

Turning onto a Two-Way Street A Tutorial on The SAS System and ODBC

Peter J. Lund
Washington State Office of Financial Management

The SAS System and ODBC

ODBC (Open Database Connectivity) is a Microsoft standard which provides a common interface through which compliant applications can exchange data. Beginning with version 6.10, the SAS System has allowed access to ODBC-compliant databases, such as Microsoft Access, Paradox, Oracle and Excel, through the SAS/ACCESS Interface to ODBC module. An exciting addition was introduced in SAS 6.11 for Windows, the SAS ODBC driver. For the first time ODBC-compliant applications can directly access SAS datasets. The combination of the SAS/Access to ODBC module and the SAS ODBC driver allows the SAS System to continue to be a powerful part of an integrated data management solution.

The goal of this tutorial is to be conceptual and practical. To demonstrate how to make ODBC work with SAS, rather than the details of how ODBC works. It is like Driver's Ed. and Auto Mechanics. Both very useful, but just as one can learn to drive without ever looking under the hood, one can begin to use ODBC without understanding all the nuts and bolts of how it works.

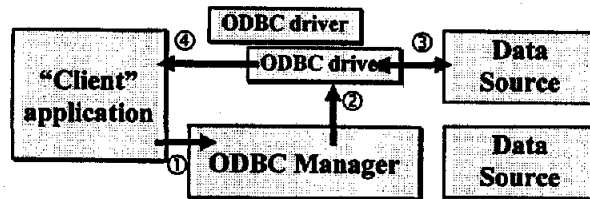
ODBC Components

Having said that, it might still be helpful to begin with a quick "conceptual" view of how ODBC works. Here are a few terms that are often used in conjunction with ODBC and will help lay a foundation for our discussion:

- **ODBC Driver** - application-specific software (DLL) which allows access to a particular type of database. For example, the SAS ODBC driver allows ODBC-compliant applications access to SAS datasets. Drivers are usually provided by the database vendor, though there are third-party vendors who write and supply ODBC drivers. Note: The SAS ODBC driver is a freely distributable DLL.

- **ODBC Data Source** - the description of how to get to a particular database. This includes which driver to use and where the database is physically located. (Note: For some applications, including SAS, specifying the location of the software is also part of the data source definition.)
- **ODBC Driver Manager** - operating system component which manages calls to ODBC data sources.
- **ODBC Administrator** - operating system component which handles setup of drivers and configuration of data sources.

There are also third-party vendors, licensed by Microsoft, which supply versions of the ODBC Driver Manager and ODBC Administrator which ODBC to run in non-Windows environments, such as OS/2 and UNIX.



Think of ODBC like this:

1. An application references an ODBC data source and requests some data. The request is passed to the ODBC Driver Manager.
2. The ODBC Manager looks up the data source name and the appropriate driver is loaded.
3. The driver evaluates the data request and retrieves the data, "converting" it to the ODBC standard.
4. The requesting application converts the data from the ODBC standard to its own format.

Please note: Steps 3 and 4 actually work on the "data stream". No "ODBC" copy of the data is generated.

SAS can function both as a client application using the SAS/Access to ODBC module and as

a server offering SAS datasets as a data source using the SAS ODBC driver.

For our example, let's imagine that we're managing a fantasy baseball league. All of the player statistics come to us in a Microsoft Access database. We want to get that data into SAS to analyze. The results of our analysis will be stored in SAS datasets. When we're done with our analysis, we want to be able to treat those SAS datasets as though they were part of the Access database. With ODBC there is no need to make a SAS-readable copy of the Access tables and no need to make Access-readable copies of the SAS datasets.

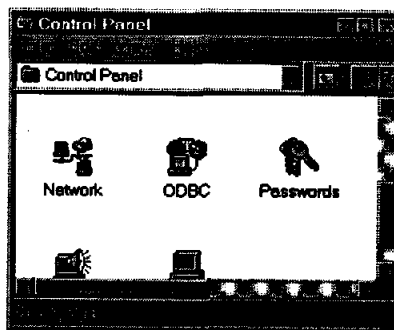


Figure 1

Setting up a SAS ODBC Data Source

To allow Access to access our SAS datasets we will use the SAS ODBC driver. There are only a few simple steps involved in setting up a SAS data source that will allow Access to treat our datasets as if they were part of the database.

First, open the Windows Control Panel and double click on the ODBC icon (Figure 1). This starts the ODBC Administrator and open up the Data Sources window (Figure 2). All currently defined data sources, and the associated driver, are displayed in the window.

There are a number of selections that can be

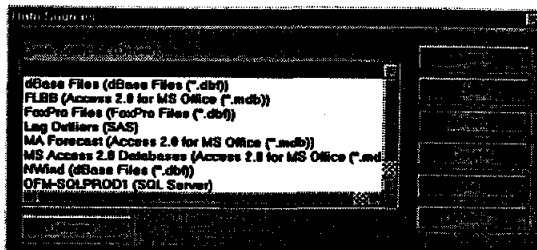


Figure 2

made in this window:

Setup... allows you to edit the information of the currently highlighted data source. Note: Double-clicking an entry in the data source list is the same as clicking Setup.

Delete the currently highlighted data source definition. (Does not affect the data associated with that definition.)

Add a new data source.

Drivers... displays a list of currently installed ODBC drivers. From here, drivers can be added or removed.

Options... sets up of ODBC tracing.

Remember, a SAS data source is simply a description to ODBC of the following:

1. Which driver to use
2. Where the datasets are located
3. Where the SAS software is located

To add our SAS data source, click on Add... and a list of currently defined ODBC drivers is displayed (Figure 3). Double-click on SAS. (Note: If SAS is not in the list, you need to go back to the Data Sources Window. Click Drivers..., then Add... and install the SAS ODBC driver.)

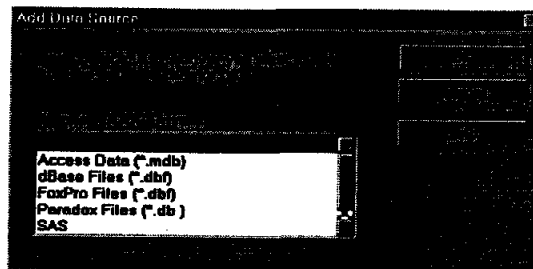


Figure 3

The SAS ODBC Driver Configuration window is displayed, with 4 tabs:

- **General:** data source name information
- **Servers:** SAS software location
- **Libraries:** SAS dataset location
- **SQL Options:** just like it says, SQL options.

The General tab is displayed first, by default, but let's look at them in an order that makes a little more conceptual sense.

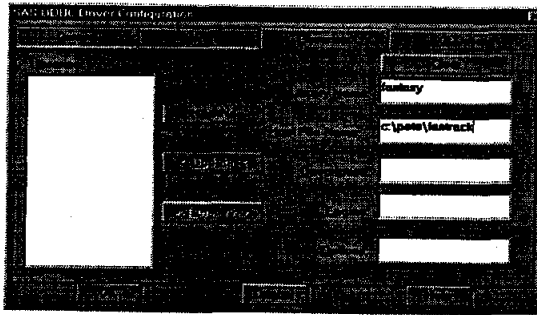


Figure 4

Libraries Tab (Figure 4) - the information on this tab tells ODBC where the SAS datasets are located. Think of this tab as the place where you enter your libname statements.

Library Name sets up the libref. It is a required field.

Host File Name sets up the path. It is a required field.

Description is an optional text description.

Engine is the SAS version of the datasets stored in the library. By default, it is the version of SAS running on the server described for this data source (see below).

Options are SAS options set for this library. The only option supported at this time is ACCESS=READONLY.

The entries in Figure 4 are analogous to the following SAS statement:

```
libname fantasy 'c:\pete\fantrack';
```

When you've entered your library information, click on <<Add<<. It will be placed in the library list on the left of the screen. Multiple libraries can be assigned to a data source.

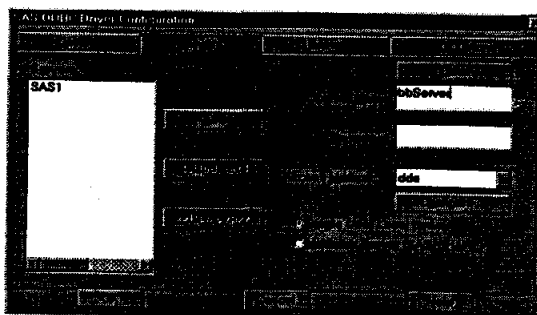


Figure 5

Servers Tab (Figure 5) - tells ODBC where the SAS software is located.

Server Name is a reference given to this particular instance of SAS. It must follow SAS naming rules, i.e. 8 characters or less, starts with an alpha, limited special characters, etc.

Password is required if the server on which the SAS software resides requires a password.

Access Method will be either DDE or TCP. If your SAS is running on your local PC, select DDE. This is true whether SAS is loaded on your local PC or your network. If SAS is running on a remote server, select TCP.

When you've entered the above information, press **Configure...** and either the Local DDE Options window (Figure 6) or the TCP Options window will appear. In our example, SAS is running on a local PC so the Local DDE Options window is displayed

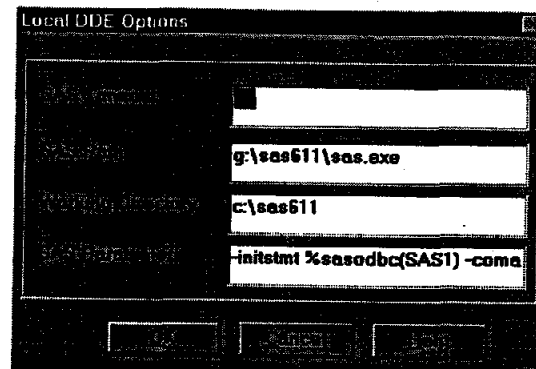


Figure 6

You're going to define the path, working directory and command line options. It is very similar to setting up a SAS icon for starting an interactive session. The parameters listed in the SAS Parameters field are those necessary to initialize the session and start PROC ODBCSEV. You will rarely, if ever, need to change them. See the SAS/ACCESS Interface to ODBC technical report for more details.

Referencing the SAS data source in another application causes a SAS session to start. The SAS Timeout option is the number of seconds to wait for that session to start before returning an ODBC error. The default is sixty (60) seconds and is more than sufficient in most cases.

Click OK to return to the Servers tab. Click <<Add<< to move the server name to the Servers list on the left of the screen.

General Tab (Figure 7) - tells ODBC what you want to call your data source and which server definition to use.

Data Source Name is used to give a descriptive name to the data source. This is the name that will display in the Data Sources window. It can contain spaces, but not the following special characters: [] () ? * = ! @. It is a required field.

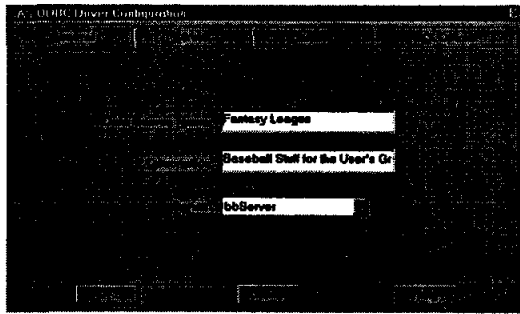


Figure 7

Description is used to give a longer, more informative, description. It is an optional field.

Server lists all the currently defined servers (see Servers tab description above).

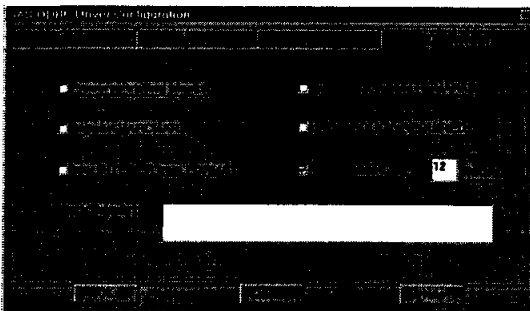


Figure 8

SQL Options Tab (Figure 8) - The following description is taken from the SAS ODBC Driver configuration on-line help system (emphasis added). "The options on this page affect the interaction between the SAS ODBC driver, SAS, and ODBC-compliant applications. The default selections should work for the majority of ODBC-compliant applications, but they may be changed depending on an application's needs." Please refer to this or the written documentation for more details on the effect and potential impact of each option.

Our SAS data source is now set up. We've told ODBC where our datasets are located and where SAS is located and assigned the SAS ODBC driver. That's all we need to allow Access to use our datasets.

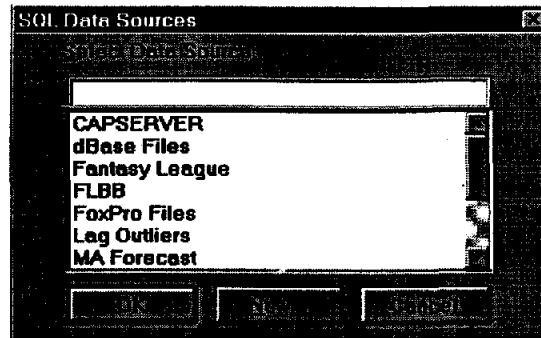


Figure 9

In Access we'll "Attach" these datasets to the existing database. Choose File..., Attach Table... and select <SQL Database> from the list. This will display the list of currently defined ODBC data sources (Figure 9). This is the same list as in the ODBC Administrator Data Sources Window. Select "Fantasy League" and a list of datasets in the library we defined will display (Figure 10). Select "flbbteam" and it will now appear in the tables list of the database (Figure 11).

When a SAS dataset is attached as a table in a database any changes, additions or deletions made in the database application affect the SAS datasets. For the most part the structure of the datasets cannot be changed.

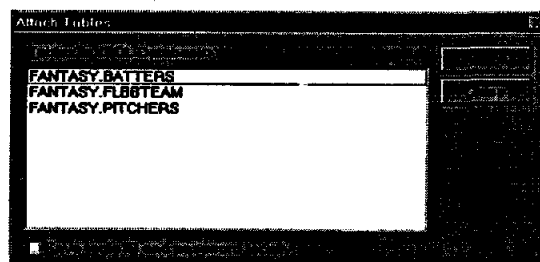


Figure 10

Note: Some applications, like Microsoft Access, require an attached table to be indexed in order to be updatable. The index can be created in SAS (using PROC SQL or PROC DATASETS) or created by an a query in the client application. In the later case, the index is stored as part of the database and no .SIS2 file is created. In other words, as far as SAS is concerned the dataset is not indexed.

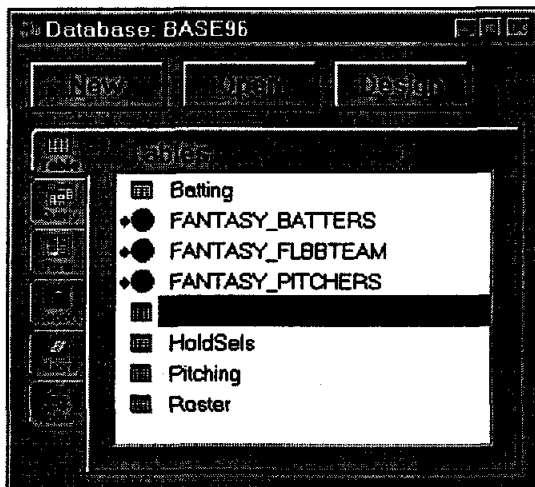


Figure 11

Accessing another ODBC database using SAS/ACCESS

Accessing another ODBC database from SAS requires the SAS/ACCESS to ODBC module. Once this module is loaded you have access to any database for which an ODBC driver is installed. The data source setup is specific to each database. The concept is similar to the SAS ODBC driver configuration described above, but the process will be different.

ODBC and PROC SQL

Initial access to ODBC databases from SAS is always done with PROC SQL or the SQL Query Window in SAS/Assist. The Query Window can be activated by starting SAS/Assist or by entering "query" on the SAS command line. It offers a "point-and-click" interface to SQL and can access ODBC data sources. Here, we'll examine the components of a simple SQL query, paying special attention to those parts which deal with the ODBC connection.

As mentioned earlier, the statistics for our fantasy baseball league come in a Microsoft Access database. We want to access and manipulate this data in SAS without having to make an intermediate copy from Access in a form that SAS can read directly, like an ASCII file. Using SAS/Access to ODBC we can get the data from the Access database tables without having to do anything in Access whatsoever. Here's a simple example which includes all the components necessary to access an ODBC data source:

```

① proc sql ;
  ② connect to odbc as stats (dsn=④Fan Stats");
  ⑤ create table batting as
    select * ⑥
  ⑦ from connection to stats
        (select * ⑧
         from BatterStats);
  ⑨ disconnect from stats ;
quit;

```

Let's look at each piece of this query.

1. proc sql;

All access to ODBC databases is done with PROC SQL.

2. connect to odbc...

Initialized contact with the ODBC Driver Manager to load a particular driver and set up access to a particular data source (see 4). Multiple ODBC connections can be established in a PROC SQL (see 3).

3. ...as stats...

An optional alias for this connection. If more than one connection is setup, the alias is required.

4. (dsn="Fan Stats");

The data source name that was assigned to the database in the ODBC administrator. Information about the type of database, the ODBC driver and the location of the database are maintained by the ODBC Driver Manager. All you have to remember is the data source name, in this case "Fan Stats".

If the data source requires a user id and password, these are coded here as well.

5. create table batting as

We want to create a SAS dataset called BATTING which will contain data from an Access table.

There are two options on the CREATE statement:

CREATE TABLE will create a SAS dataset. In our example we will create a dataset called BATTING in the WORK library. A two-level, permanent dataset could have been created.

CREATE VIEW will create a description of how to access the data. This view can then be used as any SAS dataset would be used, in any procedure or data step. Each time it is

referenced the connection to ODBC is re-established and the current data from the database is accessed.

6. select *

This is the description of what is to be kept in the SAS dataset that is being created. In this case the asterisk (*) means "select everything" that's coming from the ODBC connection.

We could have specified field names here. If we did, they would be the same field names as in the database tables that we are accessing. If the names are longer than 8 characters, SAS will truncate them to 8. If there is redundancy at 8 characters, SAS will truncate at 7 and add a numeric extension to make the names unique. For example, suppose our Access database had fields named StolenBases and StolenBasesAttempted. Both of these are too long for SAS variable names so they will be truncated to 8 characters. However, the first 8 characters of both is STOLENBA, so SAS will create variables called STOLENB1 and STOLENB2. The original field names, for all fields, are stored in the SAS variable labels.

7. from connection to stats

The FROM keyword specifies where the source of the data. In this case, our ODBC connection which we called STATS. If we hadn't used an alias, we would code:

```
from connection to odbc;
```

8. (select * from BatterStats);

The SQL statements inside the parentheses are going to be sent by the ODBC Manager to the Microsoft Access ODBC driver. In our example, we want everything from the table called BatterStats. Notice that the table name is longer than 8 characters and that we did not truncate it. That is because SAS does not evaluate the statements inside the parenthesis at all. This is called "SQL Pass-Through". The statements are "passed through" to the server application for processing.

SQL Pass-Through

This has implications for the setup of our queries. Suppose that we just wanted a dataset that contained the players names (PlayerName) and batting averages (BattingAverage). The following two queries would create identical datasets:

```
select PlayerNa,BattingA
from connection to stats
(select *
from BatterStats);
```

```
select *
from connection to stats
(select PlayerName,BattingAverage
from BatterStats);
```

Let's look at the difference between the two. In the first query we're telling ODBC to tell Microsoft Access to send the entire BatterStats table across our connection and SAS will select the two fields we want to keep (notice we truncated the field names in the code).

In the second query, we're telling ODBC to tell Access to look in the table BatterStats and only send the fields PlayerName and BattingAverage (notice the real field names). We're telling SAS to keep everything (*) that is being sent.

We get much less data traffic if we let the server application do the data subsetting for us. We can also improve efficiency if we let the server application do any subsetting of records. Suppose we wanted the names and averages of all the players who are hitting over .350 - these are the guys we really want! Again, the following queries will produce identical results:

```
select PlayerNa,BattingA
from connection to stats
(select *
from BatterStats)
where BattingA gt .350;
```

```
select *
from connection to stats
(select PlayerName,BattingAverage
from BatterStats
where BattingAverage > .350);
```

In the first query not only are all the fields being passed to SAS, but all the records as well. SAS decides which to keep, based on the value of BattingA.

In the second query, Microsoft Access passes only the two fields we've requested and only those records which meet the batting average criteria. Not only is a "shorter", "narrower" table passed to SAS but Access does all the work.

These are not always considerations. If the database tables are small or the subsetting is minimal, you probably won't notice a difference. If, however, the tables are large and network traffic and processing time are an issue it pays to be mindful of where the pieces of your query are being processed.

9. disconnect from stats;

This terminates the connection to the ODBC data source. There is an implied disconnect when PROC SQL is terminated.

Hopefully this tutorial has given you enough information to try some of the capabilities of the SAS System and ODBC. Together they can offer a tremendous amount of flexibility to your applications.

The author may be contacted at:

WA State Office of Financial Management
PO Box 43113
Olympia, WA 98504-3113
(360) 586-0707 voice
(360) 664-8941 fax
peterl@ofm.wa.gov

References

SAS Institute Inc., SAS Technical Report P-262, *SAS/ACCESS Interface to ODBC: SQL Procedure Pass-Through Facility*, Release 6.08, Cary, NC: SAS Institute Inc., 1993.

SAS Institute Inc., *SAS ODBC Driver Technical Report: User's Guide and Programmer's Reference*, Release 6.11, Cary, NC: SAS Institute Inc., 1995.

SAS Institute Inc., *Installation Instructions for the SAS System Under Microsoft Windows*, Release 6.11, Cary, NC: SAS Institute Inc., 1995.

Riba, S. David and Elisabeth A. Riba, *ODBC: Windows to the Outside World*, Proceedings of the 21st Annual International Conference, Cary, NC: SAS Institute Inc., 1996.

Boozar, Forrest, *Configuring and Using ODBC with SAS/ACCESS Software*, Proceedings of the

21st Annual International Conference, Cary, NC: SAS Institute Inc., 1996.

Trademarks

SAS and SAS/ACCESS Interface to ODBC are registered trademarks of SAS Institute Inc.

ODBC, Windows 95, Excel and Access are registered trademarks of Microsoft Inc.

Other brands and product names are registered trademarks and trademarks of their respective companies.