN.

The Easysoft ODBC-ODBC Bridge server attempts an initial connection without the ODBC UID and PWD attributes, but if that fails with an authentication error (28000) an authentication challenge will be issued, where the realm is the name of the DSN.

**Note** DSNs cannot be browsed using trusted connections.

Having gained access, DSN details can then be displayed.

## The Licenses page

The page contains a table that shows:

- **Product** The name of the product this license applies to.
- **Version** The version of the product.
- **Created** The date the license was installed.
- **Expires** The date the license expires or the string "Never", meaning the license does not expire.
- **Quantity** A product specific limit that's either the maximum allowable number of concurrent connections or the string "unlimited", meaning that there is no limit on the number of concurrent connections.

## Exporting licenses

On Windows, the **Information** page also lets you export the registry entries in RegEdit 4 format to a file called *LogDir\easysoft\_licenses.reg* where **LogDir** is an Easysoft ODBC-ODBC Bridge server configurable parameter defined on the **Configuration** page.

Choose **Export** to export your licenses. If your registry becomes damaged, you can restore the license entries by double-clicking on the *easysoft\_licenses.reg* file.

## About the Easysoft ODBC-ODBC Bridge

The Easysoft ODBC-ODBC Bridge lets applications that support ODBC connect to a datastore that's on a different platform or has a different architecture. For example, an application on Linux needs to connect to a database for which there's only a Windows ODBC driver; a 64-bit application needs to connect to a database for which there's only a 32-bit ODBC driver.

- [ODBC versions supported](#)
- [Unsupported ODBC 3.5 functionality](#)
- [Modifications to the ODBC API](#)
- [Understanding ODBC diagnostic messages](#)
- [Implementing ODBC diagnostics](#)

## ODBC versions supported

The Easysoft ODBC-ODBC Bridge supports most of ODBC 2.0 and ODBC 2.5, and all of ODBC 3.x apart from SQL\_IS\_POINTER.

All the ODBC 2.0 and 2.5 API functions required to run Perl DBD:ODBC, PHP, and Python mxODBC are present, but the full API of these deprecated versions is not supported, as new applications will be written using the ODBC 3.x API.

## Unsupported ODBC 3.5 functionality

The use of driver-specific pointer types (SQL\_IS\_POINTER) is not supported in calls to:

- [SQLSetConnectAttr](#)
- [SQLSetStmtAttr](#)
- [SQLSetDescField](#)

## Modifications to the ODBC API

### SQLBrowseConnect

The semantics of [SQLBrowseConnect](#) are slightly difficult in the context of the Easysoft ODBC-ODBC Bridge. Normally, SQLBrowseConnect provides an iterative method of discovering and enumerating the attributes and attribute values required to connect to a data source. Each call to SQLBrowseConnect returns successive levels of attributes and attribute values. When all levels have been enumerated a connection to the data source is completed and a complete connection string is returned by SQLBrowseConnect.

This process works fine when SQLBrowseConnect is called for a driver on the local machine, but when you introduce a bridge there are in effect two levels of browsing. The first level will browse the local data sources (defined in your `odbc.ini` file or in the registry), but these point to another real data source on a remote machine.

SQLBrowseConnect only supports the browsing of Easysoft ODBC-ODBC Bridge data sources on the local machine. If the user tries to connect to one of these then they will be prompted for Easysoft ODBC-ODBC Bridge attributes such as `SERVERPORT`, `LOGINUSER`, `LOGINAUTH`, `TARGETUSER`, and `TARGETAUTH`.

Once sufficient attributes have been defined to allow a bridge connection to the server, the server side of the Easysoft ODBC-ODBC Bridge will return a list of DSNs retrieved by calling [SQLDataSources](#). The browse stops here and so the remote data source must already be set up with sufficient information to allow a connection.

This implementation avoids complications with possible clashes of attributes between the Easysoft ODBC-ODBC Bridge and the remote ODBC driver that would make it impossible to return a final connection string.



## Understanding ODBC diagnostic messages

The Easysoft ODBC-ODBC Bridge works in conjunction with a number of other types of software, such as ODBC applications, driver managers, drivers, and DBMSs.

If a diagnostic (or error) message displays when a user connects by using the Easysoft ODBC-ODBC Bridge, the error does not necessarily lie in the Easysoft ODBC-ODBC Bridge. The message should usually indicate where the problem lies.

For example, examine the following diagnostic messages:

```
[Easysoft ODBC (Client)]Invalid authorization specification
```

This error was produced because the LogonUser and LogonAuth attributes were invalid and the connection attempt was refused. Only the Easysoft ODBC-ODBC Bridge was involved in this process.

```
[Easysoft ODBC (Server)][Microsoft][ODBC Driver Manager]  
Data source name not found and no default driver specified
```

This error was produced by the Microsoft ODBC driver manager on the Easysoft ODBC-ODBC Bridge server machine because the TargetDSN attribute specified a DSN that did not exist. The last item in square brackets was the ODBC Driver Manager and therefore is the component that generated the error text.

The text is also prefixed with [Easysoft ODBC (Server)], which means that the error occurred with the driver manager at the server end.

```
[Easysoft ODBC (Server)][Microsoft][ODBC SQL Server Driver][SQL Server]  
Login failed for user 'demo'.
```

This error was produced when the TargetUser and TargetAuth specified by the Easysoft ODBC-ODBC Bridge client was passed through the DBMS, which refused the connection.

The last item in square brackets is SQL Server and so you know that SQL Server turned down the connection attempt.

Therefore, if you encounter any error messages when using the Easysoft ODBC-ODBC Bridge, please note the last item specified in square brackets, because that is the component that generated the error text and the error may not lie in the Easysoft ODBC-ODBC Bridge.

If you are integrating the Easysoft ODBC-ODBC Bridge with your own ODBC application, it is worthwhile implementing diagnostics in your application because then your error messages will indicate in which component the problem lies.

## Implementing ODBC diagnostics

This section explains how to implement ODBC diagnostics in your own ODBC application, so that the application can display diagnostic messages if an error occurs at any point in the connection process.

### Status returns

All ODBC APIs return a status value that may be used to check whether the function succeeded or not.

In C, you can test the return value from an ODBC function using the macro `SQL_SUCCEEDED`:

```
SQLRETURN fsts;

/* Assume environment has already been allocated */
SQLHENV envh;
SQLHDBC dbch;

fsts = SQLAllocHandle(SQL_HANDLE_DBC, envh, &dbch);

if (!SQL_SUCCEEDED(fsts))
{
    /* an error occurred allocating the database handle */
}
else
{
    /* Database handle allocated OK */
}
```

The macro `SQL_SUCCEEDED` is defined as:

```
#define SQL_SUCCEEDED(rc) (((rc)&(~1))==0)
```

Virtually all ODBC functions can return two values which indicate success:

- `SQL_SUCCESS`
- `SQL_SUCCESS_WITH_INFO`

Both of these returns cause the `SQL_SUCCEEDED` macro to result in 1. If a function returns `SQL_SUCCESS_WITH_INFO` it means that the call succeeded, but an informational message was produced.

For example, with some drivers you might set the cursor type, prepare a statement, and then execute it. When [SQLExecute](#) is called the statement is acted upon, but the driver might change the cursor type to something else.

In this case, `SQLExecute` would return `SQL_SUCCESS_WITH_INFO` and the driver would add a diagnostic indicating the cursor type had been changed.

You should note that a few ODBC functions return a status that fails the `SQL_SUCCEEDED` macro, but do not necessarily indicate an error. For example, [SQLFetch](#) can return `SQL_NO_DATA` indicating there is no further rows in the result set, this is not necessarily an error.

### Obtaining diagnostics

When an ODBC function returns an error or `SQL_SUCCESS_WITH_INFO`, the ODBC driver will

associate a diagnostic with the handle used in the ODBC call.

You can obtain the diagnostic to find out what failed by calling [SQLGetDiagRec](#) with the handle you used in the ODBC call that failed.

The ODBC driver may associate multiple diagnostic records with a handle. You can call [SQLGetDiagField](#) and request the SQL\_DIAG\_NUMBER attribute to find out how many diagnostics exist.

Alternatively, as diagnostic records start at 1, you can repeatedly call [SQLGetDiagRec](#) asking for record 1, then 2 (and so on) until [SQLGetDiagRec](#) returns SQL\_NO\_DATA.

As an example, the following C function takes a function name string, handle type, and handle and retrieves all the diagnostics associated with that handle.

```
void extract_error
    (char *fn,
     SQLHANDLE handle,
     SQLSMALLINT type)
{
    SQLINTEGER i = 0;
    SQLINTEGER native;
    SQLCHAR state[7];
    SQLCHAR text[256];
    SQLSMALLINT len;
    SQLRETURN ret;

    fprintf(stderr,
            "\n"
            "The driver reported the following diagnostics whilst running "
            "%s\n\n",
            fn);
    do
    {
        ret = SQLGetDiagRec(type, handle, ++i, state, &native, text, sizeof(text),
                           &len);
        if (SQL_SUCCEEDED(ret)) printf("%s:%ld:%ld:%s\n", state, i, native, text);
    }

    while (ret == SQL_SUCCESS);
}
```

Using this example, which attempts to allocate a database handle, you could use `extract_error` as follows:

```
SQLRETURN fsts;

/* Assume environment has already been allocated */
SQLHENV envh;
SQLHDBC dbch;
```

```
fsts = SQLAllocHandle(SQL_HANDLE_DBC, envh, &dbch);

if (!SQL_SUCCEEDED(fsts))
{
    extract_error("SQLAllocHandle for dbc", envh, SQL_HANDLE_ENV);
    exit(1);
}
else
{
    /* Database handle allocated OK */
}
```

ODBC 2.x applications use [SQLError](#) instead of SQLGetDiagRec.

## Diagnostic fields

When you call SQLGetDiagRec, you can retrieve three diagnostic fields:

- State
- Native error code
- Message text

The state is a five character SQLSTATE code. The first two characters indicate the class and the next three indicate the subclass. SQLSTATE codes provide detailed information about the cause of a warning or error.

The native error code is a code specific to the data source. This number is often extremely useful to the driver developers in locating an internal error or state.

The message text is the text of the diagnostic.

This string takes one of two forms:

For errors and warnings that do not occur in a data source the format is:

```
[vendor-identifier]component-supplied-text
```

otherwise it's:

```
[vendor-identifier][ODBC-component-identifier]data-source-supplied-text
```

# Index

## B

Block Statement Attributes attribute, [41](#)  
BlockFetchSize attribute, [39](#)

## C

Certificate File attribute, [42](#)  
connecting to ODBC data sources, [36](#)  
Connection Attempts attribute, [40](#)  
connection failover, [43](#)

## CPUs

support for multiple, [28](#)

## D

data source attributes, [38](#)  
DecAsNumeric attribute, [43](#)  
Description attribute, [38](#)  
Disguise wide characters attribute, [39](#)  
DSN attribute, [38](#)

## E

Easysoft ODBC-ODBC Bridge  
  configuring, [21](#), [30](#)  
  connecting to an ODBC data source, [36](#)  
  data source attributes, [38](#)  
  fallback servers, [43](#)  
  inetd and, [33](#)  
  installing, [6](#)  
    on Linux or UNIX, [8](#)  
    on Windows, [6](#)  
  ODBC API support, [61](#)  
  overview, [5](#)  
  reporting and statistics, [54](#)  
  restricting access to, [29](#)  
  testing with oobping, [46](#)  
  tracing, [53](#)  
  uninstalling  
    on Linux or UNIX, [18](#)  
    on Windows, [8](#)  
  web administrator, [21](#)

Encrypt Communication attribute, [42](#)

## I

inetd, [33](#)  
installing the Easysoft ODBC-ODBC Bridge, [6](#)

## M

Map SQLExecDirect attribute, [40](#)  
MetaData\_ID\_Identifier attribute, [43](#)  
MetaDataBlockFetch attribute, [39](#)

## N

network interface cards  
  support for multiple, [28](#)

## O

ODBC connection pooling, [28](#)  
ODBC data sources  
  connecting to, [36](#)  
  restricting access to, [29](#)  
ODBC diagnostics, [65](#)  
oobping  
  testing the Easysoft ODBC-ODBC Bridge with,  
    [46](#)  
Override Length in SQLGetxxx attribute, [40](#)  
Override UID/PWD attribute, [39](#)

## P

Pass all SQLGetInfo requests on attribute, [39](#)  
Password attribute, [38](#)

## S

Servers attribute, [38](#)  
SQL  
  log statements that fail, [56](#)

## T

TargetAuth attribute, [38](#)  
TargetDSN attribute, [38](#)  
TargetUser attribute, [38](#)  
tracing, [53](#)

## U

uninstalling  
  on Linux or UNIX, [18](#)  
  on Windows, [8](#)  
Unquote\_Catalog\_Fns attribute, [42](#)  
Username attribute, [38](#)

## W

web administrator, [21](#)