Accessing non-SAS® Data Files on Remote Servers via ODBC

Vecdet Mehmet-Ali, Statistics Canada, Ottawa, Ontario

Accessing data from various remote databases of different architectures with a SAS application is much easier than the fragmented documentation suggests. This paper describes tips and techniques used in a production application which produces network traffic statistics with data from a network monitor (HP Open View), a cable management database (dBase III), a hardware inventory database (Paradox) and a financial database (Paradox). The data navigation and retrieval uses Open Data Base Connectivity (ODBC) managed from a SAS program. The retrieval performance of SAS 6.10 under Windows 3.1, SAS 6.10 and SAS 6.11 under NT running on 486-66 cpu with 16 Meg, SAS 6.10 and SAS 6.11 under NT running on 586-100 cpu with 32 Meg are compared.

Introduction

The focus of this paper is the portion of an application which retrieves and merges data from cable management, hardware inventory and financial databases. The first section describes what is needed in the Windows environment to connect to the remote databases with ODBC (Open Data Base Connectivity). The second section describes the SAS code and options needed to retrieve data through the linkage. The third section describes facilities to automate the daily processing and the security. The fourth section describes the transport of data to the mainframe where it is merged with network monitor data and the last section compares the performance under different SAS versions, cpu and operating systems.

Setting up the Windows environment

The first step is to acquire ODBC. This may seem obvious, but ODBC is not included with either Windows or versions of the SAS System prior to 6.10. It is supplied by Microsoft and is distributed with a number of Microsoft and other vendor products for no extra charge. For example, after installing SAS 6.10 or SAS 6.11 on your workstation, you can run setup.exe from sasiodbc\setupw32 or from sasiodbc\setupw16 to install the SAS Driver. This will add an ODBC icon in the Windows Control Panel.

Select this icon to install ODBC drivers(16 or 32 bit) and to configure data sources. If you are running Windows 3.1 the 16-bit ODBC drivers are sufficient, but for any version of NT you need 32-bit ODBC drivers. You can acquire these from Microsoft or other software vendors. SAS driver is provided by SAS Institute when you purchase SAS 6.10 or SAS 6.11.
• The traffic runs LIFCO and executes the reading inventory data module.
• Using Dynamic Data Exchange, DDE, LIFCO transfers all the information from Excel to the SAS system. After a validation process with multiple data sets, the information of the spreadsheets is transferred to the Novell server, into a temporary directory.
• After the batch updating process, the central database is ready with the information.

![Image](image.png)

Figure 10. Inventory Data Entry Process

The process used in the Flow module is the same as in the Inventory module. The only difference between the modules is that instead of sending an Excel file the agents send an ASCII file with the information.

The use of familiar spreadsheets and ASCII files for entering the data, and the DDE facility of the SAS system for Windows, made the data feeding to LIFCO real easy!!

REPORT GENERATION

Once the data is in the system, report generation and screen display facilities were crucial to the users. All the reports were written using SAS/AF and SAS/SCL combined with the SAS system. We were able to create a lot of customized reports that merged many data sets into one user-friendly report. Figures 11 to 13 illustrate different examples of the report generation feature.

GRAPHIC REPORTS

Linking SAS/AF, SAS/SCL and SAS/GRAPH, we were able to create specialized graphs for summarizing the information. This graphs (bar charts and line graphs) are particularly useful for analyzing the container inventory development through time. This was a desired feature by FMG managers. Figures 14 to 16 illustrate different examples of the graphic feature.
SAS code to access data via ODBC

Shown below are fragments of SAS code with SQL statements written to request information from the above database management systems. In addition to showing examples of accessing three different databases, the examples show many of the optional constructs used in SQL queries.

* Switch SAS log to a permanent file ;
DATA _null_;
call symput( "knov. put(time()), hhm2.2" );
call symput( "tday. put(today(), weekdate3. )" );
Filename logfile "\m\netacct\intrface\log\tday. &now. .bt";
PROC printto log=logfile new;

* Get technical information about premises wiring from ;
* dBase file Pwire ;
Libname netacct m:\netacct\intrface ;
PROC sql;
  connect to odbc (dsn=d\Base Files );
  create table netacct.cms as
  select *
    from connection to odbc
    (select * from pwire where cir_type not like 'SPARE%' and cir_type not like 'VOICE%');
  disconnect from odbc;
  quit;

* Process and manipulate received data according to the ;
* application ;
:
:

* Get finance related information from paradox file Finance ;
PROC sql;
  connect to odbc (dsn=Paradox Files );
  create table netacct.cmstx as
  select stcid, deviceid, devrate length=4 informat 4.
    format 4., nettype, hubport, netseg, jackid, remvdate
  informat mmddyy8. format date9.
  from connection to odbc
  (select "stcid/circuit#" as stcid, "monthly rate" as devrate, "removal date" as remvdate from Finance)
  full join
    netacct.cms
    on stcid = deviceid;
  delete from netacct.cmstx
    where remvdate = . and jackid = ' ';
  update netacct.cmstx
  set
    deviceid = ' ';
  alter table netacct.cmstx
  drop stcid;
  disconnect from odbc;
  quit;

* Get hardware information from paradox file Hardware ;
PROC sql;
  connect to odbc (dsn=Paradox Files );
  create table netacct.cmstxhhrd as
  select deviceid, hubport, netseg, nettype, jackid, devrate, devtype, trim(manufact)]) || trim(model)]) || trim(descript)] as devlocsc length=40.
  trim(bldnfor)]) || trim(roomsec) as devlocn
  length=12, substr(frc,1,3) as frc
  from connection to odbc
  (select "stcid/circuit#" as stcid, type as devtype.
  full join
    netacct.cmstxhhrd
    on stcid = deviceid;
  update netacct.cmstxhhrd
  set
    frc = 0 || substr(frc,1,3)
  where frc = ' ';
  disconnect from paradox;
  quit;

* Create a transport file ;
PROC sput data=netacct.cmstxhhrd
  file="m:\netacct\intrface\cmstxhhrd.txt";

* Change to default log and exit ;
PROC printto ;
Filename logfile clear;
Endsas;
Run;

Automating the daily process

The remote files are updated during the day by the application areas so our daily job is executed after working hours both to avoid conflict and to get the files reflecting all the changes for the day. As a bonus, running at night provides better performance as there is less traffic on the network (no access delays) and no conflict from other tasks.

At the time the application was written, our environment was SAS 6.10 running under Windows 3.11. We needed a job scheduler that would start SAS at a fixed time each day. For this we used a utility called KickOff which comes with WordPerfect 6.0. KickOff initiates jobs according to user defined events. We have continued to use KickOff under NT 3.5 because NT’s “at” command fails when a server is involved.

![KickOff utility](image)

Clicking on the <<Add>> or <<Edit>> buttons, opens the Edit / Add window.
The command to start SAS and execute the daily job is entered on the command line:

c:sas610\ws\sas.exe -autoexec
c:\netacct\sasini\nautoexec.sas -sasuser
c:sas610\ws\sasuser -work d:\saswork

The time is set to 6:30 p.m. and the repeat interval is set to 1 day to enable the job to run at 6:30 p.m. everyday.

The program code shown in the section “SAS code to access data via ODBC” is stored in ‘n:\netacctl\sasin\nada\’i’, where it will be referenced by the daily processing. The SAS autoexec file needed to include the daily job for execution is ‘nautoexe’:

* SAS Autoexec file ;
Filename sasin ‘\moc1\acctng\netacctl\sasin\’;
%include sasin\nada\i\sas);

Security Considerations
Leaving a workstation powered up and running overnight in an office environment has security problems of its own. There is however an easy way of getting around this problem. You can have a password protected screen saver. To do this select the Desktop icon in the Control Panel to see the Desktop window.

From Screen Saver list, select Marquee Display and click on the <<Setup>> button. This will open up a window to define a message text, font, color, etc. Select an appropriate text to ask co-workers not to power your workstation off. To lock the computer against unauthorized access, select the “Password protected” button in the Desktop window. Windows 3.1 will open a window to assign a password for the screen saver. Windows NT will assign your Logon password automatically.

Moving data to the MVS system
In the next phase of the application the transport file is transferred to the mainframe using FTP and then converted back to a SAS dataset. In most installations there will be several alternative ways to accomplish this function. We chose the FTP route because of very elaborate security procedures for accessing our mainframe. The side benefit is that it cleanly breaks the application into pieces such that we can rerun or replace components at will.

The job is submitted by a mainframe job scheduling system. The scheduler’s tools are used to generate the dataset name as CMmmdd where ‘mmdd’ is the month and the day of the data. The datasets are stored in a PDS called ‘CARS.DATANETyymmm.SAS6’ where ‘yyym’ is the year and the month of the data.

Here is a sample JCL and SAS code where step P1S1 executes FTP and step P1S2 executes SAS 6.08 on the mainframe:

//NA%DDD. JOB (0165, MN06, 2), USERID, CLASS=P //*
// Phase 1 Step 1 - Bring transport file to the mainframe
//P1S1 EXEC PGM=FTP, REGION=4M
//(SYSPRINT DD SYSOUT=*
//(OUTPUT DD SYSOUT=*
//(INPUT DD *
//mcc1
//userid
//password
//cd acctng/ netacctl/ intrface
//binary
//get cpport.text ‘carr.a.netacctl.cmsbhrd.text’ (REPLACE
//QUIT
//(Phase 1 Step 2 - Convert transport file to a SAS dataset
//(P
//(P1S2 EXEC SAS6
//(IMPRFILE DD DISP=SHR,
SAS Performance comparison

We replaced Windows 3.11 with Windows NT 3.5 in August, 1995. While 16-bit ODBC drivers worked fine with Windows 3.1, Windows NT 3.5 needed 32-bit ODBC drivers. Currently we have both sets of drivers installed on the workstation.

We installed the beta version of SAS 6.11 in August, 1995 and began running the same daily job in parallel. We were pleased to find the dramatic performance improvement provided by the migration to NT. We also found that the SQL procedures run slightly faster under SAS 6.11 than SAS 6.10 in the similar environment. The production version of SAS 6.11 was installed in December 1995, with no apparent differences from the beta version.

<table>
<thead>
<tr>
<th></th>
<th>Creating Table NETACCT.CMSTX with 8145 rows and 41 columns mm:ss</th>
<th>Modifying Table NETACCT.CMSTX with 7 columns mm:ss</th>
<th>Updating 8145 rows NETACCT.CMSTXHRD mm:ss</th>
</tr>
</thead>
<tbody>
<tr>
<td>486-66 cpu with 16M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAS 6.11 - Windows NT with 32-bit drivers</td>
<td>2:28</td>
<td>1:11</td>
<td>2:32</td>
</tr>
<tr>
<td>586-100 cpu with 32M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAS 6.10 - Windows NT with 32-bit drivers</td>
<td>0:56</td>
<td>0:46</td>
<td>1:25</td>
</tr>
<tr>
<td>SAS 6.11 - Windows NT with 32-bit drivers</td>
<td>0:54</td>
<td>0:44</td>
<td>1:16</td>
</tr>
</tbody>
</table>

Conclusion

When the SAS Institute’s support for Microsoft’s ODBC standard became a reality with version 6.10, the world of non-SAS data on remote servers became easily accessible.

References


Acknowledgements

The author wishes to thank Roy Hammond for his supervision and invaluable assistance.

SAS is a registered trademark of SAS Institute Inc., in the USA and other countries. @ indicates USA registration.

dBase is a registered trademark of Borland, Inc.

Paradox is a registered trademark of Borland, Inc.

Windows NT is a registered trademark of Microsoft, Inc.

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

Author

Vecdet Mehmet-Ali
Statistics Canada
Room-3117 Main Building, Tunney’s Pasture
Ottawa ON K1A 0T6
Phone: (613) 951-2390  - Fax: (613) 951-5198
E-mail: mehmvec@statcan.ca