

Paper 77-28

Automatically Combining the Data from a Variety of DBMSs with ODBC and PROC SQL

Han Wu, Jason Au, Craig Toyota and Yan Xu[†]
TB Control Program, Department of Health services, Los Angeles County

Key words: ODBC, PROC SQL, DBMS

Abstraction

Many dynamical data analyses and reports need to be submitted periodically in health report systems. Very often the reports are generated from the multiple data that are stored in different DBMS or spreadsheets. Using ODBC technology and SAS/ACCESS software can pull data directly into SAS from different data sources. In the given example, first, four ODBC drivers associated with Micro SQL, Paradox, Micro Access, and Micro Excel files were installed in ODBC Data Source Administrator on Microsoft Window. Next, for each of the ODBC drivers, four correspondent data sources were set up. Then, a SAS MACRO was introduced how to use ODBC and PROC SQL sending statements to the DBMS for execution. In the PROC SQL Pass-Through, 'CONNECT' command establishes the connection with the ODBC driver; 'SELECT' expression passes queries to the source files; 'CREATE' command creates a required SAS subset data file based on the 'WHERE' clause. Finally, one run generated the requested results from the raw data stored in different DBMS by the MACRO.

Introduction

Tuberculosis (TB) Control Program of Los Angeles County has registered the surveillance data from TB cases, suspects, contacts, and other populations at high risk infecting TB in Los Angeles County. The data was stored in DOS-based Paradox application and is being replaced by a web-based application written by using Microsoft

Visual InterDev and Active Server Pages. Also some data from special projects, such as TB case hospitalization, drug resistant TB cases, and etc. are stored in Micro Excel, Micro Access or dBase files. Many dynamical data analyses and reports need to combine the data from different sources. Therefore, automatically and directly pulling data from different database management systems (DBMS) or spreadsheets into SAS will be very handy and efficient. Traditionally, for the data from relational databases that have a table structure of columns and rows, using import function or program of SAS can import the data into SAS directly. For non-relational databases, using ODBC (Open Database Connectivity) technology and SAS/ACCESS software can pull the data into SAS from different DBMS or spreadsheets.

In this paper, we gave an example written by a SAS macro to explain how to use ODBC and PROC SQL to send statements to the DBMS for execution. In the example, A SAS program will generate a list of TB patients, who were admitted into public hospital, with the information about their diagnosis date and the susceptibility test result of INH from four different kinds of databases: the database 'TB_all' in Micro SQL has the variable of TB diagnosis date (date_dx) after the year 2002; the database 'TB_old' in DOS-based Paradox has the variable of TB diagnosis date (date_dx) before the year 2002; the database 'hos_all' in Micro Access has a variable (type_hos) that indicates who were admitted into a public or private hospital; the database 'lab_all' in Micro Excel has the variable (INH) that shows if the patient was sensitive (S) or

resistant (R) to INH. All four databases have a common variable 'id' that is the ID number for each patient.

Program

```
%macro SQLPASS (dsn, new_file, var,
                 original_file, where, num); Ø
proc sql noprint;
connect to odbc (dsn=&dsn); Û
create table &new_file as Û
select * from connection to odbc Û
(select id &var from "&original_file" Û
&where); Ý
disconnect from odbc;
quit;

data new&num; ß
set &new_file;
proc sort;
by id;
run;
%mend;

%sqlpass(sql, registry, date_DX, TB_all,
where date_dx>'01jan2002'd, 1); Þ
run;
%sqlpass(paradox, archive, date_DX,
TB_old, where date_dx>'01jan2001'd,
2); Þ
run;
%sqlpass(access, hospital, type_hos,
hos_all, where type_hos='Public', 3); Þ
run;
%sqlpass(excel, resist, INH, sheet1$, where
inh='R', 4); Þ
run;

data final; à
merge new1 new2 new3 new4;
by id;
proc print data=final;
title 'Hospitals and INH susceptibility test of
TB cases diagnosed in 2001 - 2002';
run;
```

Result

Hospitals and INH susceptibility test of TB cases diagnosed in 2001 - 2002

id	Date_DX	Type_hos	INH
01	02/13/02	Public	R
02	06/28/02	Public	R
03	10/17/01	Public	R

Discussion

First, four interface engine ODBC drivers, associated with Micro SQL, the DOS-based Paradox, Micro Access, and Micro Access files need to be installed in ODBC Data Source Administrator on Microsoft Window. Next, for each of the ODBC drivers, four data sources named 'sql', 'archive', 'access' and 'excel' need to be set up in the Data Source window. Thus, the ODBC manger uses the information in the four data sources to communicate with the ODBC database drivers and retrieve the requested data. The detail steps were demonstrated in Processing Database and Spreadsheet Data with SAS/ACCESS software Course Notes.¹ Then, the MACRO SQLPASS Ø uses ODBC and PROC SQL to send statements to the DBMS for retrieving the requested data from 'TB_all', 'TB_old', 'Hos_all' and 'lab_all'.

To simplicity's sake, the data in Micro SQLPASS was used as an example to explain how data was pulled into SAS in the PROC SQL Pass-Through. First, 'connect to odbc (dsn=&dsn)' Û establishes the connection with the ODBC driver and 'dsn' provides the data source name that has been previously set up as 'sql' in ODBC Data Source Administrator on Microsoft Window; next, 'select * from connection to odbc' Û provides all variables to SQL query ('select id &var 'Û), that selects the variable id and &var (date_dx) from '&original_file' (TB_all) Û into SAS; then, 'create table &new_file as' Û creates a SAS temporary

data file &new_file ('registry') and '&Where' command Ÿ subsets the data based on the condition of 'where date_dx>'o1jan2002'd' in MACRO call P; finally, 'data new&num' B creates the data new&num (new1), that contains all TB cases were diagnosed in the year 2002.

In same procedures, the second new SAS files new2, that contains all TB cases were diagnosed in the year 2001 in the Paradox file 'TB_old', was created based on the condition of 'where date_dx>'o1jan2001'd' in MACRO call P. The third macro call P created SAS files new3 containing all TB cases who was hospitalized in a public hospital from the Micro ACCESS file 'hos_all', based on the condition of 'where type_hos='Public''. The fourth macro call P created the file new4 containing all TB cases who resisted to drug INH from the Micro Excel file 'lab_all', based on the condition of 'where inh='R''. 'Data final' à prints the results after merging data 'new1', 'new2', 'new3' and 'new4'. Therefore, one run generated the requested result from the raw data stored in different DBMS by the SAS program.

Conclusion

Using ODBC technology and PRO SQL Pass-Through of SAS/ACCESS software can automatically and directly pull data from different DBMS or spreadsheets into SAS. By combining the SAS program skills introduced previously²⁻⁴, one run can automatically generate a pre-designed summary table from the raw data in multiple DBMSs. This method has advantages of more reliability, consistency and efficiency and should be used extensively in reporting systems. The tips in the program can save your time and make your life easier!

References

1. Alisan Boes, et a (1998). *Processing Database and Spreadsheet Data with SAS/ACCESS® software Course Notes*. Book code 58235, prepared 19Aug98, 195-197. SAS Institute Inc.
2. Han Wu, et al (2002). "Using Automated Reports to Dynamically Monitor TB Control Performance." *Proceedings of the National TB Controllers Workshops Association 2002*. Alexandria, VJ, USA.
3. Han Wu (2002). "Automatically Creating Pre-designed Report Tables to Monitor Tuberculosis Control Projects by a SAS Program." *Proceedings of the 7th Annual Conference of International Union Against TB and Lung Disease 2002*. Vancouver, BC, Canada.
4. Han Wu & Ying Chen (2001). "Automatically Creating a Summary Table from Multiple Outcomes by a SAS Program." *Proceedings of the Ninth Conference of Western User of SAS Software 2001*, San Francisco, CA, USA.

Contact information

HanWu, MD and MPH
Epidemiologist
Chief, Epidemiology Services
Tuberculosis Control Program
Department of Health Service
Los Angeles County

2615 South Grand Ave.
Los Angeles, CA 90007, USA
Tel: (213) 744-6243
Fax: (213) 749-0926
Email: hwu@dhs.co.la.ca.us

†: Optic Engineer & Data Analyst
Department Of Science & Technology
Corning Inc
Fountain Valley, CA 92708, USA
Email: xuy@corning.com