

# **ODBC driver and MBS SQL Functions in FileMaker**

We have made a ton of example code in the last decade showing how to connect to Microsoft SQL Server from MacOS or Linux using FreeTDS and our [MBS FileMaker Plugin](#). And there are two ways to do it:

## **1. With loading driver directly**

```
Set Variable [$result; Value: MBS("SQL.SetClient"; $Connection; "ODBC")]
#Tell plugin where freetds library is
Set Variable [$result; Value: MBS("SQL.SetConnectionOption"; $Connection; "ODBC.LIBS"; "/Users/cs/Desktop/libtdsodbc.dylib")]
#Connect to database
Set Variable [$result; Value: MBS("SQL.Connect"; $Connection; "DRIVER={FREETDS};Server=192.168.2.32;Uid=SA;PWD=test;Database=test;TDS_VERSION=7.2;Port=1433")]
```

As you see, we point to the dylib as ODBC library, load it and connect through it.

## **2. With loading iODBC and having driver in connection string**

```
Set Variable [$result; Value: MBS("SQL.SetClient"; $Connection; "ODBC")]
#Tell plugin where freetds library is
Set Variable [$path; Value: "/Users/cs/Desktop/libtdsodbc.dylib" ]
#Connect to database
Set Variable [$result; Value: MBS("SQL.Connect"; $Connection; "DRIVER={" & $path & "};Server=192.168.2.32;Uid=SA;PWD=test;Database=test;TDS_VERSION=7.2;Port=1433")]
```

As you see, we let the plugin load libiodbc.dylib (ODBC manager) and then have it load the library.

## **What is better?**

Both ways do work, so for most customers there is no difference. But the second way is better as the libiodbc does some additional services like text encoding conversions, provide additional functions like a directory of the various database drivers, functions to browser for databases and a way to work with data sources.

So we recommend in general that all customers prefer the second way over the ODBC manager to make sure all things work well. For the next weeks, we'll try to update a couple of blog posts and examples projects.