Paper 25-27

Accessing MICROSOFT EXCEL and MICROSOFT ACCESS Through the Use of a Simple Libname Statement

Kee Lee, Purdue University, West Lafayette, Indiana

ABSTRACT

With the help of the SAS Access®, ODBC, and OLEDB, using MICROSOFT EXCEL Spreadsheet, MICROSOFT ACCESS data table, ORACLE data table and many other types of data that are ODBC/OLEDB compliant has never been easier. Among other methods, you can use the LIBNAME statement to assign a libref to the external data. Once a libref is successfully assigned, you can use the data as you use the native SAS® data set in the DATA step and many SAS procedures. You can even use SAS/ASSIST® to access different kinds of external data by pointing and clicking. For people who need to integrate and distribute data to and from different types of external data, this feature certainly can alleviate a lot of pain. This paper focuses on access MICROSOFT EXCEL and MICROSOFT ACCESS data. Let's see how it works…

To do this, I use PC with NT or Window 2000, SAS/BASE product and SAS/ACCESS® for PCs. Proper ODBC drivers for different types of external data are installed. This is for SAS beginners and above.

INTRODUCTION

Most my clients are MICROSOFT EXCEL and MICROSOFT ACCESS users. Many of them are also ORACLE data users. Very often I have to create databases from the data sent from them in different types of data created from different kind of software. Furthermore, I need to send electronic data to them in a format they are familiar with. Luckily, with SAS, ODBC, and OLEDB, I can accomplish all in a simple SAS program. In this paper, I am going to concentrate on showing you how I access the MICROSOFT EXCEL file and MICROSOFT ACCESS file through the LIBNAME statement.

WHAT ARE ODBC AND OLEDB?

Open Database Connectivity (ODBC) is a programming interface that enables programs to access data in database management systems.

Microsoft OLE DB is an API (application programming interface) that provides access to data, which can be in many forms, including a database table, an email file, a text file, or other kind of file. SAS/ACCESS interface accesses data from these sources through the OLE DB data providers. You specify the data provider, data source, and other connection information in a SAS/ACCESS LIBNAME statement and the SQL Procedure.

WHAT IS THE DIFFERENCE BETWEEN USING ODBC AND USING OLEDB?

Using ODBC to define libref for MICROSOFT EXCEL data, you need to define a named range in the MICROSOFT EXCEL worksheet. (Find out how to create a named range in MICROSOFT spreadsheet, search for help in MICROSOFT EXCEL.) For both MICROSOFT ACCESS and MICROSOFT EXCEL, you need to define the data source (DSN) first before you assign a libref to the data. (Find out how to define ODBC data source, search for help in Windows.)

Using OLEDB, you can directly code the data source in the program or use the SAS/ACCESS OLE DB services 'prompt' to define it interactively. With OLEDB you don't need to define the named range for MICROSOFT EXCEL.

USING ODBC IN THE LIBNAME STATEMENT TO ACCESS MICROSOFT EXCEL DATA

Let's assume I have a MICROSOFT EXCEL file named 'demo.xls' and it has a worksheet 'sheet1' with a named range called 'sheet1'. Its ODBC data source name (DSN) is 'odbcxls'.

Here is how to assign the libref 'odbcxls' to the data source odbcxls:
(Of course, libref can be any name you prefer to use.)

```
libname odbcxls odbc dsn=odbcxls;
```

Use the libref odbcxls in the DATA step:

```
data fromxls;
/*SAS data set  stores the extracted xls data*/
set odbcxls.sheet1;
/*points to DSN odbcxls which links to demo.xls*/
run;
```

You can write the SAS data back to the MICROSOFT EXCEL file and create a new worksheet.

```
data xlsodbc.sheet2;
/*sheet2 is the new sheet in demo.xls;*/
set fromxls;
newfield='anything'; *add a new field to it if you want to;
run;
```

Use the libref odbcxls in PROC PRINT :

```
proc print data=xlsodbc.sheet1 uniform;
run;
```

You can set options and add the WHERE statement to limit the data as you do when using the SAS data set.

USING ODBC IN THE LIBNAME STATEMENT TO ACCESS MICROSOFT ACCESS DATA TABLE

Again, assume an ODBC data source (DSN) 'odbcmdb' is defined to point to the file demo.mdb.
Assign a libref to the data source:

```
libname  odbcmdb dsn=odbcmdb;
```

Use the libref in DATA step:

```
data frommdb;
set odbcmdb.fromsas ;
```

```
/*fromsas is the data table in the demo.mdb file*/
run;
```

Use the libref in PROC PRINT

```
proc print data=odbcmdb.fromsas uniform;
run;
```

Write the SAS data to MICROSOFT ACCESS file and create a new table:

```
data odbcmdb.table2;  *create new table in demo.mdb;
set frommdb;
run;
```

USING OLEDB IN THE LIBNAME STATEMENT TO ACCESS MICROSOFT EXCEL DATA

There are two ways to use LIBNAME to associate a data source with OLEDB. If you don't know what options you need to put in the statement, you can use the SAS/ACCESS OLEDB services by simply type:

```
libname oledbxls oledb;
```

Once you submit the statement, the system will prompt you to fill in the data source name and other options.

The other way is to directly connect to the data source by typing the statement with the required parameters as following example:

```
libname oledbxls oledb provider="Microsoft.Jet.OLEDB.4.0"
properties=("data source"='c:\demo\demo.xls')
provider_string=" Excel 8.0";
```

Once the libref is assigned, you can use it in DATA step and PROC procedures. Here are some of the examples:

```
data fromxls;
set oledbxls.'sheet1$'n;

proc print data=oledbxls.'sheet1$'n;

proc sql;
Select school from oledbxls.'sheet1$'n ;
run;
```

Write SAS data to a new sheet in the file 'demo.xls':

```
data oledbxls.sheet3;
set fromxls;
libname oledbxls clear;
```

The convenience of using OLEDB to access MICROSOFT EXCEL data is that you don't need to define data source and create a named range in the worksheet. The catch is that you have to remember to use 'sheet1$'n with the libref instead of using 'sheet1'. OLEDB assigns the '$' to the worksheet name, and you put the 'n' to resolve the quoted name.

USING OLEDB IN THE LIBNAME STATEMENT TO ACCESS MICROSOFT ACCESS DATA

When you define libref using OLEDB for MICROSOFT ACCESS data, you don't need to provide 'provider string'. If you want to, you can use the prompt to furnish all the parameters or you can add the following in your code:

```
libname olemdb oledb
provider="Microsoft.Jet.OLEDB.4.0"
properties=('data source'='c:\demo\demo.mdb');
```

Extract data to SAS data set in the DATA step.

```
data frommdb;*get MSACCESS data into SAS data set;
set oledbmdb.fromsas ;
/*fromsas is the name of the data table in demo.mdb;*/
run;
```

Write and create a new data table in demo.mdb.

```
data oledmdb.table3; *create a new table in demo.mdb;
set frommdb; *from the SAS data set;
run;
```

**note: the OLEDB engine will not support the replace;

Of course, you can use the MICROSOFT ACCESS data in other procedures as use the MICROSOFT EXCEL.

CONCLUSION

I cannot show you all the syntax of using ODBC and OLEDB in the LIBNAME statement to define libref for different ODBC/OLEDB compliant data such as ORACLE, MICROSOFT FOX PRO and others. I hope the above examples give you some idea of the capabilities this feature can offer. For those who need to integrate data from different sources and distribute data to different formats, this kind of connectivity is seamless and painless. For those who want to try it on other types of data, it is certainly fun and worthy to explore.

REFERENCES

Window 2000 Help: Using Data Sources (ODBC)
SAS System Help:OLE DB Chapter, First Edition

TRADEMARK

The author may be contacted at :
Kee Lee
Associate Registrar for Internal Operations
Purdue University
Office of the Registrar
1095 Hovde Hall
West Lafayette IN 47907
(765) 494-7228
Fax: (765) 494-0570
E-mail: cklee@purdue.edu