Getting Started With FileMaker Pro 5.5 ODBC Connectivity

And An Example of Connecting FileMaker to an External Data Source

Getting Connected

The first step in getting FileMaker® software to "talk" to an external ODBC data source is to set up the ODBC DSN (Data Source Name). Go to the start menu and choose Settings -> Control Panels. If you're using Windows 95/98/ME double click the "ODBC 32" application – if you're using Windows 2000, double click "Administrative Tools" and then double click "Data Sources (ODBC)". For this article, we're going to connect to a Microsoft SQL Server database that resides on the local machine.

When creating a new ODBC DSN, you can use either the "File" or "System" type of DSN. In general, you'll want to use a System DSN (see the Microsoft web site at http://www.microsoft.com for more on ODBC DSN).

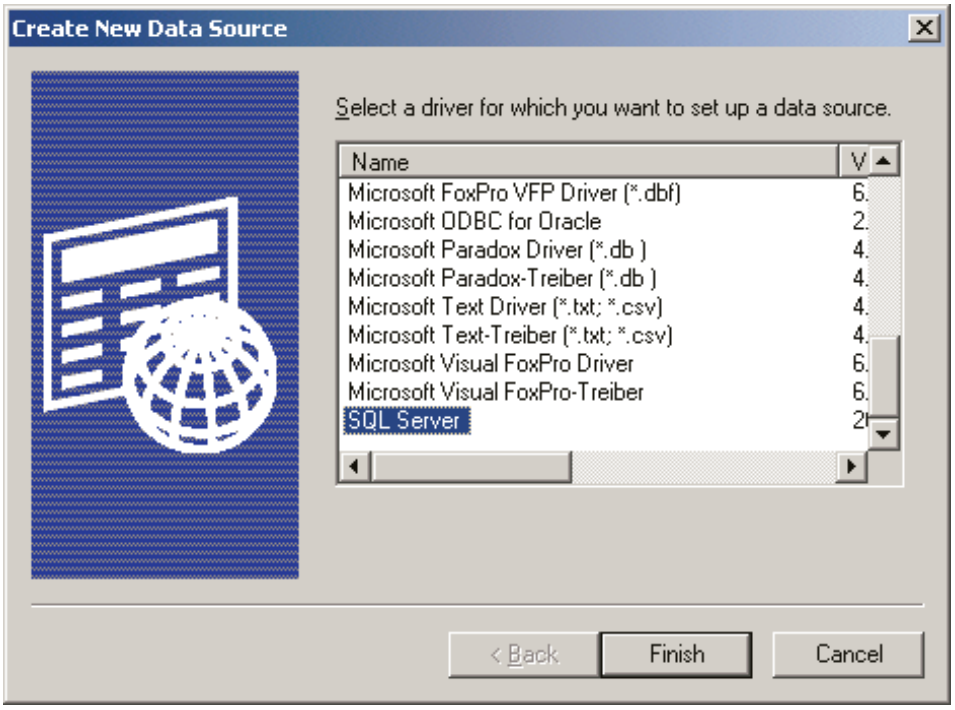
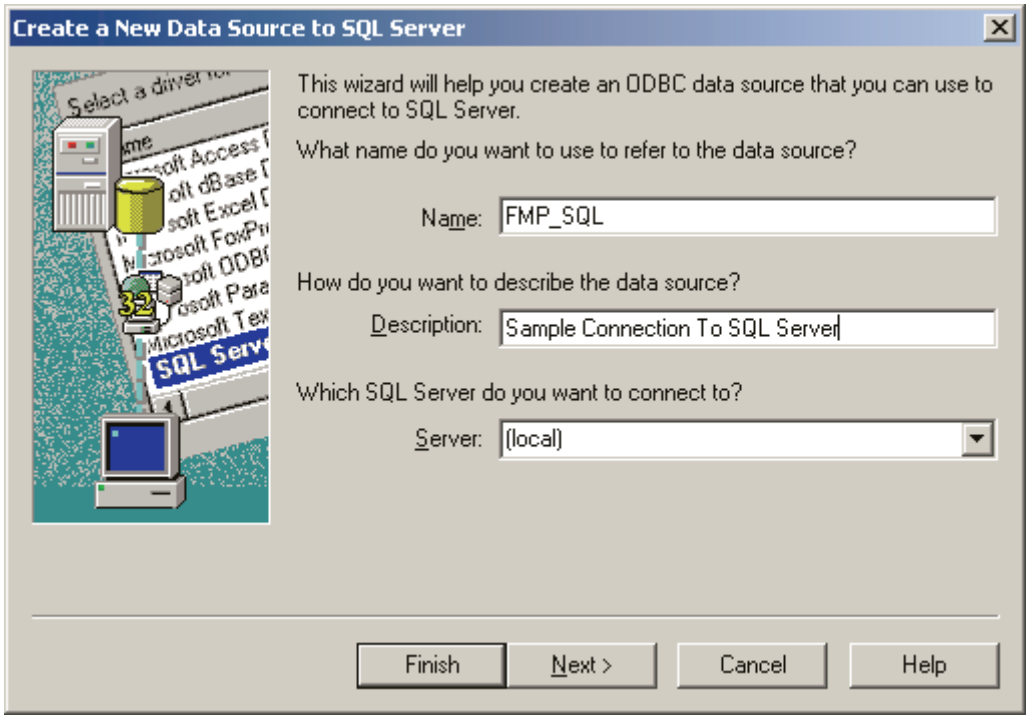


Figure 1 – When setting up the ODBC DSN, you will first choose the type of DSN to create.

Step 1 – Specify The Connection
The next step is to choose the appropriate ODBC driver. In this case, we want to connect to Microsoft SQL Server. You'll notice that there's a driver called "FileMaker Pro SQL Server" – DO NOT use this driver. This driver is for connecting FROM SQL Server TO FileMaker. Instead, use the driver called "SQL Server" and click "Finish".

Step 2 – Choose the type of data source to which you are connecting.
The next screen will allow you to give the DSN a name and description. Name your DSN something simple. A name like "SQL_Server" is a lot easier than "095484_JTURR_SERVER." Plus, if you're going to deploy your solution on a network or to more than one user – each user MUST

set up a DSN with **exactly** the same name – or your solution will not work. You must also specify the name or IP address of the server you're going to connect to. In our case, since SQL Server is running on the same machine, we can just choose the "(local)" option (you can change the server at any time).



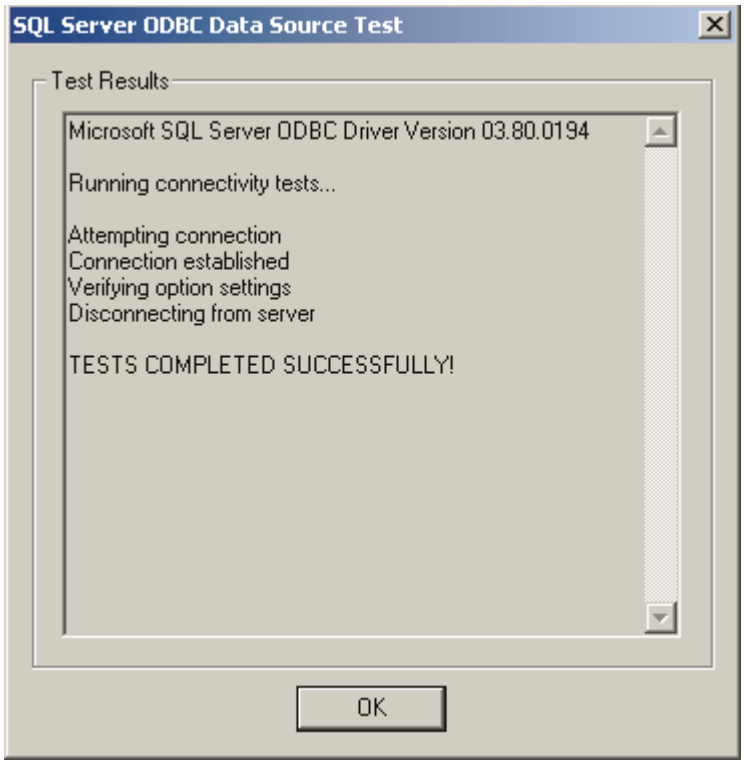
Step 3 – Give your data source a simple name and a description.
The next step is to choose an authentication method. Click the radio button next to "With SQL Server authentication using a login ID and password entered by the user" and enter "sa" (no quotes) in the username and leave the password blank. If you're connecting to a database in a corporate environment – chances are they've changed this "master" default password to something else – so get a login id and password from your Network Administrator or DBA.

Step 4 – Set the authentication method and login information.
On the next screen, you can set the "Default Database" – which you would want to do if you were going to tie your solution to a specific SQL Server database – but for our example just leave the checkbox UNCHECKED and click "Next."

Step 5 – Setting the database context.
You can change the default database to a specific database if your solution requires it. For our example, leave it unchecked.

Step 6 – Other Options
The last dialog allows you to set other, advanced options such as logging, database language, encryption, etc. For our example leave everything as entered and click the "Finish" button.

Step 7 – Testing the connection.
Finally, the last dialog presents a summary of the information you have entered, and provides a button called "Test Data Source." You DO want to click "Test Data Source."



After clicking the "Test Data Source" button, you should see "TESTS COMPLETED SUCCESSFULLY!" in the dialog box. Once you see this message you can click "OK" and close the ODBC control panel.

SQL Query flexibility via new FileMaker Pro 5.5 features

Now that we have a valid ODBC DSN set up, we can move to the FileMaker side. One of the great new ODBC connectivity features in FileMaker Pro 5.5 is the "Execute SQL" script step. Below, we will see how to create a SQL database and add/delete records from it – but let's start by executing a dynamic SQL query to get data INTO FileMaker.

In previous FileMaker Pro versions (4.1 through 5.0), importing data from an external source was done via ODBC Import feature. The same is true in FileMaker Pro 5.5 – but the HUGE difference in FileMaker Pro 5.5 is the fact that we can specify a field for the SQL query rather than having to create a separate import script for each SQL query. This means that you can use calculations, globals, and scripts to define the query to run and have a SINGLE import script.

This new FileMaker Pro 5.5 feature gives developers great improvements in productivity and control of the FileMaker software connection to external data sources.

For our example we will create a simple 11-field database with one global text field called "gSQL" and plain text fields "Text01" through "Text10".

Once the database is created, we will create a generic "Execute SQL" script. First, we perform the import so we can save the import order in a script:

1. Choose "Import Records…" from the "File" menu.

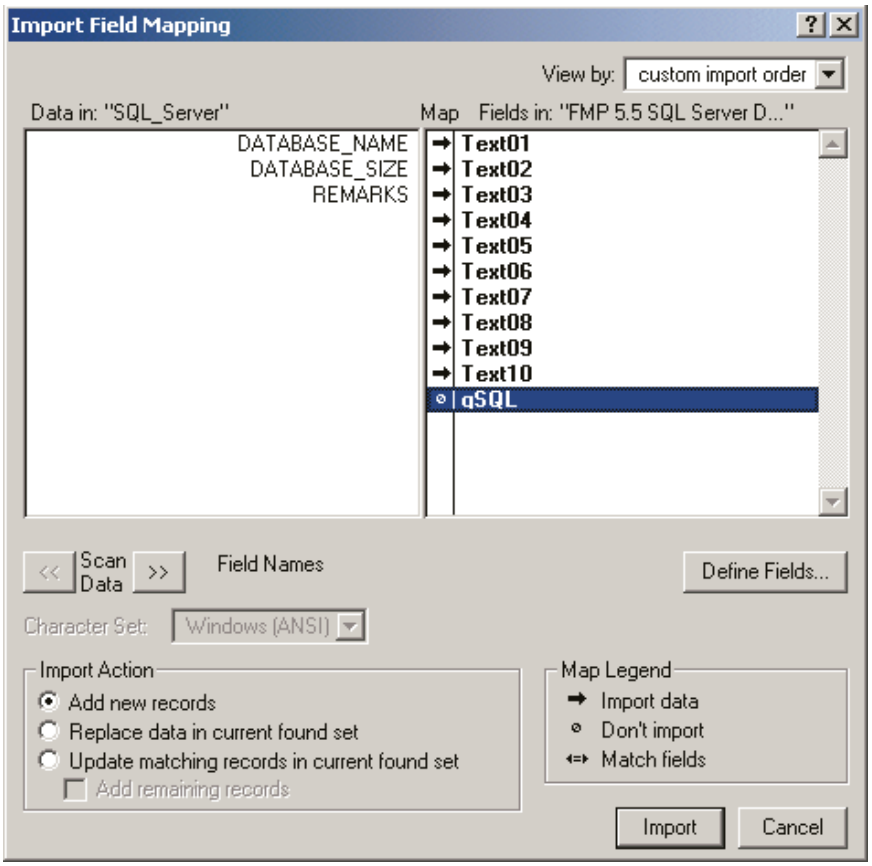
2. From the "Files of type" pop-up menu at the bottom of the dialog, choose "ODBC Data Sources (*.*)."

3. Choose the "FMP_SQL" DSN from the list

4. Enter your user name and password. DO check the "Save user name and password" checkbox at the bottom of the dialog and click "OK."

5. In the "SQL Builder" dialog – simply enter `sp_databases` and click the "Execute" button.

6. In the "Import Field Mapping" dialog, set the import flags (so they show a right-facing arrow) for text fields "Text01" to "Text10, move the "gSQL" field to the bottom of the list and uncheck the import flag (so it shows an open circle). Make sure the "Add new records" radio button is checked and click "Import."



By setting up the import order in this fashion – you can execute any generic SQL statement and see the first 10 fields of the result.

The SQL statement we executed, `sp_databases`, is a stored procedure (think of it as the equivalent of a FileMaker script) in the "master" database in SQL Server. It will return the name of each database and the size in bytes of each database. If your query executed successfully, you should see the name of each of the databases in field "Text01" and the size (in bytes) of each database in field "Text02".

Now, we're going to create a script called "SQL Import" that we can use to execute ANY SQL statement and automatically import the data into our database:

1. Open ScriptMaker® and create a new script called "SQL Import"

2. Click the "Clear All" button.

3. Double click "Import Records" from the "Available Steps" on the left.

4. Check the "Restore import order" and "Perform without dialog" checkboxes.

5. Click the "Specify…" button in the lower right. The next three steps are identical to the import we just performed:

6. Choose "ODBC Sources (*.*)" from the "Files of type" pop-up at the bottom of the dialog.

7. Choose the "FMP_SQL" DSN that we created earlier.

8. Enter the user name and password – and DO check the "save user name and password" checkbox.

9. After entering the user name and password, you'll see a brand new dialog that allows you to specify the SQL query using a string you type in yourself, using the "Query Builder" (the only option in FileMaker Pro 4.1 through 5.0) or, taking advantage of the new FileMaker Pro 5.5 flexibility, a field. Click the radio button next to "field value:" and choose the "gSQL" field. Click "OK".

10. Click "OK" – our script only needs a single script step. In the keep or replace options dialog box, check the radio button next to "Replace" for the "Import Order". Click "Done".

Performing the SQL Query

Now we will run our first query. As part of the standard installation of SQL Server, there are two databases that are automatically created – "pubs" and "Northwind". For our first query, enter the following into the "gSQL" field:

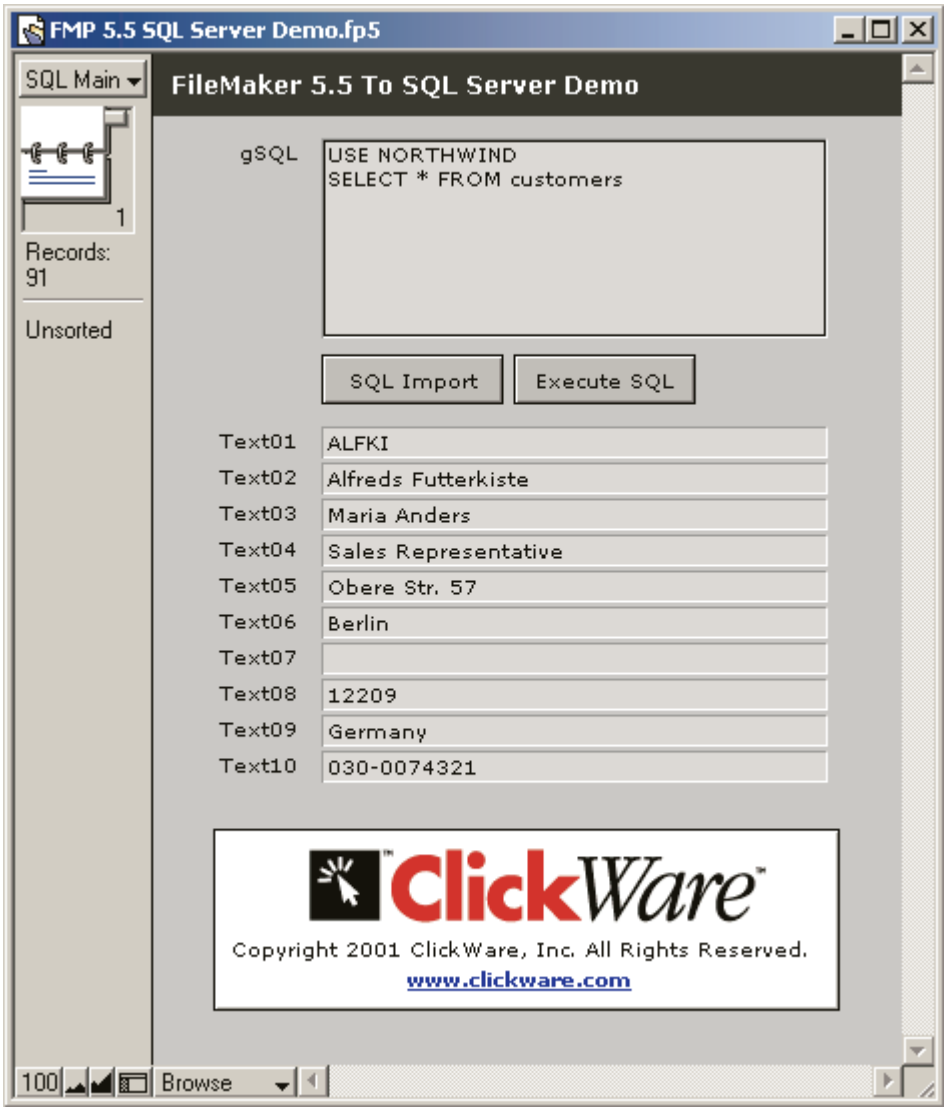
```
USE PUBS
SELECT au_id, au_fname, au_lname FROM authors
```

Choose the "SQL Import" script – and you should see a found set of 23 records with the author ID in field "Text01", the author first name in field "Text02" and the author last name in field "Text03."

Let's issue another SQL command against a different database. Enter this query into the gSQL field:

```
USE NORTHWIND
SELECT * FROM customers
```

Choose the "SQL Import" script – and you should see a found set of 91 records with the customer ID, company, contact name, contact title, address, city, region, postal code, country and phone number for each record.



Security and Using The Execute SQL Script Step

The Execute SQL ScriptMaker step, introduced in FileMaker Pro 5.5, is a very powerful command and allows the database developer to communicate from FileMaker to the SQL data source. The Execute SQL script step will execute any valid SQL command, including INSERT, UPDATE, DELETE, CREATE, ALTER and DROP. While this powerful command allows the FileMaker developer to perform any action on the database, the execution of the SQL is left to the back-end data source, and must be valid based on security set up on the source. That is, FileMaker will pass the SQL command to the data source; the source checks that the user has the proper permission to execute the SQL command. If the proper permissions are in place, the SQL command is executed.

The security and the permissions to execute or not execute DML (Data Manipulation Language, i.e. INSERT, UPDATE, DELETE) or DDL (Data Definition Language – i.e. CREATE, ALTER, DROP) SQL statements is entirely controlled by the external data source. Most popular SQL database platforms enable system administrators, DBAs (Data Base Administrator), and other IT staff to control the permissions assigned to each username/login.

For example, an administrator may grant the SELECT permission on every table in a certain database, but limit the INSERT command to a single table, and completely disable UPDATE and DELETE. Because the ODBC connection must log in to the database using a username and password (like any other user) – special groups with limited access rights can be established to ensure data integrity.

Now back to our example: We have imported data into FileMaker Pro 5.5 using a dynamic query (and Import Records) – wouldn't it be great if we could INSERT, UPDATE or DELETE data in a SQL database as well? With FileMaker Pro 5.5, it's as simple as writing the SQL statement and executing (assuming your username/password has the permission to do so).

For our example, in this next section we're going to insert a new record into an external data source. This means we're going to have to create a new script called "Execute SQL":

1. Open ScriptMaker and create a new script called "Execute SQL"
2. Click the "Clear All" button.
3. Double click "Execute SQL" from the "Available Steps" on the left (at the very bottom).
4. DO check the "Perform without dialog" checkbox.
5. Click the "Specify…" button in the lower right.
6. Click the "Specify…" button in the "Specify ODBC Source" section in the top of the dialog and choose the "FMP_SQL" DSN that we created earlier.
7. Enter the user name and password – and DO check the "save user name and password" checkbox.
8. Click the radio button next to "field value:" and choose the "gSQL" field. Click "OK".
9. Click "OK" – our script only needs a single script step. Click "Done.

In the "gSQL" field – type the following SQL statement:

```
USE NORTHWIND
INSERT INTO customers (companyname, contactname) VALUES ('ClickWare,
Inc., 'Bob Cusick')
```

When you issue a SQL statement that contains values in a TEXT field, you MUST put single quotes around the text. If you have a single quote within the text, you MUST use two single quotes in place of the single quote. For example, if we were to add another record for Tommy O'Brien from ABC Company, we would use the following SQL statement:

```
USE NORTHWIND
```

```
INSERT INTO customers (companyname, contactname) VALUES ('ClickWare,
Inc., 'Tommy O''Brien')
```

Once again, if you didn't receive an error it will seem like nothing happened, but you've created another record in the your external SQL database. To test this, we'll enter another query that is the equivalent of the FileMaker "Find All Records" command:

```
USE NORTHWIND
SELECT * FROM customers
```

You should have a found set with the two records you added.

Conclusion

As you can see, the connections are simple and straight-forward. We can take these concepts much further: In addition to inserting, updating and deleting data from the SQL data source, you can use FileMaker to create reports, or to update Website data with data from FileMaker, or to import online orders from a Web server into a database. With a combination of imports and some scripting you can create an entire "front end" to your SQL databases using FileMaker Pro 5.5.

About the Author

Bob Cusick is President of ClickWare, Inc., a member of the FileMaker Solutions Alliance, and a Technical Editor of the FileMaker Advisor Magazine. He has been creating custom FileMaker, SQL, and browser-based solutions since 1990, and is an internationally recognized author, speaker and trainer. http://www.clickware.com, bob@clickware.com.