

Easysoft ODBC-JDBC Gateway User's Guide

This manual documents version 2.8.n of the Easysoft ODBC-JDBC Gateway.

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-JDBC Gateway | 6 |
| Installing on Windows | 6 |
| Uninstalling on Windows | 8 |
| Installing on Linux or UNIX | 8 |
| Uninstalling on Linux or UNIX | 16 |
| Defining the JVM to use | 17 |
| Defining the JVM on Windows | 17 |
| Defining the JVM on Linux and UNIX | 17 |
| Setting JVM Options | 18 |
| Connecting to your Java data store | 19 |
| Connecting from Linux or UNIX | 19 |
| Connecting from Windows | 20 |
| Connection attributes | 21 |
| DSN-less connections | 28 |
| Logging | 29 |
| ODBC Driver Manager logging on Linux or UNIX | 29 |
| ODBC Driver Manager logging on Windows | 29 |
| Finding out what product version you have on Windows | 32 |
| Client applications | 33 |
| Microsoft Access | 34 |
| Linking a table | 34 |
| Importing a table | 34 |
| Microsoft Excel | 35 |
| Data Connection Wizard | 35 |
| Microsoft Query | 35 |
| PowerPivot | 35 |
| Microsoft Power BI | 36 |
| Tableau Desktop | 37 |
| Microsoft SQL Server | 38 |
| Oracle | 40 |
| Connecting ODBC-JDBC Gateway to Oracle on Windows | 40 |
| Connecting ODBC-JDBC Gateway to Oracle on Linux and UNIX | 41 |
| LibreOffice | 44 |
| Go | 45 |
| Node.js | 47 |
| Perl | 49 |
| PHP | 52 |
| Python | 54 |
| R | 56 |
| About the Easysoft ODBC-JDBC Gateway | 58 |
| ODBC API support | 59 |
| Unicode ODBC calls | 61 |
| XA support | 62 |
| The xa_open string | 62 |
| The xa_switch_t structure | 63 |
| Index | 64 |

Getting started

- [Overview](#)
- [Installing the Easysoft ODBC-JDBC Gateway](#)
- [Defining the JVM to use](#)
- [Connecting to your Java data store](#)
- [Logging](#)

Overview

The Easysoft ODBC-JDBC Gateway allows applications that support ODBC to connect to any data store that supports JDBC.

The Easysoft ODBC-JDBC Gateway uses a JDBC driver to connect to the data store. You specify which JDBC driver to use by configuring an ODBC data source for the Easysoft ODBC-JDBC Gateway.

For example, you want to connect SQL Server to Data Lake. You would create a Easysoft ODBC-JDBC Gateway ODBC data source that points at the JDBC driver for Data Lake. You would then tell SQL Server to use the ODBC data source, which then connects to Data Lake by using JDBC.

Installing the Easysoft ODBC-JDBC Gateway

Install the Easysoft ODBC-JDBC Gateway on the computer where the application you want to connect to your Java data store is running.

- [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-JDBC Gateway 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-JDBC Gateway 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-JDBC Gateway 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-JDBC Gateway 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-JDBC Gateway. 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-JDBC Gateway:

1. In License Manager, enter your contact details.

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

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

2. Choose **Request License**.

You're prompted to choose a license type.

3. Do one of the following:

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

-Or-

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

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

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

6. Choose how to get your license when prompted.

7. Do one of the following:

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

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

-Or-

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

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

-Or-

- Choose **Email Request** to email your license request to our licensing team.

Once we've processed your request, we'll email your license to the email address you supplied in License Manager.

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

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

Once you've licensed the Easysoft ODBC-JDBC Gateway, the installation is complete.

Repairing the installation

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

Uninstalling on Windows

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

Removing the Easysoft ODBC-JDBC Gateway

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

| | |
|-------------|---|
| Note | Easysoft product licenses are stored in the Windows registry. When you uninstall, your licenses are not removed, so you do not need to relicense the product if you reinstall or upgrade. |
|-------------|---|

Installing on Linux or UNIX

The installation can be done by anyone with root access.

1. [Download the Easysoft ODBC-JDBC Gateway 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-JDBC Gateway is installed.
3. Unpack the distribution and cd into the resultant directory.
4. As root, run:

```
./install
```

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

Further information

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

Preinstallation requirements

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

You need an ODBC Driver Manager to use the Easysoft ODBC-JDBC Gateway 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-JDBC Gateway 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-JDBC Gateway 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-jdbc-gateway-2.8.0-linux-x86-64-ubuntu164.tar
```

This creates a directory with the same name as the tar file (without the .tar postfix) containing further archives, checksum files, an installation script, and various other installation files.

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

```
# cd odbc-jdbc-gateway-2.8.0-linux-x86-64-ubuntu164
```

License to use

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

Answering questions during the installation

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

For example:

```
Do you want to continue? (y/n) [n]:
```

The possible answers to this question are y or n. The default answer when you type nothing and press Enter is n.

Running the installer

If you are considering running the installation as a non-root user, we suggest you review this carefully as you will have to get a root user to manually complete some parts of the installation afterwards. We recommend installing as the root user. (If you're concerned about the changes that will be made to your system, refer to [Changes made to your system.](#))

To start the installation, run:

```
./install
```

You need to:

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

We recommend accepting the default installation path.

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

Locating or installing unixODBC

We strongly recommend you use the unixODBC Driver Manager because:

- The installation script is designed to work with unixODBC and can automatically add Easysoft ODBC-JDBC Gateway and data sources during the installation.
- Most applications and interfaces that support ODBC are compatible with unixODBC. The Easysoft ODBC-JDBC Gateway 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-JDBC Gateway 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-JDBC Gateway 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-JDBC Gateway installation script:

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

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

```
Would you like to request a Easysoft ODBC-JDBC Gateway 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-JDBC Gateway), 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-JDBC Gateway 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-JDBC Gateway 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-JDBC Gateway in the default location, you need to do one of the following:

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

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

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

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

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

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

Uninstalling on Linux or UNIX

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

```
odbcinst -u -d -n Easysoft ODBC-JDBC Gateway
```

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

Defining the JVM to use

Defining the JVM on Windows

The Easysoft ODBC-JDBC Gateway uses a JVM to load the target JDBC driver. When you install the Easysoft ODBC-JDBC Gateway, Setup searches for a compatible JVM. To check which JVM the Easysoft ODBC-JDBC Gateway is using, run **Configure Java Interface** from the Easysoft ODBC-JDBC Gateway start menu group.

You should also run Configure Java Interface if:

- You want to use a different or newer JVM with the Easysoft ODBC-JDBC Gateway. For example, you may want to do this if you have updated your JVM since you ran Setup.
- You ran Easysoft ODBC-JDBC Gateway Setup without a JVM being installed.

Note To obtain a JVM, download the JRE from either the Oracle web site or the IBM web site. The JVMs included with these JREs have both been tested with the Easysoft ODBC-JDBC Gateway.

If you want to use a 64-bit application with the Easysoft ODBC-JDBC Gateway, you need to use the 64-bit Easysoft ODBC-JDBC Gateway with a 64-bit JVM.

If you want to use a 32-bit application with the Easysoft ODBC-JDBC Gateway, you need to use the 32-bit Easysoft ODBC-JDBC Gateway with a 32-bit JVM.

Note that whether you use a 32-bit or 64-bit JVM is irrelevant to the target JDBC driver.

1. If your application is 64-bit, run the 64-bit version of the Configure Java Interface tool. If your application is 32-bit, run the 32-bit version of the Configure Java Interface tool.

If the JVM Library Path box is empty or you want to use a different JVM, follow steps 2 and 3. Otherwise, skip to step 4.

2. If you know the location of the JVM that you want the Easysoft ODBC-JDBC Gateway to use, in the **JVM Library Path** box, enter the path of the JVM. For example:

C:\Java\jvm.dll

3. Alternatively, to browse for the JVM, click the ... button. In the **Select JVM** dialog box, choose the **Browse**. In the **Known JVM Libs** list, double-click the JVM.

If you are unsure where the JVM is installed, choose the ... button. In the **Select JVM** dialog box, choose **Search** and then browse to the directory you want to search. The search results are displayed in the **Known JVM Libs** list. In this list, double-click the JVM that you want to use. If the list is empty, search a different directory for JVMs.

4. To test whether the JVM is compatible with the Easysoft ODBC-JDBC Gateway, choose **Test and Save if OK**.

If the test results show that the JVM is incompatible and you have more than one JVM installed, choose **Cancel** to exit Configure Java Interface. Start Configure Java Interface, click the ... button and then double-click a different JVM. Choose **Test and Save if OK** to check the new JVM.

If your JVM is incompatible and you do not have another JVM installed, you need to obtain a different one.

Defining the JVM on Linux and UNIX

The Easysoft ODBC-JDBC Gateway installation script searches for a compatible JVM during setup.

18 Setting JVM Options

This is stored as JvmPath in the entry for Easysoft ODBC-JDBC Gateway in odbinst.ini (normally located in /etc.) For example:

```
[Easysoft ODBC-JDBC Gateway]
Driver = /usr/local/easysoft/ojg/libo2jg.so
Setup = /usr/local/easysoft/ojg/libo2jgS.so
JvmPath = /usr/jdk1.3.1/jre/lib/i386/client/libjvm.so
```

To specify a different JVM, edit this file.

Setting JVM Options

Usually, you do not need to change any JVM configuration options to use a JVM with the Easysoft ODBC-JDBC Gateway. If you do need to specify any JVM settings, the Easysoft ODBC-JDBC Gateway provides the following ways to do this:

On Windows:

1. Start Configure Java Interface from the Easysoft ODBC-JDBC Gateway start menu group.

To set options for a 64-bit JVM, choose **Configure Java Interface (64-bit)**. To set options for a 32-bit JVM, choose **Configure Java Interface**.

2. In the JVM Options box, type the JVM options.

On Linux and UNIX:

- Add this entry to the [Easysoft ODBC-JDBC Gateway] section in odbinst.ini:

```
JVMOPTIONS = jvmoption
```

Separate multiple JVM options with a space. For example:

```
-Xmx2048M -Xms1024M
```

For more information about the available options, refer to the documentation for your JVM.

JVM options specified in the **Configure Java Interface** dialog box or odbinst.ini apply to all Easysoft ODBC-JDBC Gateway data sources. On Linux and UNIX, you can override the global JVM options for a particular data source, by setting the JVM_OPTIONS environment variable. For example:

```
JVM_OPTIONS="-Xmx2048M -Xms1024M"
export JVM_OPTIONS
```

Connecting to your Java data store

The Easysoft ODBC-JDBC Gateway uses a JDBC driver to connect to the data store. You specify which JDBC driver to use by configuring an ODBC data source for the Easysoft ODBC-JDBC Gateway.

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

Connecting from Linux or UNIX

Creating an ODBC data source

There are two ways to create a data source to your ODBC-JDBC Gateway data:

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

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

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

Your Easysoft ODBC-JDBC Gateway data source in `odbc.ini` must contain these attributes:

- The directory where the JDBC driver's `.jar` file is located (ClassPath)
- The JDBC driver's class name (DriverClass)
- The JDBC driver's connection URL (URL)

For example:

```
[ODBC_JDBC_SAMPLE]
```

20 Connecting from Windows

```
Driver      = Easysoft ODBC-JDBC Gateway
ClassPath   = /tmp/infor-compass-jdbc.jar
DriverClass = com.infor.idl.jdbc
URL         = jdbc:infordatalake://my_tenant
```

The Easysoft ODBC-JDBC Gateway must be able to find the following shared objects:

- libodbcinst.so

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

- libeslicshr.so

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

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

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

The isql query tool lets you test your Easysoft ODBC-JDBC Gateway data sources. To test the Easysoft ODBC-JDBC Gateway connection:

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

```
SQL> select * from Customers;
```

–Or–

4. Enter help to return a list of tables:

```
SQL> help
```

Connecting from Windows

Creating an ODBC data source

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

```
odbcad32.exe
```

–Or–

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

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

Note

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

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

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

Connection attributes

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

Setting on Linux and UNIX

Your Easysoft ODBC-JDBC Gateway data source in odbc.ini must contain these attributes:

- ClassPath
- DriverClass
- URL

For example:

```
[ODBC_JDBC_SAMPLE]
Driver           = Easysoft ODBC-JDBC Gateway
ClassPath        = /tmp/infor-compass-jdbc.jar
DriverClass      = com.infor.idl.jdbc
URL              = jdbc:infordatalake://my_tenant
```

These optional attributes may be set in odbc.ini.

- Async_Cancel
- BigInt2Char
- CanDescribeParam
- Clean_Metadata
- DisableMoreResults
- Description
- Password

- ReuseCL
- Single_Statement
- Strip_Escape
- Strip_Quote
- User
- With_Schema
- XAClass
- XA_Connection_String
- XA_Enlist
- XIDClass
- wchardefaultc

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

| Name | Value |
|----------------------|--|
| CanDescribeParam | <p>When CanDescribeParam is turned on, the Easysoft ODBC-JDBC Gateway provides support for describing parameters if the target JDBC driver supports PreparedStatement.getParameterMetaData</p> <p>By default, CanDescribeParam is turned off.</p> |
| XA_Enlist | <p>When XA_Enlist is turned on, the Easysoft ODBC-JDBC Gateway uses the XA interface to access the JDBC driver. This makes the XA connection available for use by the Easysoft ODBC-JDBC Gateway and any work done by the gateway is under the control of the Transaction Manager. (Your ODBC application also needs to turn off the ODBC auto-commit mode by using SQLSetConnectAttr with the SQL_ATTR_AUTOCOMMIT attribute.)</p> <p>If you want to use the Easysoft ODBC-JDBC Gateway in the context of a distributed XA transaction, turn on the XA_Enlist option. Otherwise, leave the option set to its default value, which is turned off.</p> |
| XA_Connection_String | <p>The DB field value in the xa_open string. For example, you specify a database named "payroll" with the following xa_open string clause:</p> <pre>DB=payroll</pre> <p>you therefore also need to specify "payroll" as the value for the XA_Connection_String setting:</p> <pre>XA_Connection_String=payroll</pre> |

| Name | Value |
|----------|--|
| XAClass | <p>The class the JDBC driver implements the XADataSource interface with.</p> <p>Examples</p> <p>For the Oracle JDBC driver, use:</p> <pre>XAClass = oracle.jdbc.xa.client.OracleXADataSource</pre> <p>For the Microsoft SQL Server JDBC driver, use:</p> <pre>XAClass = com.microsoft.sqlserver.jdbc.SQLServerXADataSource</pre> |
| XIDClass | <p>The method the JDBC driver provides to create XA transaction ids (Xid). The Transaction Manager uses Xids to coordinate the branches of a distributed transaction.</p> <p>Examples</p> <p>For the Oracle JDBC driver, use:</p> <pre>XIDClass = oracle.jdbc.xa.OracleXid</pre> <p>For the SQL Server JDBC driver, use:</p> <pre>XIDClass = com.microsoft.sqlserver.jdbc.XidImpl</pre> |

Setting on Windows

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

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

| Name | Value |
|--------------|---|
| User Name | <p>The name of a database user to pass to the JDBC driver, if required.</p> <p>The attribute value is passed to the JDBC driver as part of a Properties object in the connect call.</p> <p>The equivalent Java code is:</p> <pre> Connection con; Properties prop; if (uid != null) { prop.put("user", uid); } con = DriverManager.getConnection(url, p); </pre> <p>This value can be overridden at the ODBC level by passing either a non-NULL user name in the call to SQLConnect or a UID=<i>value</i> attribute in the connection string passed to SQLDriverConnect.</p> <p>It can also be supplied in the JDBC URL supplied to the JDBC driver, but the syntax of this is JDBC driver dependent.</p> |
| Password | <p>The database password to pass to the JDBC driver, if required.</p> <p>The attribute value is passed to the JDBC driver as part of a Properties object in the connect call.</p> <p>The equivalent Java code is:</p> <pre> Connection con; Properties prop; if (pwd != null) { prop.put("password", pwd); } con = DriverManager.getConnection (url, p); </pre> <p>This value can be overridden at the ODBC level by passing either a non-NULL user name in the call to SQLConnect or a PWD=<i>value</i> attribute in the connection string passed to SQLDriverConnect.</p> <p>It can also be supplied in the JDBC URL supplied to the JDBC driver, but the syntax of this is JDBC driver dependent.</p> |
| Driver Class | <p>The class name of the target JDBC driver. For example, to connect to the Data Lake JDBC driver, you'd set Driver Class to</p> <pre>com.infor.idl.jdbc</pre> <p>The equivalent Java code is:</p> <pre>Class.forName(<i>driverClass</i>);</pre> |

| Name | Value |
|------------------|--|
| Class Path | <p>A list of the .jar files required to enable Java to load the JDBC driver class. Separate multiple .jar files with a ; on Windows and : on Linux and UNIX.</p> <p>Note that the order in which the .jar files are specified can be significant.</p> |
| URL | <p>The JDBC URL required to connect to the target database</p> <p>The JDBC URL string is passed as the first argument to:</p> <pre>Connection con; con = DriverManager.getConnection(url, p);</pre> <p>For information about the JDBC URL syntax, refer to the JDBC driver documentation.</p> <p>Here's an example for the Data Lake JDBC driver:</p> <pre>jdbc:infordatalake://my_tenant</pre> |
| Strip Quote | <p>Some JDBC drivers can't accept double quotes around column and table names, and in some cases the quotes will result in an error, as they may invoke unwanted case-sensitive behaviour.</p> <p>As an example, when Strip Quote is turned on, the following SQL:</p> <pre>SELECT "A" FROM "T"</pre> <p>will, when sent to the JDBC driver, be transformed to:</p> <pre>SELECT A FROM T</pre> <p>By default, Strip Quote is turned off.</p> |
| Single Statement | <p>Some JDBC drivers are only capable of using a single active result set, but may return a value of either zero or greater than one from <code>DatabaseMetaData.getMaxStatements()</code>.</p> <p>When turned on, Single Statement forces the ODBC driver to return a value of 1 from the SQLGetInfo call to find <code>SQL_MAX_CONCURRENT_ACTIVITIES</code>.</p> <p>By default, Single Statement is turned off.</p> |
| Modify Metadata | <p>Some JDBC drivers only return a partial result set from calls such as DatabaseMetaData.getTypeInfo().</p> <p>When turned on, Modify Metadata causes the ODBC driver to modify the values in the result set returned from metadata calls (such as SQLGetInfo) so that they conform to the expected result set specification.</p> <p>By default, Modify Metadata is turned off.</p> |

| Name | Value |
|-----------------|--|
| Reuse CL Object | <p>When loading a JDBC driver, the Easysoft ODBC-JDBC Gateway creates a java.net.URLClassLoader and this is then used to load the JDBC driver.</p> <p>When Reuse CL Object is turned on, the URLClassLoader object is retained between calls to SQLConnect and SQLDriverConnect, improving connection times and reducing system resource consumption.</p> <p>Reuse CL Object should only be turned off if problems are encountered connecting to multiple different JDBC drivers.</p> <p>By default, Reuse CL Object is turned on.</p> |
| Strip Escape | <p>While some JDBC drivers can accept ODBC-type escape sequences, some JDBC drivers are unable to understand them.</p> <p>When Strip Escape is turned on, the ODBC driver can modify the SQL passed to the JDBC driver, removing the ODBC sequences.</p> <p>For example, the following SQL when passed to the ODBC driver:</p> <pre>SELECT * FROM T WHERE DATE_FIELD = {d '2025-02-01'}</pre> <p>will be altered to:</p> <pre>SELECT * FROM T WHERE DATE_FIELD = '2025-02-01'</pre> <p>when Strip Escape is turned on.</p> <p>By default, Strip Escape is turned off.</p> |
| Bigint Default | <p>The ODBC specification states that if SQL_C_DEFAULT is used in combination with SQL_BIGINT fields, the result returned will be in SQL_C_BIGINT format.</p> <p>When Bigint Default is turned on, a CHAR string is returned under these conditions, as some applications (Microsoft Access in particular) do not know of the SQL_BIGINT data type and therefore expect the data to be returned as a string.</p> <p>By default, Bigint Default is turned off.</p> |
| Async Cancel | <p>By default, the Easysoft ODBC-JDBC Gateway does not call PreparedStatement.Cancel() and ResultSet.Cancel() when the ODBC function SQLCancel is called. Several JDBC Drivers will fail if this is done in separate threads, which the ODBC specification allows.</p> <p>When Async Cancel is turned on, the Easysoft ODBC-JDBC Gateway does call the .Cancel() methods.</p> <p>By default, Async Cancel is turned off.</p> |

| Name | Value |
|---------------------|---|
| WCHAR Default | <p>The ODBC specification allows applications to request the format in which data is returned, and for each SQL data type there is a default type. The specification says that for a wide character field, the default return type is a SQL_WCHAR, a Unicode data type.</p> <p>By default, the Easysoft ODBC-JDBC Gateway map a SQL_WCHAR column when requested as a SQL_DEFAULT to a SQL_WCHAR, in accordance with the ODBC specification.</p> <p>However, for tables with one or more WCHAR fields that are part of the primary key of that table, Microsoft Access expects that conversion to result in SQL_CHAR data. The symptom for this is that the table opens, but displays #deleted in all the fields.</p> <p>When WCHAR Default is turned on, the Easysoft ODBC-JDBC Gateway uses the SQL_CHAR conversion for SQL_DEFAULT requests on a SQL_WCHAR type. This is the conversion that Access expects.</p> <p>This option is a workaround for Access. Enabling WCHAR Default causes the Easysoft ODBC-JDBC Gateway to behave in an non-standard way and may cause problems with other applications. If this is the case, create a separate ODBC data source for use with Access and only turn on WCHAR Default in that data source.</p> <p>By default, WCHAR Default is turned off.</p> |
| Disable MoreResults | <p>By default, the Easysoft ODBC-JDBC Gateway maps a SQLMoreResults call to a statement.getMoreResults call. However, some JDBC drivers do not properly implement getMoreResults. Turning on Disable MoreResults allows the JDBC driver to always return SQL_NO_DATA from a SQLMoreResults call.</p> <p>When turned on, the Easysoft ODBC-JDBC Gateway does not call getMoreResults in the JDBC driver.</p> <p>By default, Disable MoreResults is turned off.</p> |

DSN-less connections

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

Here's an example Easysoft ODBC-JDBC Gateway connection string:

```
DRIVER={Easysoft ODBC-JDBC  
Gateway};DRIVERCLASS=com.infor.idl.jdbc;CLASSPATH=/tmp/infor-compass-  
jdbc.jar;URL=jdbc:infordatalake://my_tenant
```

For a list of the other attributes you can set in the connection string, refer to the earlier topics in this section.

Logging

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

To turn on logging, refer to the following sections.

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

ODBC Driver Manager logging on Linux or UNIX

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

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

For example:

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

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

ODBC Driver Manager logging on Windows

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

```
odbcad32.exe
```

-Or-

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

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

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

Manager, expand the process list for **Windows Command Processor** or **Windows PowerShell**, as appropriate, and check whether the process for your programming language contains (32-bit).

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

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

6. Choose **Start Tracing Now**.

Note

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

Finding out what product version you have on Windows

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

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

The product version displays below.

Client applications

How to work with ODBC-JDBC Gateway data in some example applications and programming languages:

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

Microsoft Access

1. [Install the Easysoft ODBC-JDBC Gateway](#) on same computer as Access.
2. [Configure an ODBC data source](#).

Linking a table

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

Importing a table

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

Microsoft Excel

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

Data Connection Wizard

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

Microsoft Query

1. Choose **Data > Get Data > From Other Sources > From Microsoft Query**.
2. In the **Choose Data Source** dialog box, choose your ODBC-JDBC Gateway data source from the list, and then choose **OK**.
3. If you want to change the sort order of your data, use the **Sort Order** screen to sort the results of your query (this is the equivalent of a SQL ORDER BY clause), and then choose **Next**. Choose **Finish** to return your ODBC-JDBC Gateway data to Excel.

PowerPivot

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

Microsoft Power BI

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

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

Note

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

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

Your ODBC-JDBC Gateway data is now available to use in Power BI visualisations.

Tableau Desktop

1. [Install the Easysoft ODBC-JDBC Gateway](#) on same computer as Access.
2. [Configure an ODBC data source](#).
3. In Tableau Desktop, choose **Other Databases (ODBC)**.
4. In the **Other Databases (ODBC)** dialog box, choose your Easysoft ODBC-JDBC Gateway data source.
5. Choose **Connect**, and then choose **Sign in**
6. Select a database from the **Database** list.
7. Double-click **New Custom SQL**.
8. In the **Edit Custom SQL** box, enter an SQL query, and then choose **OK**.
9. Choose **Update Now**.

Microsoft SQL Server

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

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

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

SQL Server verifies the linked server by testing the connection.

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

A four part table name has the format:

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

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

```
EXEC sp_tables_ex @table_server = 'ODBC-JDBC Gateway'
```

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

```
SELECT * FROM [ODBC-JDBC Gateway]..DBO.MyTable
```

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

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

```
SELECT * FROM OPENQUERY([ODBC-JDBC Gateway], 'SELECT * FROM MyTable')
```

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

```
-- In Server Options, set both RPC and RPC Out to `True`.
EXEC ('INSERT INTO MyTable (MyCol1, MyCol2, MyCol3, MyCol4, MyCol5)
VALUES (''MyValue1'' , ''MyValue2'' , ''MyValue3'' , ''MyValue4'' ,
''MyValue5'')')
AT ODBC-JDBC Gateway

UPDATE OPENQUERY ([ODBC-JDBC Gateway], 'SELECT MyCol1 FROM MyTable WHERE
MyCol1 = ''MyValue1'') SET MyCol1='MyNewValue'
DELETE OPENQUERY (ODBC-JDBC Gateway, 'SELECT MyCol1 FROM MyTable WHERE MyCol1
= ''MyValue1'')
```

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

Oracle

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

Connecting ODBC-JDBC Gateway to Oracle on Windows

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

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

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

```
HS_FDS_CONNECT_INFO = "ODBC-JDBC Gateway"
```

Replace ODBC-JDBC Gateway with the name of your Easysoft ODBC-JDBC Gateway data source.

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

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

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

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

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

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

Replace oracle_host with the host name of your Oracle machine.

6. Start (or restart) the Oracle Listener:

```
cd %ORACLE_HOME%\bin
```



```
lsnrctl stop
lsnrctl start
```

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

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

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

9. Try working with your ODBC-JDBC Gateway data. For example:

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

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

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

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

Notes

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

Connecting ODBC-JDBC Gateway to Oracle on Linux and UNIX

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

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

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

```
HS_FDS_CONNECT_INFO = "ODBC-JDBC Gateway"
```

Replace ODBC-JDBC Gateway with the name of your Easysoft ODBC-JDBC Gateway data source.

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

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

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

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

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

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

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

Replace oracle_host with the host name of your Oracle machine.

6. Start (or restart) the Oracle Listener:

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

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

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

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

9. Try working with your ODBC-JDBC Gateway data. For example:

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

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

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

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

Notes

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

LibreOffice

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

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

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

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

Go

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

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

4. Create and then use Go to run this script, which retrieves some ODBC-JDBC Gateway data:

```
package main

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

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

    var (
        name string
    )

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

    if err != nil {
        log.Fatal(err)
    }
    defer rows.Close()
    for rows.Next() {
        err := rows.Scan(&name)
        if err != nil {
            log.Fatal(err)
        }
        log.Println(name)
    }

    err = rows.Err()
    if err != nil {
        log.Fatal(err)
    }

    defer db.Close()
}
```

```
}
```



Node.js

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

```
npm install odbc
```

4. Create and then use Node.js to run this script, which retrieves some ODBC-JDBC Gateway data:

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

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

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

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

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

7. These scripts insert, update, and then delete some ODBC-JDBC Gateway data:

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

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

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

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


Perl

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

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

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

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

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

5. Create and then use Perl to run this script, which retrieves some ODBC-JDBC Gateway data:

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

my $sql = "SELECT MyCol1 FROM MyTable";

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

$sth->execute();

my($Col);

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

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

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

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

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

my @row;

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

```

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

    $dbh->disconnect if ($dbh);

```

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

```

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

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

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

$dbh->disconnect if ($dbh);

```

8. These scripts insert, update, and then delete some ODBC-JDBC Gateway data:

```

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

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

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

$dbh->disconnect if ($dbh);

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

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

$sth->execute('MyValue1');

$dbh->disconnect if ($dbh);

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

my $sth = $dbh->prepare('DELETE FROM MyTable WHERE MyCol1 = ?')

```

```
        or die "Can't prepare statement: $DBI::errstr";  
  
$sth->execute('MyValue1');  
  
$dbh->disconnect if ($dbh);
```

Further information

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

PHP

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

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

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

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

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

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

7. These scripts insert, update, and then delete some ODBC-JDBC Gateway data:

```
<?php
$cnx = odbc_connect("ODBC-JDBC Gateway", "", "");
$stmt = odbc_prepare($cnx, "INSERT INTO MyTable (MyCol1, MyCol2, MyCol3,
MyCol4, MyCol5) VALUES (?, ?, ?, ?, ?)");
```

```
$success = odbc_execute($stmt, array('MyValue1', 'MyValue2', 'MyValue3',  
'MyValue4', 'MyValue5'));  
odbc_close($cnx);  
?>  
  
<?php  
$cnx = odbc_connect("ODBC-JDBC Gateway", "", "");  
$stmt = odbc_prepare($cnx, "UPDATE MyTable SET MyCol1 = 'MyNewValue' WHERE  
MyCol1 = ?");  
$success = odbc_execute($stmt, array('MyValue1'));  
odbc_close($cnx);  
?>  
  
<?php  
$cnx = odbc_connect("ODBC-JDBC Gateway", "", "");  
$stmt = odbc_prepare($cnx, "DELETE FROM MyTable WHERE MyCol1 = ?");  
$success = odbc_execute($stmt, array('MyValue1'));  
odbc_close($cnx);  
?>
```

Further information

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

Python

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

```
pip list
```

If you don't have pip installed:

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

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

```
pip install pyodbc
```

5. Create and then use Python to run this script, which retrieves some ODBC-JDBC Gateway data:

```
import pyodbc

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

cnxn.close()
exit()
```

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

```
import pyodbc

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

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

```
import pyodbc

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

8. These scripts insert, update, and then delete some ODBC-JDBC Gateway data:

```
import pyodbc

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

```
import pyodbc

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

```
import pyodbc

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

Further information

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

R

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

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

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

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

5. Create and then use R to run this script, which retrieves some ODBC-JDBC Gateway data:

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

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

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

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

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

8. These scripts insert, update, and then delete some ODBC-JDBC Gateway data:

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



```
quit()

library("RODBC")
ch <- odbcConnect("ODBC-JDBC Gateway")
sqlQuery(ch, paste("UPDATE MyTable SET MyCol1 = 'MyNewValue' WHERE MyCol1 = 'MyValue1'"))
odbcClose(ch)
quit()

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

About the Easysoft ODBC-JDBC Gateway

The Easysoft ODBC-JDBC Gateway allows applications that support ODBC to connect to any data store that supports JDBC. For example, use the Easysoft ODBC-JDBC Gateway to connect SQL Server to Data Lake.

- [ODBC API support](#)
- [XA support](#)

ODBC API support

Use this table to find out what ODBC API functions the Easysoft ODBC-JDBC Gateway supports:

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

| Function | Status |
|---------------------|---------------|
| SQLGetConnectOption | Supported |
| SQLGetCursorName | Supported |
| SQLGetData | Supported |
| SQLGetDescField | Supported |
| SQLGetDescRec | Supported |
| SQLGetDiagField | Supported |
| SQLGetDiagRec | Supported |
| SQLGetEnvAttr | Supported |
| SQLGetFunctions | Supported |
| SQLGetInfo | Supported |
| SQLGetStmtAttr | Supported |
| SQLGetStmtOption | Supported |
| SQLGetTypeInfo | Supported |
| SQLMoreResults | Supported |
| SQLNativeSql | Supported |
| SQLNumParams | Not supported |
| SQLNumResultCols | Supported |
| SQLParamData | Supported |
| SQLParamOptions | Supported |
| SQLPrepare | Supported |
| SQLPrimaryKeys | Supported |
| SQLProcedureColumns | Supported |
| SQLProcedures | Supported |
| SQLPutData | Supported |
| SQLRowCount | Supported |
| SQLSetConnectAttr | Supported |
| SQLSetConnectOption | Supported |
| SQLSetCursorName | Supported |
| SQLSetDescField | Supported |
| SQLSetDescRec | Supported |
| SQLSetEnvAttr | Supported |
| SQLSetParam | Supported |
| SQLSetPos | Not supported |
| SQLSetScrollOptions | Not supported |
| SQLSetStmtOption | Supported |

| Function | Status |
|--------------------|-----------|
| SQLSetStmtAttr | Supported |
| SQLStatistics | Supported |
| SQLTablePrivileges | Supported |
| SQLTables | Supported |
| SQLTransact | Supported |

The parameter based calls are not supported because support for them is not provided in JDBC 2.

Unicode ODBC calls

If available, the Easysoft ODBC-JDBC Gateway supports the Unicode version (with suffix "W", for example, SQLColAttributeW) of the ODBC calls it implements.

XA support

The Easysoft ODBC-JDBC Gateway can provide access to XA resources in the context of a distributed transaction.

To access an XA resource, add a Easysoft ODBC-JDBC Gateway data source that connects to the JDBC driver for the resource by using the XA interface.

Note To be involved in a distributed transaction, the target JDBC driver must implement the XA features of the JDBC 2.0 Optional Package.

In the Easysoft ODBC-JDBC Gateway data source, set the XA_Enlist data source option to 1. Use XAClass to specify the JDBC driver's XADataSource class. The class that the JDBC driver implements XADataSource with is used to produce XA connections. Use XIDClass to specify the method the JDBC driver provides to create Xids. The Transaction Manager uses Xids to identify each transaction in a distributed transaction. The XA_Connection_String attribute is required if the DB field is present in the xa_open string. The XA_Connection_String attribute value must be the same as that of the DB field.

This example data source sets up an XA connection to the Oracle JDBC driver:

```
[XA-OJG]
Driver = Easysoft ODBC-JDBC Gateway
DriverClass = oracle.jdbc.OracleDriver
User = system
Password = manager
ClassPath = /usr/local/oracle/jdbc/lib/ojdbc14.jar
URL = jdbc:oracle:oci8:@//my_host:1521/my_servicename
XA_Enlist = 1
XA_Connection_String = my_database
XAClass = oracle.jdbc.xa.client.OracleXADataSource
XIDClass = oracle.jdbc.xa.OracleXid
```

Note that if the XA attributes are present, the data source can only be used to access the target database as an XA resource under the control of a Transaction Manager. If you need to access the same database with a non-XA connection, configure a separate data source that does not contain the XA attributes.

The xa_open string

The Transaction Manager uses the xa_open string to connect to the Resource Manager.

The format for the Easysoft ODBC-JDBC Gateway xa_open string is:

```
ODBCJDBC_XA+DSN=data_source+LogFile=log_file+DB=db_name
```

ODBCJDBC_XA is the Easysoft ODBC-JDBC Gateway Resource Manager name. Use the DSN field to specify the Easysoft ODBC-JDBC Gateway data source that you configured for the resource. To enable logging, include the optional LogFile field. You need to use this field rather than the standard tracing mechanisms to enable logging for an XA connection. The DB field allows the ODBC connection to find the matching XA connection.

This WebSphere MQ example shows the corresponding xa_open string to use for the XA-OJG data source:

```
XAOpenString=ODBCJDBC_XA+DSN=XA-OJG+LogFile=/tmp/sql.log+DB=my_database
```

The xa_switch_t structure

The Easysoft ODBC-JDBC Gateway xa_switch_t structure name is ojc_xaosw. This structure contains the entry points for the Easysoft ODBC-JDBC Gateway Resource Manager. Applications requiring XA support should be built to use this structure. The Easysoft ODBC-JDBC Gateway translates calls into the entry points for the corresponding Java javax.sql.XADataSource methods.

Index

A

Access

- importing ODBC-JDBC Gateway data, [34](#)
- linking ODBC-JDBC Gateway data, [34](#)

Async Cancel attribute, [26](#)

B

Base

- working with ODBC-JDBC Gateway data, [44](#)

Bigint Default attribute, [26](#)

C

CanDescribeParam attribute, [22](#)

Class Path attribute, [25](#)

connecting to your Java data store, [19](#)

connection string attributes, [28](#)

D

data source attributes, [21](#)

Database attribute, [23](#)

Description attribute, [23](#)

DG4ODBC

- working with ODBC-JDBC Gateway data, [40](#)

Disable MoreResults attribute, [27](#)

Driver Class attribute, [24](#)

DSN attribute, [23](#)

DSN-less connections, [28](#)

E

Easysoft ODBC-JDBC Gateway

- adding ODBC data sources

- on Linux or UNIX, [19](#)

- on Windows, [20](#)

- connecting your Java datastore, [19](#)

- data source attributes, [21](#)

- DSN-less connections, [28](#)

- installing

- on Linux or UNIX, [8](#)

- on Windows, [6](#)

- JVMs and, [17](#)

- logging, [29](#)

- ODBC API support, [59](#)

- overview, [5](#)

- uninstalling

- on Linux or UNIX, [16](#)

- on Windows, [8](#)

- XA support, [62](#)

Excel

- importing ODBC-JDBC Gateway data with Data Connection Wizard, [35](#)

- importing ODBC-JDBC Gateway data with PowerPivot, [35](#)

- importing ODBC-JDBC Gateway data with

Query, [35](#)

G

Go

working with ODBC-JDBC Gateway data, [45](#)

J

JVM, [17](#)

L

LibreOffice

working with ODBC-JDBC Gateway data, [44](#)

linked server

working with ODBC-JDBC Gateway data, [38](#)

log files, [29](#)

M

Modify Metadata attribute, [25](#)

N

Node.js

working with ODBC-JDBC Gateway data, [47](#)

O

ODBC API function support, [59](#)

ODBC connection string attributes, [28](#)

ODBC data sources

adding

on Linux or UNIX, [19](#)

on Windows, [20](#)

Oracle

working with ODBC-JDBC Gateway data, [40](#)

P

Password attribute, [24](#)

Perl

working with ODBC-JDBC Gateway data, [49](#)

PHP

working with ODBC-JDBC Gateway data, [52](#)

Power BI

importing ODBC-JDBC Gateway data, [36](#)

PowerPivot

importing ODBC-JDBC Gateway data, [35](#)

Python

working with ODBC-JDBC Gateway data, [54](#)

R

R

working with ODBC-JDBC Gateway data, [56](#)

Reuse CL Object attribute, [26](#)

S

Single Statement attribute, [25](#)

SQL Server

working with ODBC-JDBC Gateway data, [38](#)

Strip Escape attribute, [26](#)

Strip Quote attribute, [25](#)

T

Tableau

working with ODBC-JDBC Gateway data, [37](#)

trace files, [29](#)

U

uninstalling

on Linux or UNIX, [16](#)

on Windows, [8](#)

URL attribute, [25](#)

User Name attribute, [24](#)

W

WCHAR Default attribute, [27](#)

X

XA

overview, [62](#)

xa_open string, [62](#)

xa_switch_t string, [63](#)

XA_Connection_String attribute, [22](#)

XA_Enlist attribute, [22](#)

XAClass attribute, [23](#)

XIDClass attribute, [23](#)