Connect to and query Microsoft SQL Server using SAS/ACCESS Interface to ODBC

fortuneWalla
August 16th, 2015

Created on Sunday, May 18th, 2014 at 3:24 am

One of the important skills in SAS is being able to connect & query a local or remote database, conduct data analysis in SAS & write the new information back to the database. SAS provides access to a variety of databases as well as different ways of doing so. In this article, access to Microsoft SQL Server (MSSQL) using SAS/ACCESS ODBC interface is explained. If you are using SAS at an institute or organization, your Systems Administrator would have setup everything. The steps in this article might vary depending on the software configuration.

Introduction

SAS methods for accessing relational database data [1]

> SAS/ACCESS Interface to Relational Databases is a family of interfacesâ€“each licensed separatelyâ€“with which you can interact with data in other vendor databases from within SAS. SAS/ACCESS provides these methods for accessing relational DBMS data.

- You can use the **LIBNAME** statement to assign SAS librefs to DBMS objects such as schemas and databases. After you associate a database with a libref, you can use a SAS two-level name to specify any table or view in the database. You can then work with the table or view as you would with a SAS data set.
- You can use the **SQL pass-through** facility to interact with a data source using its native SQL syntax without leaving your SAS session. SQL statements are passed directly to the data source for processing.
- You can use **ACCESS** and **DBLOAD** procedures for indirect access to DBMS data. Although SAS still supports these procedures for database systems and environments on which they were available for SAS 6, they are no longer the recommended method for accessing DBMS data.

Not all SAS/ACCESS interfaces support all of these features. To determine which features are available in your environment, see SAS documentation.

SAS/ACCESS interfaces to connect with MSSQL. [1]
SAS provides many ways to establish a connection to a MSSQL database depending on your site license. Some of them are:

- SAS/ACCESS Interface to **Microsoft SQL Server:**
  This has been tested and certified against Data Direct Technologies Connect ODBC and Data Direct SequeLink ODBC products.
- SAS/ACCESS Interface to **ODBC**, including ParAccel, Microsoft Parallel Data Warehouse, and more:
  Open database connectivity (ODBC) standards provide a common interface to a variety of data sources. The goal of ODBC is to enable access to data from any application, regardless of which DBMS handles the data.
- SAS/ACCESS Interface to **OLE DB:**
  Microsoft OLE DB is an application programming interface (API) that provides access to data that can be in a database table, an e-mail file, a text file, or another type of file. This SAS/ACCESS interface accesses data from these sources through OLE DB data providers such as Microsoft Access, Microsoft SQL Server, and Oracle.

Software components setup:

1) ODBC DSN Source Setup:
Please read the previous article about setting up MSSQL to accept remote connections. Make sure you setup the DSN to connect with SQL Server for your OS. Your institution/organization System Administrator should help you out with this. This is the procedure for Windows OS:
http://broadwin.com/Manual/EngMan/23.3.3_Create_an_ODBC_DSN_to_SQL_Server.htm

2) SAS/ACCESS Interfaces availability: Before starting make sure your SAS installation contains the SAS/ACCESS Interface tools to connect with external databases.
SOURCE CODE:

```
Proc setinit noalias;
Run;
```

OUTPUT: Partial output shown below.

```
NOTE: PROCEDURE SETINIT used (Total process time):
      real time           0.04 seconds
      cpu time            0.00 seconds

Product Expiration Dates:
---SAS/ACCESS Interface to Microsoft SQL Server
         14DEC2015
---SAS/ACCESS Interface to ODBC
         14DEC2015
---SAS/ACCESS Interface to OLE DB
         14DEC2015
---SAS/ACCESS Interface to Teradata
         14DEC2015
```

Ideally you should have Microsoft SQL Server, ODBC & OLE DB. However, having only ODBC is enough, since most databases support ODBC

SAS ODBC MSSQL connection examples:[2]

1) Setting up SAS ODBC MSSQL connection with prompt using LIBNAME statement.

This uses the prompt method. The `SYSDBMSG` variable is used to write the connection details to the SAS LOG.

SOURCE CODE:

```
libname sql odbc prompt;
%put %superq(sysdbmsg);
```

a) You are prompted to select a DSN source.
b) You are prompted to enter the username & password of the remote MSSQL connection.
c) If the connection is successful, the following message is displayed.

OUTPUT:

SAS LOG:

```
10   libname sql odbc prompt;
NOTE: Libref SQL was successfully assigned as follows:
      Engine:        ODBC
      Physical Name: dbpcsql
11   %put %superq(sysdbmsg);
ODBC:
DSN=dbpcsql;Description=dbpcsql;UID=userName;PWD=passWord;APP=SAS
9.2 for
Windows;WSID=DBPC;Network=DBMSSOCN
```

2) Setting up SAS ODBC MSSQL connection to *master* database using LIBNAME statement:

SOURCE CODE:

```
LIBNAME mssql odbc user='userName' password='passWord'
datasrc=dbpcsql schema=dbo;

data new;
set mssql.spt_monitor;
run;

proc print data=mssql.spt_monitor;run;
```

OUTPUT:

SAS LOG:

```
1    LIBNAME mssql odbc user='userName' password=XXXXXX
datasrc=dbpcsql schema=dbo;
```

```
NOTE: Libref MSSQL was successfully assigned as follows:
      Engine:        ODBC
      Physical Name: dbpcsql
2    data new;
3    set mssql.spt_monitor;
4    run;

NOTE: There were 1 observations read from the data set
MSSQL.spt_monitor.
NOTE: The data set WORK.NEW has 1 observations and 11 variables.
NOTE: DATA statement used (Total process time):
      real time           0.34 seconds
      cpu time            0.03 seconds


5    proc print data=mssql.spt_monitor;run;
NOTE: Writing HTML Body file: sashtml.htm

NOTE: PROCEDURE PRINT used (Total process time):
      real time           1.28 seconds
      cpu time            0.28 seconds
```

Results Viewer:sashtml

The SAS System

Obs	lastrun	cpu_busy	io_busy	idle	pack_received	pack_sent	connec
1	02APR2010:17:34:58.817	9	7	792	28	28	14

3) Setting up SAS ODBC MSSQL connection to *AdventureWorksDW* database using LIBNAME statement:
SOURCE CODE:

```
LIBNAME mssqlaw odbc user='userName' password='passWord'
datasrc=dbpcsql qualifier=AdventureWorksDW2008R2 schema=dbo;

proc print data=mssqlaw.DimCustomer(obs=15);run;
```

OUTPUT

SAS LOG:

```
7    LIBNAME mssqlaw odbc user='userName' password=XXXXXX
datasrc=dbpcsql
7  ! qualifier=AdventureWorksDW2008R2 schema=dbo;
NOTE: Libref MSSQLAW was successfully assigned as follows:
      Engine:        ODBC
      Physical Name: dbpcsql
8
```

```
9    proc print data=mssqlaw.DimCustomer(obs=15);run;

NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.37 seconds
      cpu time            0.06 seconds
```

Results Viewer:sashtml

The SAS System

Obs	CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Su
1	11000	26	AW00011000		Jon	V	Yang	0	1970-04-08	M	
2	11001	37	AW00011001		Eugene	L	Huang	0	1969-05-14	S	
3	11002	31	AW00011002		Ruben		Torres	0	1969-08-12	M	
4	11003	11	AW00011003		Christy		Zhu	0	1972-02-15	S	
5	11004	19	AW00011004		Elizabeth		Johnson	0	1972-08-08	S	
6	11005	22	AW00011005		Julio		Ruiz	0	1969-08-05	S	

Obs	CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Su
7	11006	8	AW00011006		Janet	G	Alvarez	0	1969-12-06	S	
8	11007	40	AW00011007		Marco		Mehta	0	1968-05-09	M	
9	11008	32	AW00011008		Rob		Verhoff	0	1968-07-07	S	
10	11009	25	AW00011009		Shannon	C	Carlson	0	1968-04-01	S	
11	11010	22	AW00011010		Jacquelyn	C	Suarez	0	1968-02-06	S	
12	11011	22	AW00011011		Curtis		Lu	0	1967-11-04	M	
13	11012	611	AW00011012		Lauren	M	Walker	0	1972-01-18	M	
14	11013	543	AW00011013		Ian	M	Jenkins	0	1972-08-06	M	

Obs	CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Su
15	11014	634	AW00011014		Sydney		Bennett	0	1972-05-09	S	

4) Setting up SAS ODBC MSSQL connection to *master* database using SQL pass-through method:
SOURCE CODE:

```
proc sql outobs=15;
connectto odbc as msql2 (user='userName' password='passWord'
datasrc=dbpcsql );
select * from connection to msql2 (select * from
master.dbo.spt_monitor);
disconnect from msql2;
quit;
```

OUTPUT:

SAS LOG:

```
17   proc sql outobs=15;
18      connect to odbc as msql2 (user='userName' password=XXXXXX
datasrc=dbpcsql );

19      select * from connection to msql2 (select * from
master.dbo.spt_monitor);
20      disconnect from msql2;
21   quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.04 seconds
      cpu time            0.01 seconds
```

Results Viewer:sashtml

The SAS System

lastrun	cpu_busy	io_busy	idle	pack_received	pack_sent	connections	pack_errors
02APR2010:17:34:58.817	9	7	792	28	28	14	0

5) Setting up SAS ODBC MSSQL connection to *AdventureWorksDW* database using SQL pass-through method:
SOURCE CODE:

```
proc sql outobs=15;
```

```
connectto odbc as msql2 (user='userName' password='passWord'
datasrc=dbpcsql );
select * from connection to msql2 (select * from
AdventureWorksDW2008R2.dbo.DimCustomer);
disconnect from msql2;
quit;
```

OUTPUT:

SAS LOG:

```
22   proc sql outobs=15;
23      connect to odbc as msql3 (user='userName' password=XXXXXX
datasrc=dbpcsql );

24      select * from connection to msql3 (select * from
24 ! AdventureWorksDW2008R2.dbo.DimCustomer);
WARNING: Statement terminated early due to OUTOBS=15 option.
25      disconnect from msql3;
26   quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.35 seconds
      cpu time            0.03 seconds
```

Results Viewer:sashtml

The SAS System

CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Suffix
11000	26	AW00011000		Jon	V	Yang	0	1970-04-08	M	
11001	37	AW00011001		Eugene	L	Huang	0	1969-05-14	S	
11002	31	AW00011002		Ruben		Torres	0	1969-08-12	M	
11003	11	AW00011003		Christy		Zhu	0	1972-02-15	S	
11004	19	AW00011004		Elizabeth		Johnson	0	1972-08-08	S	
11005	22	AW00011005		Julio		Ruiz	0	1969-08-05	S	
11006	8	AW00011006		Janet	G	Alvarez	0	1969-12-06	S	
11007	40	AW00011007		Marco		Mehta	0	1968-05-09	M	

CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Suffix
11008	32	AW00011008		Rob		Verhoff	0	1968-07-07	S	
11009	25	AW00011009		Shannon	C	Carlson	0	1968-04-01	S	
11010	22	AW00011010		Jacquelyn	C	Suarez	0	1968-02-06	S	
11011	22	AW00011011		Curtis		Lu	0	1967-11-04	M	
11012	611	AW00011012		Lauren	M	Walker	0	1972-01-18	M	
11013	543	AW00011013		Ian	M	Jenkins	0	1972-08-06	M	
11014	634	AW00011014		Sydney		Bennett	0	1972-05-09	S	

Summary:

The SAS/ACCESS interface component is used to connect to various kinds of databases. Three main ways of connecting SAS with MSSQL are using SAS/ACCESS interface for MS SQL Server, SAS/ACCESS interface for OLE DB & SAS/ACCESS interface for ODBC. SAS allows data retrieval/storage using both LIBNAME statement and direct SQL Pass-Through statement method.

What is shown above is very elementary code just to get the process started. Students learning data science/analytics should try to use different combinations of SAS methods & SAS interfaces to retrieve data from a database, manipulate it using SAS/SQL & write the processed data sets back to the database.

References:

1. SAS/ACCESS 9.2 for Relational Databases Reference, Fourth Edition. http://support.sas.com/documentation/cdl/en/acreldb/63647/HTML/default/viewer.htm#titlepage.htm
2. SAS 9.2 SQL Procedure User's Guide. http://support.sas.com/documentation/cdl/en/sqlproc/62086/HTML/default/viewer.htm#titlepage.htm