

## Connect To An MS SQL Database from Linux and Mac OS X

Before you can connect to an MS SQL database from a Linux or OS X machine, there is some additional configuration that is required. This guide is intended to walk users through the configuration process. There are slight differences in configuration between Linux and OS X, so follow the appropriate section.

Questions, comments or concerns: [denverpsmith@gmail.com](mailto:denverpsmith@gmail.com).

For additional help, create a new issue request on our GitHub page:

<https://github.com/ODM2/ODMToolsPython/issues/new>

*Additional Technical References:*

<https://code.google.com/p/pyodbc/wiki/GettingStarted>

<http://richbs.org/post/43142767072/connecting-to-microsoft-sql-server-from-unix>

<https://code.google.com/p/django-pyodbc/wiki/FreeTDS>

## Configuration for Linux

### Required Packages

FreeTDS - A free implementation of the Tabular Data Stream protocol that is used by Microsoft for their database products.

| Manjaro Linux                        | Ubuntu   |
|--------------------------------------|--|
| <pre>\$ sudo pacman -S freetds</pre> | <pre>\$ sudo apt-get install freetds-dev freetds-bin tdsodbc</pre> |

unixODBC - An open specification for providing application developers with a predictable API with which to access Data Sources.

| Manjaro Linux                         | Ubuntu   |
|---------------------------------------|--|
| <pre>\$ sudo pacman -S unixodbc</pre> | <pre>\$ sudo apt-get install unixodbc unixodbc-dev</pre> |

myODBC - ODBC driver/connector for mariadb.

| Manjaro Linux                       | Ubuntu                                       |
|-------------------------------------|--|
| <pre>\$ sudo pacman -S myodbc</pre> | <pre>\$ sudo apt-get install libmyodbc</pre> |

## Configuring FreeTDS

Edit the `freetds.conf` file:

```
$ sudo vim /etc/freetds/freetds.conf
```

Add an entry for a new server as follows:

```
[SERVERNAME]
host = HOSTNAME or xxx.xxx.xxx.xxx
port = 1433
tds version = 8.0
```

Where `SERVERNAME` is the name of the database server you are going to connect to, `hostname` is the FQDN or IP address of the server. Here is an example configuration:

```
[iutahdbs]
host = iutahdbs.uwrl.usu.edu
port = 1433
tds version = 8.0
```

That's it. Now test the FreeTDS connection by making a query:

```
$ tsql -S SERVERNAME -U USERNAME
```

If you have provided a valid username, you will be prompted for a password. If everything checks out, you will get a prompt like this:

```
Password:
locale is "en_US.utf8"
locale charset is "UTF-8"
using default charset "UTF-8"
1>
```

You should now be able to query a database. Sometimes the database name can be a little bit tricky. Make sure you're using the right format (`DATABASE.dbo.TABLENAME`):

```
1> select * from iUTAH_Logan_OD.dbo.Variables
2> go
```

## Configuring ODBC

Edit the ODBC configuration file called `odbc.ini`.

```
$ sudo vim /etc/odbc.ini
```

Add the following, replacing the necessary parts:

```
[ODBC Data Sources]
ODBCNAME = Microsoft SQL Server

[ODBC_NAME_REFERENCE]
TDS_Version = 8.0
Driver = FreeTDS
Description = Add your description here.
Trace = No
Servername = SERVERNAME (from freetds.conf)
Database = DATABASE_SCHEMA_NAME

[Default]
Driver = /usr/lib/libtdsodbc.so
```

ODBC\_NAME\_REFERENCE - This can be anything you want to call it. You'll refer to this in your Django configuration (dsn).

SERVERNAME - This needs to be the same as SERVERNAME from freetds.conf

DATABASE\_SCHEMA\_NAME - This is the default database schema that will be used for queries.

Make sure that the driver points to a valid file. They are placed in different locations on some machines. On the Ubuntu server, I found it here: `/usr/lib/x86_64-linux-gnu/odbc/libtdsodbc.so`

If everything went well, you should be able to query the database like this:

```
$ isql -v ODBC_NAME_REFERENCE USERNAME PASSWORD
+-----+
| Connected!                                     |
|                                               |
| sql-statement                                |
| help [tablename]                             |
| quit                                          |
|                                               |
+-----+
SQL>
```

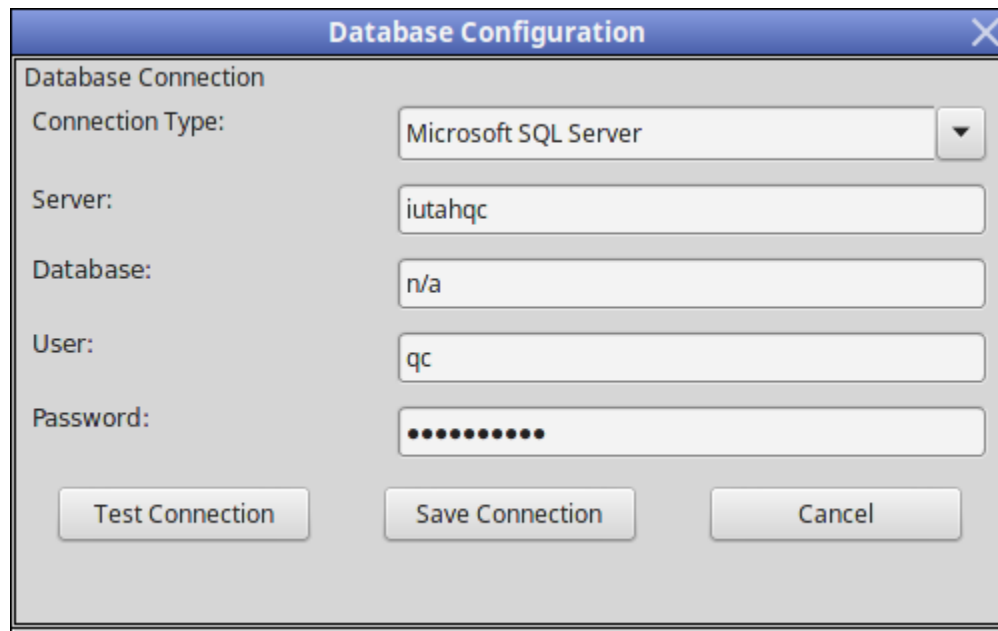
**\*Note** that if your password contains any special characters, such as an exclamation mark, you must put the password inside single quotations.

Test a query like this:

```
SQL> select * from dbo.Variables
```

As far as connecting to a database from Linux, that's all there is to it. To use the new configuration with ODMTools, select 'Microsoft SQL Server' as the connection type, use the DSN (ODBC\_NAME\_REFERENCE from odbc.ini) as the server, and enter the appropriate username and password. The database field is not applicable because it has already been set up in the odbc.ini file.

**\*Note** that in order to change the database you wish to connect to, you must edit your odbc.ini and freetds.conf files.

A screenshot of a 'Database Configuration' dialog box. The dialog has a title bar with the text 'Database Configuration' and a close button (X). Inside the dialog, there is a section titled 'Database Connection'. Below this title, there are five labeled input fields: 'Connection Type:' with a dropdown menu showing 'Microsoft SQL Server'; 'Server:' with a text box containing 'iutahqc'; 'Database:' with a text box containing 'n/a'; 'User:' with a text box containing 'qc'; and 'Password:' with a text box containing ten dots. At the bottom of the dialog, there are three buttons: 'Test Connection', 'Save Connection', and 'Cancel'.

## Configuration for Mac OS X

The easiest way to obtain the required packages is with homebrew, a package manager for OS X. To install homebrew, paste the following into the terminal:

```
ruby -e "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

### Required Packages

FreeTDS - A free implementation of the Tabular Data Stream protocol that is used by Microsoft for their database products.

unixODBC - An open specification for providing application developers with a predictable API with which to access Data Sources.

```
$ brew install freetds --with-unixodbc
```

### FreeTDS Configuration

Edit the freetds.conf file.

It will most likely be located here: `/usr/local/etc/freetds.conf`

If for some reason the file is in a different location, you can find it like this:

```
$ sudo find / -name freetds.conf
```

Append the following to the end of the file:

```
[SERVERNAME]
host = HOSTNAME or IP Address (eg. iutahqc.uwrl.usu.edu)
port = 1433
tds version = 7.1
```

Where SERVERNAME is the name of the database server you are going to connect to, hostname is the FQDN or IP address of the server. Here is an example configuration:

```
[iutahqc]
host = iutahqc.uwrl.usu.edu
port = 1433
tds version = 7.1
```

Now test the FreeTDS connection by making a query:

```
$ tsql -S SERVERNAME -U USERNAME
```

If you have provided a valid username, you will be prompted for a password. If everything checks out, you will get a prompt like this:

```
Password:
locale is "en_US.utf8"
locale charset is "UTF-8"
using default charset "UTF-8"
1>
```

You should now be able to query a database. Sometimes the database name can be a little bit tricky. Make sure you're using the right format (DATABASE.dbo.TABLENAME):

```
1> select * from iUTAH_Logan_OD.dbo.Variables
```

```
2> go
```

### unixODBC Configuration

First, edit the ODBC configuration file.

It's probably located at `/usr/local/etc/odbc.ini`

You can also search for it with the following command:

```
$ sudo find / -name odbc.ini
```

Modify the file to contain the following, making modifications as needed:

```
[ODBC Data Sources]
ODBCNAME = Microsoft SQL Server

[ODBC_NAME_REFERENCE]
TDS_Version = 7.1
Driver = FreeTDS
Description = Add your description here.
Trace = No
Servername = SERVERNAME (from freetds.conf)
Database = DATABASE_SCHEMA_NAME

[Default]
Driver = /usr/lib/libtdsodbc.so
```

ODBC\_NAME\_REFERENCE - This can be anything you want to call it. **This is the DSN that you will use as the server in ODMTools.**

SERVERNAME - This needs to be the same as SERVERNAME from freetds.conf

DATABASE\_SCHEMA\_NAME - This is the default database schema that will be used for queries.

Make sure that the driver points to a valid file. Here is an example configuration that matches the FreeTDS config that we have just setup:

```
[ODBC Data Sources]
ODBCNAME = Microsoft SQL Server

[iutahqc]
TDS_Version = 7.1
Driver = FreeTDS
Description = Add your description here.
Trace = No
```

```
Servername = iutahqc
Database = iUTAH_Logan_OD

[Default]
Driver = /usr/local/lib/libtdsodbc.so
```

If everything went well, you should be able to query the database like this:

```
$ isql -v ODBC_NAME_REFERENCE USERNAME PASSWORD
+-----+
| Connected!                                     |
|                                               |
| sql-statement                                |
| help [tablename]                             |
| quit                                          |
|                                               |
+-----+
SQL>
```

**\*Note** that if your password contains any special characters, such as an exclamation mark, you must put the password inside single quotations.

Test a query like this:

```
SQL> select * from dbo.Variables
```

At this point, you should be able to connect to the database via command line. However, in order to access the database via python, we must configure pyodbc.

First, create a new file called `FreeTDS_driver` and add the following:

```
[FreeTDS]
Description = TDS Driver (MS SQL)
Driver = /usr/local/lib/libtdsodbc.so
```

Save the file, and run this command to register the driver:

```
$ sudo odbcinst -d -i -f FreeTDS_driver
```

### pyodbc Configuration

Download the source code [here](#). Unzip the directory where ever you want.

Edit `setup.py` within the `pyodbc` directory. Look for the line that says “OS/X now ships with iODBC.” Right underneath that, remove the letter `i` from the statement:

```
settings['libraries'].append('iodbc')
```

so that it reads:

```
settings['libraries'].append('odbc')
```

```
145
146     elif sys.platform == 'darwin':
147         # OS/X now ships with iODBC.
148         settings['libraries'].append('iodbc')
149
150         # Apple has decided they won't maintain the iODBC system in OS/X
151
152         # For now target 10.7 to eliminate the warnings.
```

Now that the `setup.py` file is configured correctly, `pyodbc` is ready to be installed. From your virtual environment, do a `pip install` of `pyodbc` and use the modified package.

```
$ pip install -e PATH_TO_PYODBC
```

For example:

```
(odmtools)denversmith@HsaKBaw:~$ pip install -e ~/Downloads/pyodbc-3.0.7
Obtaining file:///Users/denversmith/Downloads/pyodbc-3.0.7
Installing collected packages: pyodbc
  Running setup.py develop for pyodbc
Successfully installed pyodbc
```

At this point, we need to update `pyodbc` to the latest version and give it the correct paths to required libraries. To do so, run the following command in your terminal:

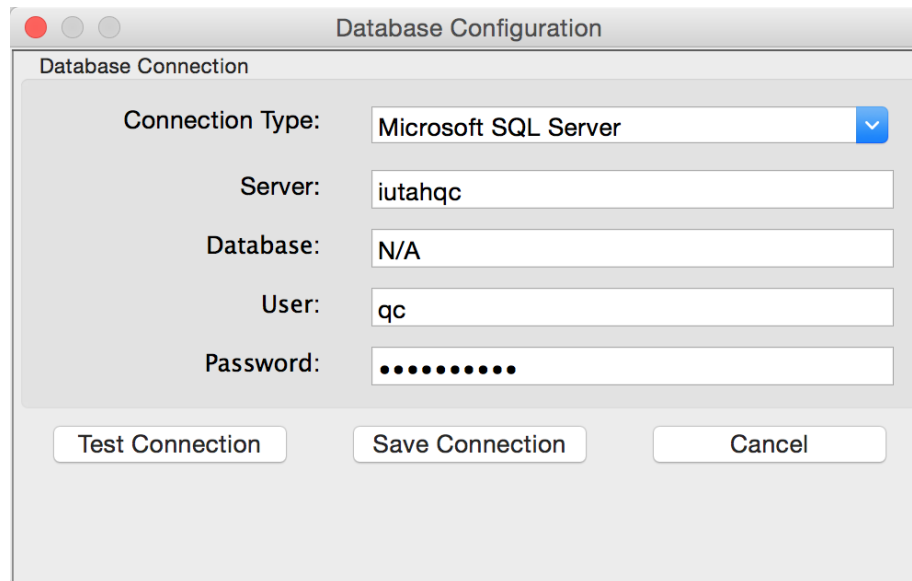
```
$ pip install -U \
    --global-option=build_ext \
    --global-option="-I/usr/local/include" \
    --global-option="-L/usr/local/lib \
    pyodbc
```

If everything was successful, your machine is now configured to connect to an MS SQL database. To use the new configuration with ODMTools, select ‘Microsoft SQL Server’ as the connection type, use the DSN (ODBC\_NAME\_REFERENCE from `odbc.ini`) as the server, and



enter the appropriate username and password. The database field is not applicable because it has already been set up in the `odbc.ini` file.

**\*Note** that in order to change the database you wish to connect to, you must edit your `odbc.ini` and `freetds.conf` files.



The image shows a 'Database Configuration' dialog box with a title bar containing standard window controls (red, yellow, and grey buttons). The main area is titled 'Database Connection' and contains several input fields: 'Connection Type' is a dropdown menu set to 'Microsoft SQL Server'; 'Server' is a text field containing 'iutahqc'; 'Database' is a text field containing 'N/A'; 'User' is a text field containing 'qc'; and 'Password' is a text field filled with ten dots. At the bottom of the dialog are three buttons: 'Test Connection', 'Save Connection', and 'Cancel'.

| Field           | Value                |
|-----------------|----------------------|
| Connection Type | Microsoft SQL Server |
| Server          | iutahqc              |
| Database        | N/A                  |
| User            | qc                   |
| Password        | .....                |