

Easysoft ODBC- MongoDB Driver User's Guide

This manual documents version 1.0.n of the Easysoft ODBC-MongoDB Driver.

Copyright © 1993-2025 Easysoft Limited.

All rights reserved.

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

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

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

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

Table of contents

| | |
|--------------------------------------------------------------|----|
| Getting started | 5 |
| Installing the Easysoft ODBC-MongoDB Driver | 6 |
| Installing on Linux or UNIX | 6 |
| Uninstalling on Linux or UNIX | 14 |
| Installing on Windows | 15 |
| Uninstalling on Windows | 17 |
| Connecting to MongoDB | 18 |
| Connecting from Linux or UNIX | 18 |
| Connecting from Windows | 19 |
| Connection attributes | 20 |
| DSN-less connections | 25 |
| Creating a schema for your MongoDB data | 25 |
| Logging | 31 |
| ODBC Driver Manager logging on Linux or UNIX | 31 |
| Easysoft ODBC-MongoDB Driver logging on Linux and UNIX | 31 |
| ODBC Driver Manager logging on Windows | 31 |
| Easysoft ODBC-MongoDB Driver logging on Windows | 33 |
| Finding out what product version you have on Windows | 34 |
| Client applications | 35 |
| Microsoft Access | 36 |
| Linking a table | 36 |
| Importing a table | 36 |
| Microsoft Excel | 37 |
| Data Connection Wizard | 37 |
| Microsoft Query | 37 |
| PowerPivot | 37 |
| Microsoft Power BI | 38 |
| Microsoft SQL Server | 39 |
| Oracle | 41 |
| Connecting MongoDB to Oracle on Windows | 41 |
| Connecting MongoDB to Oracle on Linux and UNIX | 42 |
| LibreOffice | 45 |
| Go | 46 |
| Node.js | 47 |
| Perl | 49 |
| PHP | 52 |
| Python | 54 |
| R | 56 |
| About the Easysoft ODBC-MongoDB Driver | 58 |
| ODBC API and scalar functions | 59 |
| API functions | 59 |
| Scalar functions | 61 |
| Data type mapping | 64 |
| Finding out more about data types on Windows | 65 |
| Features | 66 |
| SQL engine | 66 |
| Authentication | 66 |
| Catalog support | 66 |
| Embedded documents | 66 |
| Arrays | 67 |

| | |
|---------------------------------------------|----|
| SQL support..... | 68 |
| Example queries..... | 68 |
| Example inserts, updates, and deletes | 74 |
| Index | 77 |

Getting started

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

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

Installing the Easysoft ODBC-MongoDB Driver

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

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

Installing on Linux or UNIX

The installation can be done by anyone with root access.

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

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

2. Copy the distribution to a temporary directory on the machine where the application you want to connect to MongoDB 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-MongoDB Driver you need:

- The Bourne shell in /bin/sh. If your Bourne shell is not located there, you may need to edit the first line of the installation script.
- Various commonly used commands such as:

```
grep, awk, test, cut, ps, sed, cat, wc, uname, tr, find, echo, sum, head, tee, id
```

If you do not have any of these commands, they can usually be obtained from the Free Software Foundation. As the tee command does not work correctly on some systems, the distribution includes a tee replacement.

- Depending on the platform, you'll need up to 10 MB of temporary space for the installation files and up to 10 MB of free disk space for the installed programs. If you also install the unixODBC Driver Manager, these numbers increase by approximately 1.5 MB.
- For Easysoft licensing to work, you must do one of the following:
 - Install the Easysoft ODBC-MongoDB Driver in /usr/local/easysoft.
 - Install the Easysoft ODBC-MongoDB Driver elsewhere and symbolically link /usr/local/easysoft to wherever you chose to install the software.

The installation will do this automatically for you so long as you run the installation as someone with permission to create /usr/local/easysoft.

 - Install the Easysoft ODBC-MongoDB Driver elsewhere and set the EASYSOFT_ROOT environment variable. For more information about setting the EASYSOFT_ROOT environment variable, refer to [Post installation steps for non-root installations](#).
- An ODBC Driver Manager.

Easysoft ODBC-MongoDB Driver distributions include the unixODBC Driver Manager.

- You do not have to be the root user to install, but you will need permission to create a directory in the chosen installation path. Also, if you are not the root user, it may not be possible for the installation to:
 1. Register the Easysoft ODBC-MongoDB Driver with unixODBC.
 2. Create the example data source in the SYSTEM odbc.ini file.
 3. Update the dynamic linker entries (some platforms only).

If you are not root, these tasks will have to be done manually later.

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

What you can install

This distribution contains:

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

You need an ODBC Driver Manager to use the Easysoft ODBC-MongoDB Driver from your applications. The distribution therefore contains the unixODBC Driver Manager. Most (if not all) UNIX and Linux applications support the unixODBC Driver Manager. For example, Perl DBD::ODBC, PHP, Python, and so on.

You do not have to install the unixODBC Driver Manager included with this distribution. You can use an existing copy of unixODBC. For example, a version of unixODBC installed by another Easysoft product, a version obtained from your operating system vendor, or one that you built yourself. However, as Easysoft ensure that the unixODBC distributed with the Easysoft ODBC-MongoDB Driver has been tested with that driver, we recommend you use it.

If you choose to use an existing unixODBC Driver Manager, the installation script will attempt to locate it. The installation script looks for the ODBC Driver Manager in the standard places. If you have installed it in a non-standard location, the installation script prompts you for the location. The installation primarily needs unixODBC's odbcinst command to install drivers and data sources.

Where to install

This installation needs a location for the installed files. The default location is /usr/local.

At the start of the installation, you're prompted for an installation path. All files are installed in a subdirectory of your specified path called easysoft. For example, if you accept the default location /usr/local, the product will be installed in /usr/local/easysoft and below.

If you choose a different installation path, the installation script tries to symbolically link `/usr/local/easysoft` to the easysoft subdirectory in your chosen location. This allows us to distribute binaries with built in dynamic linker run paths. If you are not root or the path `/usr/local/easysoft` already exists and is not a symbolic link, the installation will be unable to create the symbolic link. For information about how to correct this manually, refer to [Post installation steps for non-root installations](#).

Note that you cannot license Easysoft products until either of the following is true:

- `/usr/local/easysoft` exists either as a symbolic link to your chosen installation path or as the installation path itself.
- You have set `EASYSOFT_ROOT` to *installation_path/easysoft*.

Changes made to your system

The installation script installs files in subdirectories of the path requested at the start of the installation. Depending on what is installed, a few changes may be made to your system:

1. If you choose to install the Easysoft ODBC-MongoDB Driver into unixODBC, unixODBC's `odbcinst` command will be run to add an entry to your `odbcinst.ini` file. You can locate this file with `odbcinst -j`. (`odbcinst` is in *installation_path/easysoft/unixODBC/bin*, if you are using the unixODBC included with this distribution.)
2. The installation script installs an example data source into unixODBC. This data source will be added to your `SYSTEM` `odbc.ini` file. You can locate your `SYSTEM` `odbc.ini` file by using `odbcinst -j`.
3. Dynamic linker. On operating systems where the dynamic linker has a file listing locations for shared objects (Linux and FreeBSD), the installation script will attempt to add paths under the path you provided at the start of the installation to the end of this list:
 - On Linux, this is usually the file `/etc/ld.so.conf`.
 - On FreeBSD this is usually the file `/etc/defaults/rc.conf`.

Installing alongside other existing Easysoft product installations

Each Easysoft distribution contains common files shared between Easysoft products. These shared objects are placed in *installation_path/easysoft/lib*. When you run the installation script, the dates and versions of these files are compared with the same files in the distribution. The files are only updated if the files being installed are newer or have a later version number.

You should ensure that nothing on your system is using Easysoft software before starting an installation. This is because on some platforms, files in use cannot be replaced. If a file cannot be updated, you get a warning during the installation. All warnings are written to a file called `warnings` in the directory you unpacked the distribution into.

If the installer detects you're upgrading a product, the installer will suggest you delete the product directory to avoid having problems with files in use. An alternative is to rename the specified directory.

If you are upgrading, you will need a new license from Easysoft to use the new driver.

Gathering information required during the installation

During the installation, you're prompted for various pieces of information. Before installing, you need to find out whether you have unixODBC already installed and where it is installed. The installation script searches standard places like `/usr` and `/usr/local`.

However, if you installed the Driver Manager in a non-standard place and you do not install the included unixODBC, you will need to know the location.

Unpacking the distribution

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

```
tar -xvf odbc-mongodb-1.0.19-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-mongodb-1.0.19-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-MongoDB Driver and data sources during the installation.
- Most applications and interfaces that support ODBC are compatible with unixODBC. The Easysoft ODBC-MongoDB Driver and any data sources that you add during the installation are automatically available to your applications and interfaces therefore.
- The unixODBC project is currently led by Easysoft developer Nick Gorham. This means that there is a great deal of experience at Easysoft of unixODBC in general and of supporting the Easysoft ODBC-MongoDB Driver running under unixODBC. It also means that if you find a problem in unixODBC, it's much easier for us to facilitate a fix.

The installation starts by searching for unixODBC. There are two possible outcomes here:

1. If the installation script finds unixODBC, the following message displays:

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

2. If the installation script can't find unixODBC in the standard places, you will be asked whether you have it installed.

If unixODBC is installed, you need to provide the unixODBC installation path. Usually, the path required is the directory above where `odbcinst` is installed. For example, if `odbcinst` is in `/opt/unixODBC/bin/odbcinst`, the required path is `/opt/unixODBC`.

If unixODBC is not installed, you should install the unixODBC included with this distribution.

If you already have unixODBC installed, you do not have to install the unixODBC included with the distribution, but you might consider doing so if your version is older than the one we provide.

The unixODBC in the Easysoft ODBC-MongoDB Driver 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-MongoDB Driver installation script:

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

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

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

Licensing

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

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

Important After obtaining a license, you should make a backup copy of this file.

The installation script asks you if you want to request an Easysoft ODBC-MongoDB Driver license:

```
Would you like to request a Easysoft ODBC-MongoDB Driver license now (y/n) [y]:
```

You do not need to obtain a license during the installation, you can run `licshell` after the installation to obtain or view licenses.

If you answer `y`, the installation runs the `licshell` script.

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

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

```
[0] exit
[1] view existing license
[n] obtain a license for the desired product.
```

To obtain a license, select one of the options from [2] onwards for the product you're installing. The License Client then runs a program that generates a key that's used to identify the product and operating system (we need this key to license you).

After you have chosen the product to license (Easysoft ODBC-MongoDB Driver), you need to supply:

- Your full name.
- Your company name.
- An email contact address. This must be the email address that you used when you registered on the Easysoft web site.
- A reference number (also referred to as an authorization code). When applying for a trial license, press Enter when prompted for a reference number. This field only applies to full (paid) licenses.

You're then asked to choose how you want to obtain the license.

The choices are:

- [1] Automatically by contacting the Easysoft License Daemon
This requires a connection to the Internet and the ability to support an outgoing TCP/IP connection to `license.easysoft.com` on port 8884.
- [2] Write information to file
The license request is output to `license_request.txt`.
- [3] Cancel this operation

If you choose to obtain the license automatically, the License Client tries to open a TCP/IP connection to `license.easysoft.com` on port 8884 and send the details you supplied along with your machine number. No other data is sent. The data sent is transmitted as plain text, so if you want to avoid the possibility of this information being intercepted by someone else on the Internet, you should choose [2] and send the the request to us. The License daemon returns the license key, prints it to the screen and make it available to the installation script in the file `licenses.out`.

If you choose option [2], the license request is written to the file `license_request.txt`. You should then exit the License Client by choosing option [0] and complete the installation. After you have sent the license request to us, we'll return a license key. Add this to the end of the file `installation_path/easysoft/license/licenses`.

Post installation steps for non-root installations

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

1. If you attempt to install the Easysoft ODBC-MongoDB Driver under the unixODBC Driver Manager and you do not have write permission to unixODBC's `odbcinst.ini` file, the driver can't be added.

You can manually install the driver under unixODBC by adding an entry to the `odbcinst.ini` file. Run `odbcinst -j` to find out the location of the `DRIVERS` file then append the lines from `drv_template` file to `odbcinst.ini`. (`drv_template` is in the directory where the Easysoft distribution was untarred to.)

2. No example data sources can be added into unixODBC if you do not have write permission to the `SYSTEM odbc.ini` file. Run `odbcinst -j` to find out the location of the `SYSTEM DATA SOURCES` file then add your data sources to this file.
3. On systems where the dynamic linker has a configuration file defining the locations where it looks for shared objects (Linux and FreeBSD), you need to add:

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

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

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

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

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

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

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

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

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

Uninstalling on Linux or UNIX

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

1. Change directory to *installation_path*/easysoft and delete the product directory. *installation_path* is the Easysoft ODBC-MongoDB Driver installation directory, by default /usr/local.
2. If you had to add this path to the dynamic linker search paths (for example, /etc/ld.so.conf on Linux), remove it. You may have to run a linker command such as /sbin/ldconfig to get the dynamic linker to reread its configuration file. Usually, this step can only be done by the root user.
3. If you were using unixODBC, the Easysoft ODBC-MongoDB Driver entry needs to be removed from the odbcinst.ini file. To check whether the Easysoft ODBC-MongoDB Driver is configured under unixODBC, use odbcinst -q -d. If the command output contains [Easysoft ODBC-MongoDB], uninstall the driver from unixODBC by using:

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

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

Installing on Windows

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

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

Updating files that are in use

To avoid rebooting your computer, the Easysoft ODBC-MongoDB Driver installer prompts you when files that it needs to update are in use by another application or service. This frees the locked files and allows the installation to complete without a system restart. The installer uses the **Restart Manager** to locate the applications that are using files that need updating. These applications are displayed in the **Files in Use** dialog box. To avoid a system restart, choose **Automatically close applications and attempt to restart them after setup is complete**. The Easysoft ODBC-MongoDB Driver installer then uses **Restart Manager** to try to stop and restart each application or service in the list. If possible, **Restart Manager** restores applications to the same state that they were in before it shut them down.

Licensing

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

- A free time-limited trial license, which gives you free and unrestricted use of the product for a limited period (usually 14 days).
- A full license if you have purchased the product. On purchasing the product you are given an authorization code, which you use to obtain a license.

To license the Easysoft ODBC-MongoDB Driver:

1. In License Manager, enter your contact details.

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

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

2. Choose **Request License**.

You're prompted to choose a license type.

3. Do one of the following:

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

-Or-

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

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

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

6. Choose how to get your license when prompted.

7. Do one of the following:

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

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

-Or-

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

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

-Or-

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

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

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

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

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

Repairing the installation

The installer can repair a broken Easysoft ODBC-MongoDB Driver installation. For example, you can use the installer to restore missing Easysoft ODBC-MongoDB Driver files or registry keys. To do this:

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

Uninstalling on Windows

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

Removing Easysoft ODBC-MongoDB Driver data sources

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

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

```
odbcad32.exe
```

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

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

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

Repeat the previous two steps.

Removing the Easysoft ODBC-MongoDB Driver

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

Note

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

Connecting to MongoDB

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

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

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

Connecting from Linux or UNIX

Creating an ODBC data source

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

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

By default, the Easysoft ODBC-MongoDB Driver installation creates a sample SYSTEM data source named MONGODB_SAMPLE. If you're using the unixODBC included in the Easysoft ODBC-MongoDB Driver distribution, the SYSTEM `odbc.ini` file is in `/etc`.

If you built unixODBC yourself, or installed it from some other source, SYSTEM data sources are stored in the path specified with the configure option `--sysconfdir=directory`. If `sysconfdir` was not specified when unixODBC was configured and built, it defaults to `/usr/local/etc`.

If you accepted the default choices when installing the MongoDB, 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-MongoDB Driver is installed into unixODBC, it places a Easysoft ODBC-MongoDB entry into the `odbcinst.ini` file. You should always have `Driver = Easysoft ODBC-MongoDB` in your Easysoft ODBC-MongoDB Driver data sources therefore.

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

- The hostname or IP address of the MongoDB server (Server).
- The database to connect to (Database).

For example:

```
[MONGODB_SAMPLE]
Driver          = Easysoft ODBC-MongoDB
Server          = localhost
Database        = Atlas
```

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

- libodbcinst.so

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

- libeslicshr.so

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

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

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

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

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

```
SQL> SELECT * FROM Restaurants;
```

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

Connection attributes

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

Setting on Linux and UNIX

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

- Server
- Database

For example:

```
[MONGODB_SAMPLE]
Driver           = Easysoft ODBC-MongoDB
Server           = localhost
Database         = Atlas
```

These optional attributes may be set in odbc.ini.

- Auth_Source
- Ca_Dir
- Ca_File
- Crl_File
- DefaultW
- Description
- IgnoreAuth
- Logfile
- Logging

- MD_Database
- MD_Password
- MD_SRV_DNS
- MD_Server
- MD_User
- Password
- Pem_File
- Pem_Pwd
- Preview
- SASL
- SRV_DNS
- SSL
- User
- Weak_Cert_Validation

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

| Name | Value |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CharLen | The string column length to use. MongoDB does not provide the length for string columns in its column metadata. By default, the Easysoft ODBC-MongoDB Driver uses a string length of 255, which you can override with this setting. |
| insert_nulls | <p>Whether NULLS are inserted. When insert_nulls is turned off (the default), this example INSERT statement:</p> <pre>INSERT INTO TABLE t1 (c1, c2) VALUES('value', NULL)</pre> <p>inserts:</p> <pre>{ c1 : 'value' }</pre> <p>When insert_nulls is turned on, this example INSERT statement:</p> <pre>INSERT INTO TABLE t1 (c1, c2) VALUES('value', NULL)</pre> <p>inserts:</p> <pre>{ c1 : 'value', c1 : null }</pre> |

Setting on Windows

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

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

22 Connection attributes

| Name | Value |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Server | The host name or IP address of the machine on which the MongoDB server is running. |
| Database | The name of the MongoDB database to connect to. |
| User Name | <p>The MongoDB user name, if required.</p> <p>To retrieve MongoDB data, the user needs to have the read database role. To update MongoDB data, the user needs to have the readWrite database role.</p> |
| Password | The password for User Name. |
| Schema Server | <p>If you store a custom schema generated with the Easysoft MongoDB Schema Editor in a MongoDB server, supply the hostname or IP address of this server with the Schema Server attribute. If you do not want to use a custom schema or want the schema to be stored on the same host as Server, omit this attribute.</p> <p>By default, the Easysoft ODBC-MongoDB Driver samples your MongoDB data to generate the schema it needs to describe the data to ODBC client applications. If you choose to customise the schema (for example, you do not want to expose all fields in a document to your client application), you have the option to store the schema in a MongoDB server</p> |
| Schema Database | If you store a custom schema generated with the Easysoft MongoDB Schema Editor in a MongoDB server, specify the database where the schema is located with the Schema Database attribute. Otherwise, omit this attribute |
| Schema User Name | <p>If you need to supply a user name to access the database where your custom schema is stored, supply it with the Schema User Name attribute. To view the schema, the user needs to have the read database role. To create a schema, the user needs to have:</p> <p>The readWrite and dbAdmin roles.</p> <p>-Or-</p> <p>The dbOwner role.</p> |
| Schema Password | If you need to supply a user name to access the database where your custom schema is stored, supply the password for this user name with the Schema User Name attribute. |
| Default to WCHAR | <p>When turned on, MongoDB types that the Easysoft ODBC-MongoDB Driver would normally map to SQL_VARCHAR types are mapped to SQL_WVARCHAR types instead. By default, Default to WCHAR is turned off. Turn on this setting if you are using a Unicode ODBC application that expects Unicode data types. For example, Perl DBD ODBC when built with Unicode support.</p> <p>You can also use the Easysoft Schema Editor to define how MongoDB data types are mapped to ODBC data types.</p> |

| Name | Value |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ignore Auth | Some applications (for example MicroStrategy, Oracle Heterogeneous Services (DG4ODBC), Oracle BI Administration Tool) always pass a user name and password on the ODBC connection string, regardless of whether they are required by the target database. If your application behaves in this way and your MongoDB database does not require a user name and a password, turn on this setting. |
| Logging | Whether to turn on Easysoft ODBC-MongoDB Driver logging. Normally, you'll only do this if so directed by the Easysoft support team. |
| Log File | <p>The file name and path of the file you want the driver to write log information to. For example:</p> <p>C:\Windows\Temp\Easysoft.log</p> <p>If the file doesn't exist, the Easysoft ODBC-MongoDB Driver creates it.</p> |
| Preview Rows | <p>When in schema discovery mode, the Easysoft ODBC-MongoDB Driver scans 100 documents to generate a schema for a collection. To change the number of documents that the driver scans, use the Preview Rows attribute. For example, if the 101st document contained a field whose value was a different type to that of the preceding 100 documents, you would set Preview Rows to 101. You may also need to increase the Preview Rows size, if a field contained a value whose length exceeded the maximum length of that field's value in the first 100 documents. (In this circumstance, you may get an error similar to "The (maximum) expected data length is x while the returned data length is y.")</p> <p>To scan all documents in a collection (which will have performance implications for large collections), set Preview to -1.</p> |
| SSL | Turn on SSL if you want to connect to MongoDB over an encrypted connection and the target MongoDB server has been started with --sslMode allowSSL or --sslMode preferSSL. If the target MongoDB server has been started with --sslMode requireSSL, you must turn on SSL, otherwise you will get the error "An error occurred: Failed to read 4 bytes from socket" when the driver attempts to do some work (for example, executes a query). |
| SRV_DNS | Turn on this setting if you are connecting to a cloud-based MongoDB instance. For example, MongoDB Atlas. |

| Name | Value |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Weak Validation | <p>Turn on Weak Validation to connect to MongoDB over an SSL connection without presenting a certificate. The MongoDB server needs to permit such a connection, that is, have been started with <code>--sslWeakCertificateValidation</code> or <code>--sslAllowConnectionsWithoutCertificates</code>. If you want to connect without presenting a certificate, you need to turn on SSL and Weak Validation only. None of the other SSL attributes (PEM File, CA File, and so on) are required.</p> <p>If you attempt to connect to a MongoDB server that permits an SSL connection without a certificate you will get the error "Failed to handshake and validate TLS certificate" unless you turn on Weak Validation.</p> |
| Schema SRV_DNS | If you have specified a Schema Database and are connecting to a cloud-based MongoDB instance, turn on this setting. |
| PEM File | The path to the PEM file for the MongoDB server certificate. For example, <code>C:\SSL\MongoDB.pem</code> . |
| PEM Password | The password for PEM File. |
| CA File | The file containing trusted certificates. For example, <code>C:\SSL\MongoDB.crt</code> . The file must include the Certificate Authority (CA) file for the MongoDB server certificate. |
| CA Directory | The directory containing trusted certificates. For example, <code>C:\SSL</code> . The directory must contain the CA file for the MongoDB server certificate. The CA file name must be in "hash format", for example <code>bd035bae.0</code> . If you specify the CA Directory attribute, omit the CA File attribute. |
| CRL File | To prevent a connection to a MongoDB server with a revoked certificate, use CRL File specify the file that contains revoked certificates. |
| SASL | Use SASL authentication. |
| Auth Source | The authentication database for User Name. |

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-MongoDB Driver connection string:

```
Driver={Easysoft MongoDB ODBC Driver};SERVER=localhost;DATABASE=blog;
```

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

Creating a schema for your MongoDB data

In this section:

- [Running the Easysoft MongoDB Schema Editor on Windows](#)
- [The Easysoft MongoDB Schema Editor interface](#)
- [Creating a schema for your MongoDB data](#)
- [Adding columns](#)
- [Hiding and deleting columns](#)
- [Virtual tables](#)

Running the Easysoft MongoDB Schema Editor on Windows

Do one of the following:

- In **ODBC Data Source Administrator**, open the Easysoft ODBC-MongoDB Driver data source that specifies the MongoDB database whose data you want to create a schema for. Use the **Configure** button to open the Easysoft ODBC-MongoDB Driver data source configuration dialog box. Choose the **Schema** button.

-Or-

- Run the Easysoft ODBC-MongoDB Driver executable, esmgse.exe.

On 64-bit Windows, esmgse.exe is located in the %programfiles(x86)%\Easysoft Limited\Easysoft MongoDB ODBC Driver\Schema Editor folder.

On 32-bit Windows, esmgse.exe is located in the %programfiles%\Easysoft Limited\Easysoft MongoDB ODBC Driver\Schema Editor folder.

esmgse.exe accepts these command line arguments:

| Option | Description |
|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-dsn data source name</code> | <p>The ODBC data source that specifies the MongoDB database whose data you want to create a schema for.</p> <p>At a minimum, the ODBC data source needs to specify the MongoDB database. If the data source contains other values that are relevant to the schema editor, for example, the MongoDB server where you want to store the schema, there is no need to specify these values on the command line.</p> <p>Any values you do specify on the command line override ones specified in the data source.</p> <p>The Easysoft MongoDB Schema Editor looks for a user data source with the name you specify and then a system data source. (On 64-bit Windows, the Easysoft MongoDB Schema Editor looks first for a 32-bit system data source and then a 64-bit system data source.)</p> <p>The Easysoft MongoDB Schema Editor uses the first data source it finds with the specified name.</p> |
| <code>[-ds database server]</code> | The MongoDB server that serves the MongoDB database whose data you want to create a schema for. |
| <code>[-db database name]</code> | The MongoDB database whose data you want to create a schema for. |
| <code>[-du database user name]</code> | If MongoDB authentication is enabled, the user name required to access database name. |
| <code>[-dp database password]</code> | The password for database password. |
| <code>[-ss schema server]</code> | The MongoDB server where you want to store the schema. If you omit this attribute on the Easysoft MongoDB Schema Editor command line and in the data source, the schema will be stored on the same machine as database server. |
| <code>[-sc schema name]</code> | The name of the schema. The Easysoft MongoDB Schema Editor creates a database with this name to store the schema collections in. |
| <code>[-su schema user name]</code> | If MongoDB authentication is turned on, the user name required to access schema name. |
| <code>[-sp schema password]</code> | The password for schema password. |
| <code>[-cw sample data max display width]</code> | The Easysoft MongoDB Schema Editor lets you to view your MongoDB data, Use <code>-cw</code> to limit the amount of text data that appears in this view. For example, if you set <code>-cw</code> to 1000, the maximum amount of data for each text column that displays is 1000 characters. Use <code>-cw</code> if you're experiencing performance problems with the Easysoft MongoDB Schema Editor. |
| <code>[-debug filename]</code> | Where to write schema information for a document. For example, <code>c:\windows\temp\schema.txt</code> . If you include the <code>-debug</code> option, a Dump Schema button becomes available in the Easysoft MongoDB Schema Editor. Choose this button to write the schema for the selected MongoDB document to <i>filename</i> . |

| Option | Description |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>[-ssl]</code> | Include this option if you want to connect to MongoDB over an encrypted connection and the target MongoDB server has been started with <code>--sslMode allowSSL</code> or <code>--sslMode preferSSL</code> . |
| <code>[-wcv]</code> | Include this option to connect to MongoDB over an SSL connection without presenting a certificate. The MongoDB server needs to permit such a connection, that is, have been started with <code>--sslWeakCertificateValidation</code> or <code>--sslAllowConnectionsWithoutCertificates</code> . If you want to connect without presenting a certificate, you need to enable <code>-ssl</code> and <code>-wcv</code> only. None of the other SSL attributes (<code>-cafile</code> , <code>-pemfile</code> , and so on) are required. |
| <code>[-cafile filename]</code> | Use this option to specify the file containing trusted certificates. For example, <code>C:\SSL\my.crt</code> . The file must include the Certificate Authority (CA) file for the MongoDB server certificate. |
| <code>[-cadir path]</code> | Use this option to specify the directory containing trusted certificates. For example, <code>C:\SSL</code> . The directory must contain the CA file for the MongoDB server certificate. The CA file name must be in "hash format", for example <code>bd035bae.0</code> . If you specify the <code>-cadir</code> option, omit the <code>-cafile</code> option. |
| <code>[-pemfile filename]</code> | Use this option to specify the path to the PEM file for the MongoDB server certificate. For example, <code>C:\SSL\my.pem</code> . |
| <code>[-pempwd password]</code> | Use this option to specify the password for PEM File. |

If you start the Easysoft MongoDB Schema Editor without specifying the MongoDB server where the schema is stored, the editor starts in View Only mode. In this mode, you can examine your MongoDB data and the default schema generated by the editor, but cannot make any changes.

The Easysoft MongoDB Schema Editor interface

The Easysoft MongoDB Schema Editor contains three panes. The left hand pane, **Database and Tables**, displays the collections in the MongoDB database that were specified in the data source or on the command line.

The bottom right hand pane, **Data Sample**, shows the documents that were used to generate the schema. By default, the **Data Sample** pane shows the first 100 documents.

The top right hand pane, **Current Table**, displays the schema for the chosen collection:

SQL Name

The column name that's returned by the Easysoft ODBC-MongoDB Driver to the application. This is the same as the corresponding field in the MongoDB document. You may need to change the column name if the MongoDB field on which it is based contains characters that the ODBC layer or application cannot handle. To change the column name, double-click the column name and then enter a new value.

SQL Type

The ODBC SQL data type that's returned by the Easysoft ODBC-MongoDB Driver to the application. To change the ODBC SQL data type, choose a different type from the list. For example, you might change a Boolean MongoDB field from a `SQL_TINYINT` to a `SQL_BIT`.

SQL Index

If the field is part of a MongoDB index, Yes displays in the **SQL Index** column. If you hover over the column with your mouse pointer, a popup window displays to show you all the indexes that field is included in. The popup window also show you the order in which the field is included in a compound index, if applicable.

MongoDB Name

The MongoDB field name.

MongoDB Type

The BSON type for the MongoDB field value.

Length

The string length, if a string data type is shown in the **SQL Type** column. The Easysoft ODBC-MongoDB Driver reports the length shown here to the application. To change the string length, double-click the length, and then enter the new value.

Precision

The precision, if a SQL_NUMERIC data type is shown in the **SQL Type** column. The Easysoft ODBC-MongoDB Driver reports the precision shown here to the application. To change the precision, double-click the precision and then enter the new value.

Scale

The scale, if a SQL_NUMERIC data type is shown in the **SQL Type** column. The Easysoft ODBC-MongoDB Driver reports the scale shown here to the application. To change the scale, double-click the scale and then enter the new value.

Hide

Whether the column's contents and metadata are visible to the ODBC application.

Creating a schema for your MongoDB data

1. In **ODBC Data Source Administrator**, select your Easysoft ODBC-MongoDB Driver data source and then choose **Configure**.
2. In the **Schema Server** box, enter the host name or IP address of the server where you want to store the schema.
3. In the **Schema Database** box, enter the name of the schema.

The Easysoft ODBC-MongoDB Driver creates a MongoDB database with the name you specify to store the schema collections in.

4. If MongoDB authentication is enabled on the server where the schema is stored, enter a user name for this server in the **Schema User Name** box.
5. Enter the password for this user name in the **Schema Password** box.
6. Choose the **Schema** button to start the Easysoft MongoDB Schema Editor.
7. Choose Yes when prompted whether to create a new Schema.
8. In the Easysoft MongoDB Schema Editor make the changes you want.
9. Choose **Save** and then choose **Exit**.

Adding columns

For the Easysoft ODBC-MongoDB Driver to be able to retrieve a MongoDB field value, that field needs to be described in a schema. The schema can be either the one that the Easysoft ODBC-MongoDB Driver generates automatically or one that you create in the Easysoft MongoDB Schema Editor. If a field is not described in a schema, the Easysoft ODBC-MongoDB Driver returns NULL rather than the field's value.

If you add a document to a collection that contains types that are different to those in the documents sampled by the driver or schema editor, you may need to manually add columns. For example, you generated a schema in the Easysoft MongoDB Schema Editor based on a sample size of 100 documents; document 500 contains fields with different types. Rather than regenerate the schema based on a sample size of 500, you choose to manually add columns for the new fields.

1. In the Easysoft MongoDB Schema Editor, in the **Tables** pane, select the row below which you want the new column to appear.
2. Choose **Insert**.
3. In **SQL Name**, double click the column name and enter a new value if you want a different column name to be returned to the application.
4. In **SQL Type**, select the ODBC SQL data type that will be returned to the application.
5. In **MongoDB Name**, double-click the column name and enter the name of the corresponding MongoDB field name.
6. In **MongoDB Type**, select the BSON type for the MongoDB field value.
7. If **SQL Type** is a string data type, in **Length**, double-click the length and enter the length of the field to be returned to the application.
8. If **SQL Type** is a SQL_NUMERIC data type, in **Precision**, double-click the precision and enter the precision of the field to be returned to the application.
9. If **SQL Type** is a SQL_NUMERIC data type, in **Scale**, double-click the scale and type the scale of the field to be returned to the application.
10. When you have finished adding columns, choose **Save**.

Hiding and deleting columns

If you do not want MongoDB field to be visible in your application, you can either hide or delete the corresponding column in the Easysoft ODBC-MongoDB Driver.

1. Locate the column in the Easysoft MongoDB Schema Editor and then choose **Hide**.
2. Choose **Save**.

To delete a column:

1. Select the column in the Easysoft MongoDB Schema Editor and then choose **Remove**.
2. Choose **Save**.

Virtual tables

Top-level arrays can be expanded into virtual tables. The Easysoft MongoDB Schema Editor displays a **Virtual Table** check box next to eligible fields. Select this check box to expand an array into virtual tables.

To illustrate the virtual tables feature, here is an example based on this MongoDB document:

```
db.inventory.insert(
{
  item: "ABC1",
```

30 Creating a schema for your MongoDB data

```
details: {  
  
  model: "14Q3",  
  
  manufacturer: "XYZ Company"  
  
},  
  
stock: [ { size: "S", qty: 25 }, { size: "M", qty: 50 } ],  
  
category: "clothing"  
  
}  
  
)
```

Stock is an array and so can be expanded into a virtual table. The resultant table is `inventory_stock`:

```
SELECT * FROM inventory_stock  
  
"_id", "stock$index", "stock$value"  
  
"54cf68badc5ac264551cb974", 0, "{ \"size\" : \"S\", \"qty\" : 25.000000 }"  
  
"54cf68badc5ac264551cb974", 1, "{ \"size\" : \"M\", \"qty\" : 50.000000 }"
```

Logging

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

To turn on logging, refer to the following sections.

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

ODBC Driver Manager logging on Linux or UNIX

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

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

For example:

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

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

Easysoft ODBC-MongoDB Driver logging on Linux and UNIX

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

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

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

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

ODBC Driver Manager logging on Windows

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

```
odbcad32.exe
```

-Or-

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

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

Note

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

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

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

6. Choose **Start Tracing Now**.

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

Easysoft ODBC-MongoDB Driver logging on Windows

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

```
odbcad32.exe
```

-Or-

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

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

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

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

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

Finding out what product version you have on Windows

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

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

The product version displays below.

Client applications

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

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

Microsoft Access

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

Linking a table

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

Importing a table

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

Microsoft Excel

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

Data Connection Wizard

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

Microsoft Query

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

Microsoft Power BI

1. [Install the Easysoft ODBC-MongoDB Driver](#) on same computer as Power BI Desktop.
2. [Configure an ODBC data source](#).
3. In Power BI Desktop, choose **Get data from another source**.
4. In the **Get Data** dialog box, choose **ODBC**, and then choose **Connect**.
5. In the **From ODBC** dialog box, choose your MongoDB 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 MongoDB data is now available to use in Power BI visualisations.

Microsoft SQL Server

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

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

4. Right-click **Server Objects**. From the pop-up menu choose **New > Linked Server**.
5. In the **Linked server** box, enter "MongoDB".
6. From the **Provider** list, choose **Microsoft OLE DB Provider for ODBC drivers**.
7. In the **Data source** box, enter the name of your MongoDB 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-MongoDB Driver, it will not have the latest version of the System Path environment variable. The Easysoft ODBC-MongoDB Driver Setup program adds entries for the driver to the System Path. Restarting the instance makes these changes available to SQL Server, allowing it to load the Easysoft ODBC-MongoDB Driver.
 - If you made a mistake when specifying the Easysoft ODBC-MongoDB Driver, you get the error "Data source name not found and no default driver specified." If you get this error, choose **No** when prompted whether to keep the linked server and edit the value in the **Data source** box.
8. You can query your Easysoft ODBC-MongoDB Driver data either by using a:
 - Four part table name in a distributed query.

A four part table name has the format:

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

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

```
EXEC sp_tables_ex @table_server = 'MongoDB'
```

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

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

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

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

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

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

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

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

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

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

Connecting MongoDB 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 initMongodb.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 = "MongoDB"
```

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

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

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

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

```
SID_LIST_LISTENER =
( SID_LIST =
( SID_DESC=
( SID_NAME=Mongodb)
( 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:

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

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

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

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

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

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

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

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

Notes

- If you have problems connecting to MongoDB 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 initMongoDB.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 MongoDB 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 initMongodb.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 = "MongoDB"
```

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

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

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

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

```
SID_LIST_LISTENER =
( SID_LIST =
( SID_DESC=
( SID_NAME=Mongodb)
( 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:

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

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

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

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

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

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

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

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

Notes

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

LibreOffice

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

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

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

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

Go

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

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

4. Create and then use Go to run this script, which retrieves some MongoDB 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=MongoDB")
    if err != nil {
        log.Fatal(err)
    }

    var (
        name string
    )

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

    defer db.Close()
}
```

Node.js

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

```
npm install odbc
```

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

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

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

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

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

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

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

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

Perl

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

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

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

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

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

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

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

my $sql = "SELECT Surname FROM Customers";

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

$sth->execute();

my($Col);

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

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

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

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

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:MongoDB');

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 MongoDB data:

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

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

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

$dbh->disconnect if ($dbh);

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

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

$sth->execute('PowerGroup');

$dbh->disconnect if ($dbh);

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

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

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

Further information

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

PHP

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

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

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

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

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

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

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

Further information

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

Python

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

```
pip list
```

If you don't have pip installed:

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

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

```
pip install pyodbc
```

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

```
import pyodbc

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

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

```
import pyodbc

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

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

```
import pyodbc

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

```
import pyodbc

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

```
import pyodbc

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

Further information

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

R

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

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

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

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

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

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

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

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

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

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

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

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

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

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

About the Easysoft ODBC-MongoDB Driver

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

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

ODBC API and scalar functions

API functions

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

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

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

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

Scalar functions

The Easysoft ODBC-MongoDB Driver supports a number of scalar functions:

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

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

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

String functions

The Easysoft ODBC-MongoDB Driver supports these [string](#) functions:

- `ASCII(string_exp)`
- `BIT_LENGTH(string_exp)`
- `CHAR(code)`
- `CHAR_LENGTH(string_exp)`
- `CHARACTER_LENGTH(string_exp)`
- `CONCAT(string_exp1, string_exp2)`
- `INSERT(string_exp1, start, length, string_exp2)`
- `LCASE(string_exp)`
- `LEFT(string_exp, count)`
- `LENGTH(string_exp)`
- `LOCATE(string_exp1, string_exp2[,start])`
- `LTRIM(string_exp)`
- `OCTET_LENGTH(string_exp)`
- `POSITION(char_exp IN char_exp)`
- `REPEAT(string_exp, count)`
- `REPLACE(string_exp1, string_exp2, string_exp3)`
- `RIGHT(string_exp, count)`
- `RTRIM(string_exp)`
- `SOUNDEX(string_exp)`
- `SPACE(count)`

- UCASE(*string_exp*)

Numeric functions

The Easysoft ODBC-MongoDB Driver supports these [numeric](#) functions:

- ABS(*numeric_exp*)
- ACOS(*float_exp*)
- ASIN(*float_exp*)
- ATAN(*float_exp*)
- CEILING(*numeric_exp*)
- COS(*float_exp*)
- COT(*float_exp*)
- DEGREES(*numeric_exp*)
- EXP(*float_exp*)
- FLOOR(*numeric_exp*)
- LOG(*float_exp*)
- LOG10(*float_exp*)
- MOD(*integer_exp1*, *integer_exp2*)
- PI()
- POWER(*numeric_exp*, *integer_exp*)
- RADIANS(*numeric_exp*)
- RAND([*integer_exp*])
- ROUND(*numeric_exp*, *integer_exp*)
- SIGN(*numeric_exp*)
- SIN(*float_exp*)
- SQRT(*float_exp*)
- TAN(*float_exp*)
- TRUNCATE(*numeric_exp*, *integer_exp*)

Time, date, and interval functions

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

- CURRENT_DATE()
- CURRENT_TIME([*time-precision*])
- CURRENT_TIMESTAMP([*timestamp-precision*])
- DAYNAME(*date_exp*)
- DAYOFMONTH(*date_exp*)
- DAYOFWEEK(*date_exp*)
- DAYOFYEAR(*date_exp*)
- EXTRACT(*extract-field* FROM *extract-source*)
- HOUR(*time_exp*)
- MINUTE(*time_exp*)
- MONTH(*date_exp*)
- MONTHNAME(*date_exp*)
- QUARTER(*date_exp*)
- SECOND(*time_exp*)
- TIMESTAMPDIFF(*interval*, *timestamp_exp1*, *timestamp_exp2*)
- WEEK(*date_exp*)
- YEAR(*date_exp*)

System functions

The Easysoft ODBC-MongoDB Driver supports these [system](#) functions:

- DATABASE()
- USER()

Conversion functions

The Easysoft ODBC-MongoDB Driver supports the [SQL-92 CAST](#) function for conversion between compatible data types.

Data type mapping

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

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

| MongoDB data type | ODBC data type |
|---------------------------|-------------------------------|
| INTERVAL HOUR TO SECOND | SQL_INTERVAL_HOUR_TO_SECOND |
| INTERVAL MINUTE TO SECOND | SQL_INTERVAL_MINUTE_TO_SECOND |

Further reading

- [Creating a schema for your MongoDB data](#)

Finding out more about data types on Windows

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

1. Download the version of ODBC Test that matches your application's architecture from:

<https://www.easysoft.com/ftp/pub/utils/windows/odbc-test/>

2. Copy both files to a folder on the machine where Easysoft ODBC-MongoDB Driver is installed.
3. Double-click **odbcte32.exe**.
4. Select **Con > Full Connect**.
5. Choose your Easysoft ODBC-MongoDB Driver data source from the list.
6. Choose **Catalog > SQLGetTypeInfo**.
7. Either choose **SQL_ALL_TYPES=0 (1.0)** or a specific data type from the **DataType** list.
8. Choose **Results > Get Data All**.

Features

In this section:

- [SQL engine](#)
- [Authentication](#)
- [Catalog support](#)
- [Embedded documents](#)
- [Arrays](#)

SQL engine

To enable client applications to use SQL statements against MongoDB, the Easysoft ODBC-MongoDB Driver uses the SQL Engine feature to translate standard SQL-92 queries into equivalent MongoDB API calls.

Authentication

The Easysoft ODBC-MongoDB Driver enforces MongoDB's built in authentication. If your MongoDB instance uses authentication, you need to supply the user name and password for a user who has the necessary privileges, when configuring your data source. To retrieve MongoDB data, the user needs to have the read database role. To update MongoDB data, the user needs to have the readWrite database role.

Catalog support

The Easysoft ODBC-MongoDB Driver uses the name of the MongoDB database as the catalog to make it easy for the driver to work with the various ODBC applications that expect a catalog name.

Embedded documents

The Easysoft ODBC-MongoDB Driver flattens an embedded document into a relational structure. The driver also returns the original (in other words, non flattened) embedded document in a separate column, which you have the option of hiding in the Schema Editor Tool.

To illustrate how the Easysoft ODBC-MongoDB Driver handles embedded documents, here is an example. Given this document to retrieve:

```
{
  _id: "joe",
  name: "Joe Bookreader",
  address: {
    street: "123 Fake Street",
    city: "Faketon",
    state: "MA",
    zip: "12345"
  }
}
```

The Easysoft ODBC-MongoDB Driver returns:

```
"_id", "name", "address", "address_street", "address_city", "address_state",
"address_zip"
"joe", "Joe Bookreader", "{ \"street\" : \"123 Fake Street\", \"city\" : \"Faketon\",
```

```
"state" : "MA", "zip" : "12345" }", "123 Fake Street", "Faketon", "MA", "12345"
```

The embedded document containing the address is returned as the address column value. The fields in the embedded document are flattened into the columns address_street, address_city, address_state, and address_zip.

Arrays

The Easysoft ODBC-MongoDB Driver flattens an array into a relational structure. The column names include the index of the item in the array that the column represents, starting with an index of 0. The driver also returns the original (in other words, non flattened) array in a separate column, which you have the option of hiding in the Schema Editor Tool.

To illustrate how the Easysoft ODBC-MongoDB Driver handles arrays, here is an example. Given this document to retrieve:

```
db.inventory.insert(
{
  item: "ABC1",
  details: {
    model: "14Q3",
    manufacturer: "XYZ Company"
  },
  stock: [{
    size: "S",
    qty: 25
  }, {
    size: "M",
    qty: 50
  }],
  category: "clothing"
})
```

The Easysoft ODBC-MongoDB Driver returns:

```
"_id", "item", "details", "details_model", "details_manufacturer", "stock",
"stock_0_size", "stock_0_qty", "stock_1_size", "stock_1_qty", "category"
"542ac8247bbc8c71916b3eb9", "ABC1", "{ \"model\" : \"14Q3\", \"manufacturer\" : \"XYZ
Company\" }", "14Q3", "XYZ Company", "[ { \"0\" : { \"size\" : \"S\", \"qty\" : 25.000000 },
\"1\" : { \"size\" : \"M\", \"qty\" : 50.000000 } } ]", "S", 25.000000, "M", 50.000000,
"clothing"
```

The array containing the stock is returned as the stock column value. The fields in the array are flattened into the columns stock_0_size, stock_0_qty, stock_1_size, and stock_1_qty.

This behaviour can be changed by using the Easysoft MongoDB Schema Editor to split a top-level array into virtual tables. For more information, refer to [Creating a Schema for your MongoDB Data](#).

SQL support

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

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

Example queries

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

```
SELECT * FROM Customers
```

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

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

- To filter records, use WHERE. For example:

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

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

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

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

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

70 Example queries

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

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

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

```
SELECT
    *
FROM
    SalesOrders
ORDER BY
    OrderDate ASC

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

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

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

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

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

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

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

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

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

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

72 Example queries

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

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

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

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

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

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

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

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

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

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

can be replaced with:

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

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

```
SELECT * FROM Customers LIMIT 10
```

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

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

Example inserts, updates, and deletes

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

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

- Here's a SQL Server linked server example:

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

- Here's an Oracle linked table example:

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

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

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

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

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

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

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

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

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

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

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

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

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

Index

A

Access

- importing MongoDB data, [36](#)
- linking MongoDB data, [36](#)

Auth Source attribute, [24](#)

B

Base

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

C

CA Directory attribute, [24](#)

CA File attribute, [24](#)

CharLen attribute, [21](#)

connecting to MongoDB, [18](#)

connection string attributes, [25](#)

CRL File attribute, [24](#)

D

data source attributes, [20](#)

data sources

- removing
 - on Windows, [17](#)

data type mapping, [64](#)

Database attribute, [22](#)

Default to WCHAR attribute, [22](#)

Description attribute, [21](#)

DG4ODBC

- working with MongoDB data, [41](#)

DSN attribute, [21](#)

DSN-less connections, [25](#)

E

Easysoft MongoDB Schema Editor, [25](#)

Easysoft ODBC-MongoDB Driver

- adding ODBC data sources
 - on Linux or UNIX, [18](#)
 - on Windows, [19](#)

arrays, [67](#)

connecting to MongoDB, [18](#)

data source attributes, [20](#)

data type mapping, [64](#)

DSN-less connections, [25](#)

embedded documents, [66](#)

installing

- on Linux or UNIX, [6](#)
- on Windows, [15](#)

logging, [31](#)

ODBC API support, [59](#)

scalar function support, [61](#)

Schema Editor, [25](#)

SQL support, [68](#)

uninstalling

on Linux or UNIX, [14](#)

on Windows, [17](#)

Excel

importing MongoDB data with Data Connection Wizard, [37](#)

importing MongoDB data with PowerPivot, [37](#)

importing MongoDB data with Query, [37](#)

G

Go

working with MongoDB data, [46](#)

I

Ignore Auth attribute, [23](#)

insert_nulls attribute, [21](#)

installing the Easysoft ODBC-MongoDB Driver, [6](#)

L

LibreOffice

working with MongoDB data, [45](#)

linked server

working with MongoDB data, [39](#)

Log File attribute, [23](#)

log files, [31](#)

Logging attribute, [23](#)

N

Node.js

working with MongoDB data, [47](#)

O

ODBC API function support, [59](#)

ODBC connection string attributes, [25](#)

ODBC data sources

- adding
 - on Linux or UNIX, [18](#)
 - on Windows, [19](#)
- removing
 - on Windows, [17](#)

Oracle

working with MongoDB data, [41](#)

P

Password attribute, [22](#)

PEM File attribute, [24](#)

PEM Password attribute, [24](#)

Perl

working with MongoDB data, [49](#)

PHP

working with MongoDB data, [52](#)

Power BI

importing MongoDB data, [38](#)

PowerPivot

importing MongoDB data, [37](#)

Preview Rows attribute, [23](#)

Python

working with MongoDB data, [54](#)

R

R

working with MongoDB data, [56](#)

S

SASL attribute, [24](#)

scalar function support, [61](#)

Schema Database attribute, [22](#)

Schema Password attribute, [22](#)

Schema Server attribute, [22](#)

Schema SRV_DNS attribute, [24](#)

Schema User Name attribute, [22](#)

Server attribute, [22](#)

SQL Server

working with MongoDB data, [39](#)

SQL support, [68](#)

SRV_DNS attribute, [23](#)

SSL attribute, [23](#)

T

trace files, [31](#)

U

uninstalling

on Linux or UNIX, [14](#)

on Windows, [17](#)

User Name attribute, [22](#)

W

Weak Validation attribute, [24](#)