

A Process for Automatically Retrieving Database Using ODBC and SAS/ACCESS® SQL Procedure Pass-Through Facility

Leiming Li
Duke University Medical Center, Durham, NC

ABSTRACT

We know that with the release of Release 6.10 or later, The SAS® System can easily communicate with other applications, such as SAS/ACCESS® product can read and write the PC applications that are Open Database Connectivity (ODBC) compliant. But how can we retrieve all or specific tables in a database without repeated operations or tedious programming? This paper discusses how the use of a macro ODBC2SAS and ODBC administrator can achieve this task.

INTRODUCTION

In our research projects, we often need to efficiently retrieve different databases and analyze them in SAS system. This paper discusses one macro I developed for building our data management system. The macro ODBC2SAS uses SQL Procedure Pass-through facility to make the SAS/ACCESS® interface to ODBC and SQL Server and read in all or specific tables in the ODBC database automatically. Use of the macro ODBC2SAS enables our researchers to rapidly retrieve any information in the projects DOE_H and OHIO without tedious programming, such as typing hundreds of tables' names, and even without knowing anything about the database.

Before use of this macro, you need to be familiar with how the DBMS and SAS System work in your operating environment. The SAS/ACCESS® Interface to ODBC and the ODBC driver of the DBMS (here is MS Access provided by Microsoft) must be installed, and then the data source description file must be set up in ODBC administrator. This program has been working well in our SAS System 6.12 under Window95/NT4.0.

CODE OF THE MACRO

Here is the program.

```

/*****
Program 'ODBC2SAS.sas'
Created on Monday Aug 24 1998 19:51:28 PM
In directory G:\proj\sas00\sasmac\
By li000025 (Leiming Li)

```

Purpose:
To Import all or specific tables in a non-SAS database through SAS/ACCESS® interfaces to ODBC automatically.

Parameters:

- Tables -- the list of table names you want to retrieve or blank (default) which lets computer grasp all tables for you.
- First --- the option to print the tables' contents and sample records if you use "FIRST=T" instead of the default "FIRST=F", which is very useful when you want to know the database without any prior information.
- dsn ---- data-source-name, which specifies the ODBC data source to which you want to connect. Data source must be configured. DSN= indicates that the connection is attempted using the ODBC API SQL Connect, which requires a data source name. Or you can either specify one of the following arguments: PROMPT, NOPROMPT, COMPLETE or REQUIRED.
- Uid ---- DBMS-usr-id, which specifies the DBMS usr ID. The uid= argument can be used only in conjunction with the dsn = argument. It's optional.
- Pwd --- DBMS-password, which specifies the DBMS password. The pwd = argument can be used only in conjunction with the dsn = argument. It's optional.

Note:

The &dsn, &uid and &pwd are case sensitive.

*****/

```

%macro odbc2sas( tables = ,
                dsn = ,
                uid = ,
                pwd = ,
                first = F );
%if &tables ne %then
%do;
  %* Get the number of the tables and individual table names;
  %let nt = 1;
  %do %until (%scan(&tables, &nt) = );
    %let _o&nt = %upcase(%scan(&tables, &nt, %str( )));
    %if %length(&&_o&nt) > 8 %then
      %let _s&nt = %substr(&&_o&nt, 1, 8);
    %else %let _s&nt = &&_o&nt;
    %let nt = %eval(&nt + 1);
  %end;
  %let nt = %eval(&nt - 1);
%end;
%else
%do;
  %* Get all table names from a special ODBC Query if no
  specified table names;
  proc sql;
    connect to ODBC as mydb (dsn = &dsn uid = &uid
                             pwd = &pwd);
    create view list
    as select * from connection to mydb
              (ODBC::SQLTables, , , "TABLE")
    order by table_na ;

```

```

disconnect from mydb;
%put &SQLXRC &SQLXMSG;
quit;

%* Get the number of the tables and individual table names in the
view LIST;
%* Make non repeated first 8-character table names (no need if
use version 7 + of SAS);
data _null_;
set list end=eof;
by table_na;
retain rep cunt;
table_na=trim(left(table_na));
if _n_=1 then
do;
rep=substr(table_na,1,min(8,length(table_na)));
cunt=0;
end;
else
do;
if substr(table_na,1,min(8,length(table_na))) = rep then cunt+1;
else do;
rep=substr(table_na,1,min(8,length(table_na)));
cunt=0;
end;
end;
if cunt=0 then s=rep;
else if 0 < cunt <10 then s=substr(rep,1,(length(rep)-1))||left(cunt);
else if 9 < cunt <100 then s=substr(rep,1,(length(rep)-2))||left(cunt);
else;
call symput('_o'||left(_n_), trim(table_na));
call symput('_s'||left(_n_), trim(s));
if eof then call symput('nt', _n_);
run;
%end;

```

```

%* Get the table from ODBC for each one;
proc sql;
connect to ODBC (dsn=&dsn uid=&uid pwd=&pwd);
%do i=1 %to &nt;
create table &&_s&i
as select * from connection to ODBC
(select * from &&_o&i);
%end;
disconnect from ODBC;
quit;

```

```

%if %upcase(&first) eq T %then
%do;
footnote "Generated by macro ODBC2SAS, %sysfunc(time(), time.)
%sysfunc(date(),worddate).";
%do i=1 %to &nt;
title3 "Check table &&_o&i in project &dsn";
proc contents data = &&_s&i;
proc print data = &&_s&i(obs=10);
run;
title3;
%end;
footnote;
%end;

```

```

%if &syserr>0 %then %goto err;
%else %goto out;
%err: %put ERROR: The macro(ODBC2SAS) discontinued

```

```

processing;
%out:
%mend odbc2sas;

```

EXAMPLE

Here is an example for use of ODBC and the macro ODBC2SAS. For the project DOE_H, its database is an Access97 database with 150 tables. How can I let researchers easily access it within SAS system? The best way in our work environment is to use this macro taking the advantages of ODBC and SAS/ACCESS® SQL Procedure Pass – Through Facility, even though DDE (Dynamic Data Exchange) and OLE (Object Linking and Embedding) can communicate well with PC-based applications. So after installing the SAS/ACCESS interface to ODBC and Microsoft Access ODBC Drive on our computers, I created the data source name (DSN) as DOE_H. The way is: click the ODBC Administrator in the Windows Control Panel, click button “ADD”, selecting the drive “Microsoft Access Driver(*.mdb)” matching your database, then click button “finish”. On next window, define the name of the data source and database. If the database have password and use group, then you need click on button “Advanced” to set up the same “Password” and “Login name” as the database’s. The following figures show how to set up the DSN.

Figure 1 Find Control Panel

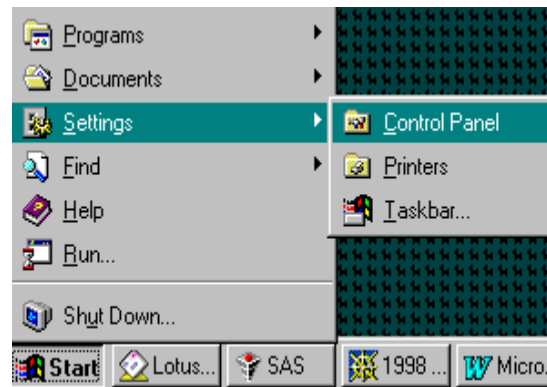


Figure 2 Find ODBC administrator

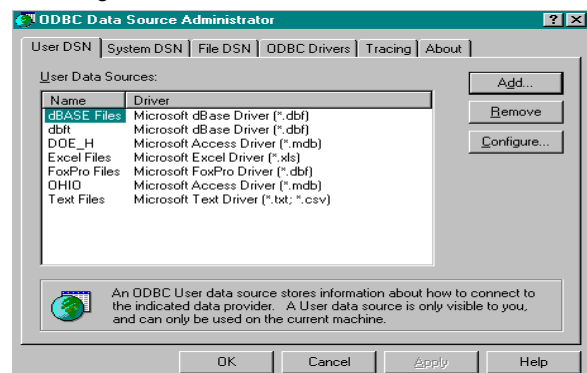


Figure 3 Open ODBC Administrator

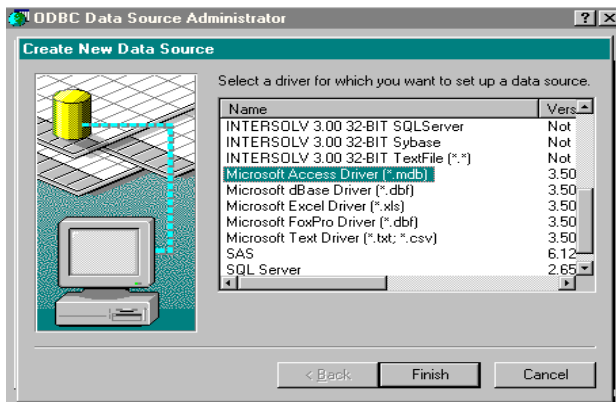


Figure 4 Select the ODBC drive

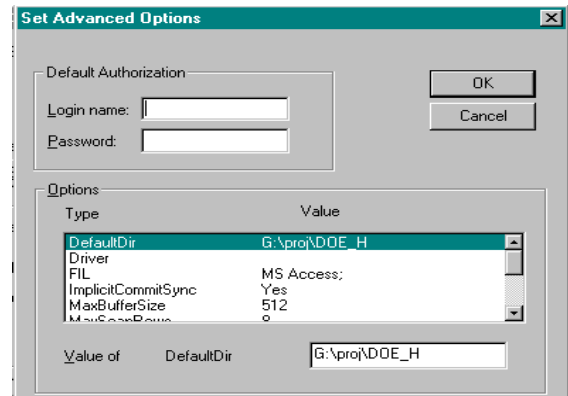


Figure 7 Set Pwd & Uid (Optional)

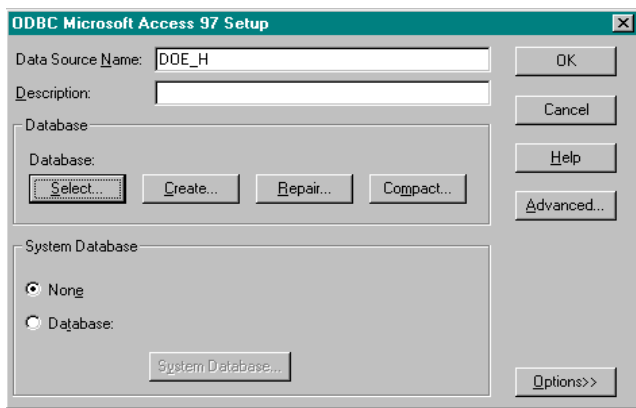


Figure 5 Define the DSN

Then I may use them as the following:

```
%include "g:\proj\sas00\standard.sas";
```

```
* Read in all tables in DOE_H and print their contents and
* sample records;
%odbc2sas(tables=,
          dsn=DOE_H,
          uid=DOEProgs,
          pwd=XXXX,
          first=T)
```

```
* To retrieve the tables workerhistory, trades and tasks
* In DOE_H project database, not print their contents and
* sample records. ;
%odbc2sas(tables=workerhistory trades tasks,
          dsn=DOE_H,
          uid=,
          pwd=XXXX,
          first=F)
```

SUMMARY

This process makes our researchers and data managers to effectively access different PC databases within SAS system. The same idea can be used in other database interface systems, such as SAS/ACCESS® interfaces to DB2 for UNIX, Informix, ORACLE, SYBASE, and SQL Server; these relational database management systems can run under various operating environments and releases of the SAS System. Finally, we can integrate all these macros as part of our data warehouse.

REFERENCES

(1) SAS Institute Inc. (1990), SAS/ACCESS Software Changes and Enhancements: SQL

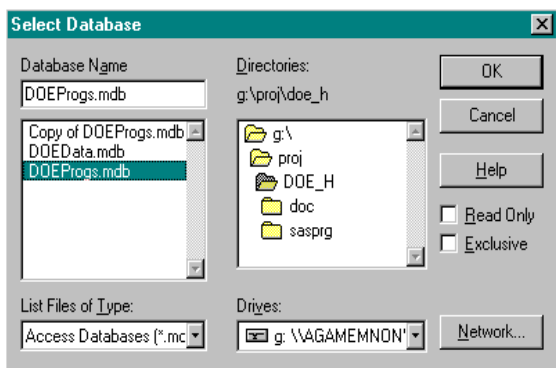


Figure 6 Select Data Source

Procedure Pass-Through Facility, Version 6, Cary, NC: SAS Institute Inc.

(2) SAS Institute Inc. (1990), SAS Guide to Macro Processing, Version 6, Second Edition, Cary, NC: SAS Institute Inc.

(3) <http://www.sas.com/techsup/> (TS329)

SAS, SAS/ACCESS® are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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

ACKNOWLEDGMENTS

Thanks to Dr. John M. Dement and Dr. Hester Lipscomb at Duke University Medical Center.

CONTACT INFORMATION

Leiming Li
Duke University Medical Center
2200 W. Main St., Site 600A
First Union Building
Durham, NC 27705
Phone: (O) 919-286-1722 X 281,
(H) 919-468-3661
E-mail: Leiming.Li@duke.edu