

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

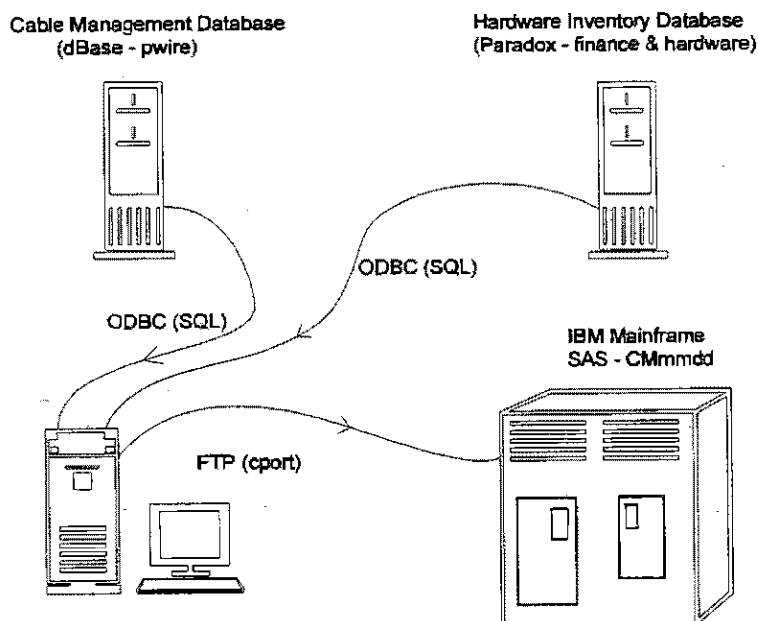
Vecdet Mehmet-Ali, Statistics Canada, Ottawa, Ontario

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

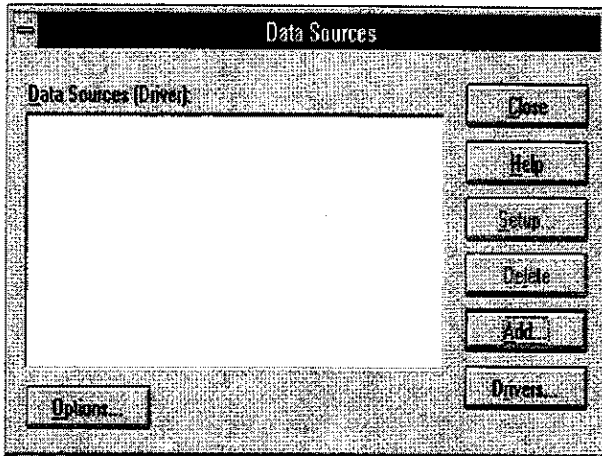
The first step is to acquire ODBC. This may seem obvious, but ODBC is not included with either Windows or versions of the SAS System prior to 6.10. It is supplied by Microsoft and is distributed with a number of Microsoft and other vendor products for no extra charge. For example, after installing SAS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

the SAS Driver. This will add an ODBC icon in the Windows Control Panel.

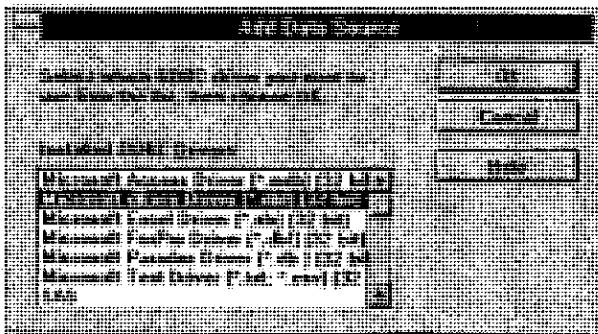
Select this icon to install ODBC drivers (16 or 32 bit) and to configure data sources. If you are running Windows 3.1 the 16-bit ODBC drivers are sufficient, but for any version of NT you need 32-bit ODBC drivers. You can acquire these from Microsoft or other software vendors. SAS driver is provided by SAS Institute when you purchase SAS 6.10 or SAS 6.11.

Defining ODBC Data Sources

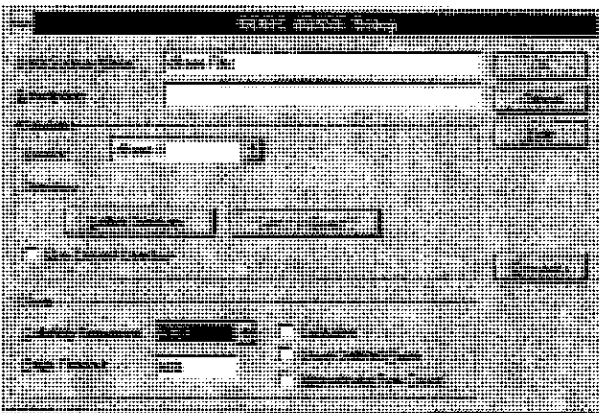
Select ODBC icon in Windows Control Panel. *Data Sources* window will be displayed:



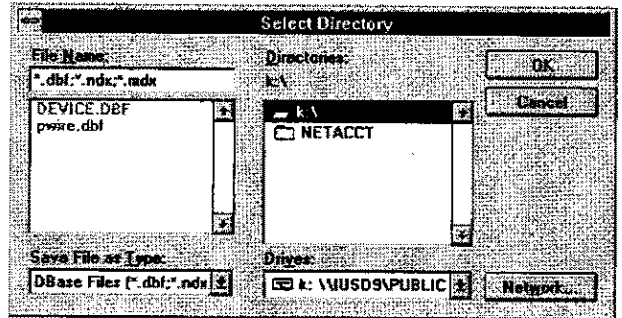
Click on the <<Add>> button. *Add Data Source* window will be displayed.



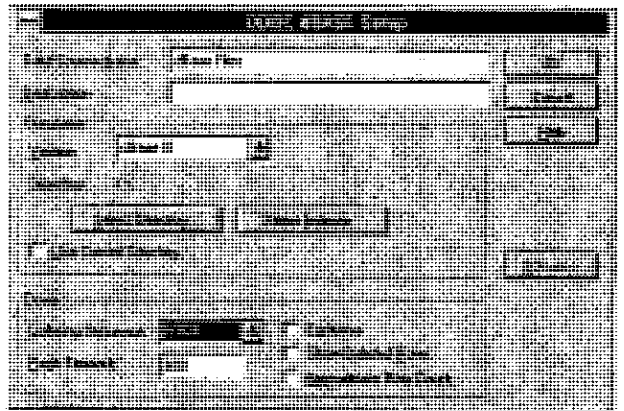
Select dBase Driver and click on the <<OK>> button. *ODBC dBase Setup* window will be displayed. Fill in 'Data Source Name' and select the version from the dropdown list, then click on the <<Select Directory>> button.



Select Directory window will be displayed. Select drive of the data, for example: k:\ and click on <<OK>>.

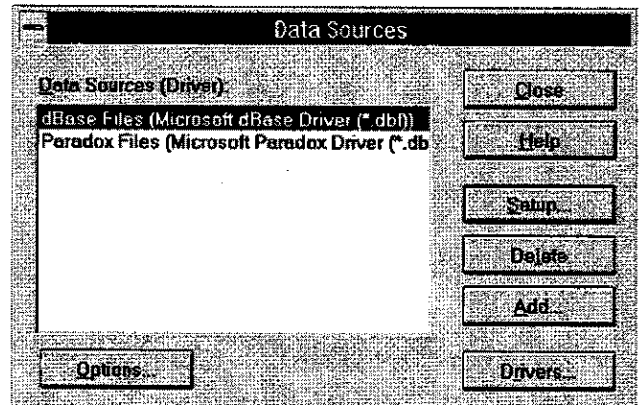


ODBC dBase Setup window will be updated as follows:



Click on the <<OK>> button to close all the open windows as you go back to the *Data Sources* window.

After repeating the above procedure for Paradox Driver, *Data Sources* window will look as follows:



Click on the <<Close>> button to complete data source definitions.

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('tnow',put(time(),hhmm2.));
  call symput('tday',put(today(),weekdate3.));
  Filename logfile "n:\netacct\infrface\log&tday.&tnow.txt";
  PROC printto log=logfile new;

* Get technical information about premises wiring from ;
* dBase file Pwire ;
Libname netacct 'n:\netacct\infrface';
PROC sql;
  connect to odbc (dsn='dBase 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.cmsbx 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 "stc id/circuit#" as stcid, "monthly rate" as
  devrate, "removal date" as remvdate from Finance)
  full join
  netacct.cms
  on stcid = deviceid;
  delete from netacct.cmsbx
  where remvdate ^= . and jackid = '';
  update netacct.cmsbx
  set deviceid = stcid
  where deviceid = '';
  alter table netacct.cmsbx
  drop stcid;
  disconnect from odbc;
quit;
```

```
* Get hardware information from paradox file Hardware ;
PROC sql;
  connect to odbc (dsn='Paradox Files');
  create table netacct.cmsbxhrd as
  select deviceid, hubport, netseg, nettype, jackid,
  devrate, devtype, trim(manufact)||' '||trim(model)||
  ' '||trim(descript) as devdesc length=40,
  trim(bldnflor) || ' ' || trim(roomsect) as devloccn
  length=12, substr(frc,1,3) as frc length=4,pe,
  contact
  from connection to odbc
  (select "stc id/circuit#" as stcid,
  type as devtype,
```

```
  manufacturer as manufact,
  "model#" as model,
  description as descript,
  "building/floor" as bldnflor,
  "room/section" as roomsect,
  "rental frc" as frc,
  "rental pe" as pe,
  "rental contact name" as contact
  from Hardware)
  right join
  netacct.cmsbx
  on stcid = deviceid;
  update netacct.cmsbxhrd
  set frc = '0' || substr(frc,1,3)
  where frc ^= ' ';
  disconnect from paradox;
quit;
```

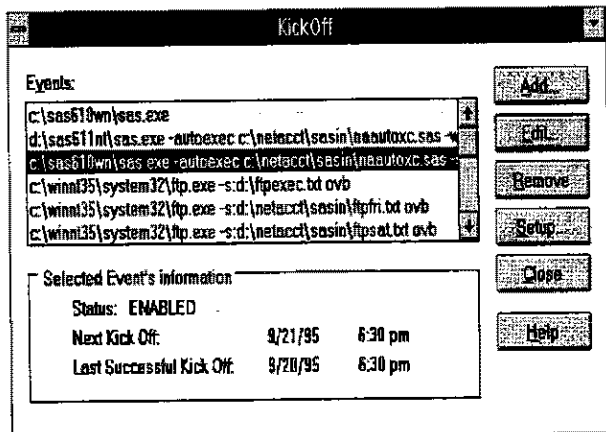
```
* Create a transport file ;
PROC cport data=netacct.cmsbxhrd
  file='n:\netacct\infrface\cport.txt';
```

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

Automating the daily process

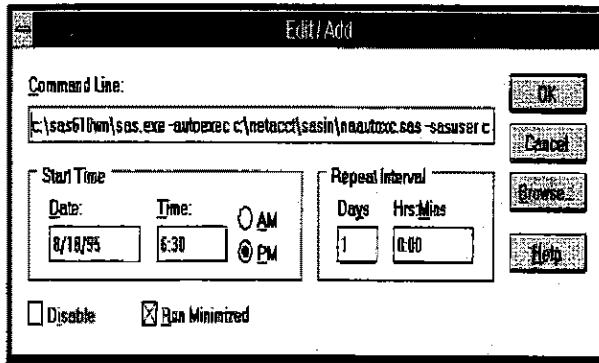
The remote files are updated during the day by the application areas so our daily job is executed after working hours both to avoid conflict and to get the files reflecting all the changes for the day. As a bonus, running at night provides better performance as there is less traffic on the network (no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

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:\sas610m\sas.exe -autoexec
c:\netacct\sasin\naautoxc.sas -sasuser
c:\sas610m\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.

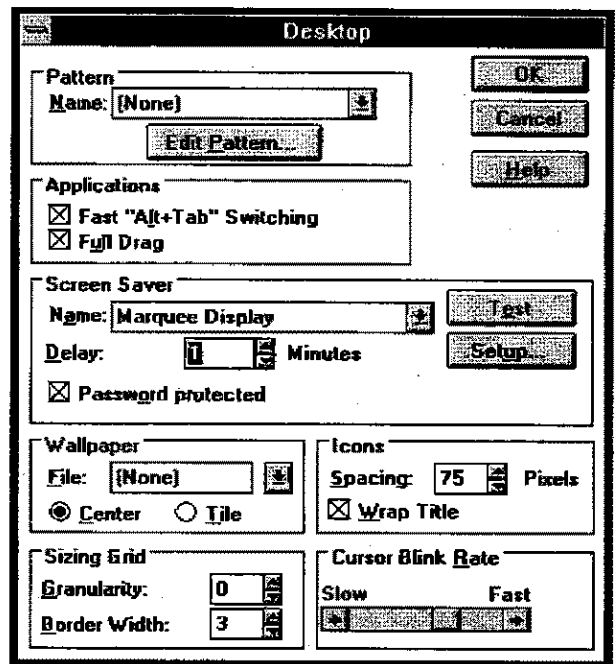
The program code shown in the section "SAS code to access data via ODBC" is stored in 'n:\netacct\sasin\nadaly', where it will be referenced by the daily processing. The SAS autoexec file needed to include the daily job for execution is 'naautoxc':

```
* SAS Autoexec file ;
Filename sasin '\mcc1\acctng\netacct\sasin\';
%include sasin(nadaly.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 CMrmdd where 'mdd' is the month and the day of the data. The datasets are stored in a PDS called 'CARS.DATA.NETymm.SAS6' where 'yymm' 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 (0155,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/netacct/interface
binary
get cport.bt 'cars.aa.netacct.cmstxhrd.txt' (REPLACE
QUIT
/* Phase 1 Step 2 - Convert transport file to a SAS dataset
/*
//P1S2 EXEC SAS6
//IMPRFILE DD DISP=SHR,
```

```
//          DSN=CARS.AA.NETACCT.CMSTXHRD.TEXT
//NET      DD DISP=OLD,
//          DSN=CARS.DATA.NET%YYMM..SAS6
//SYSIN    DD *
PROC CIMPORT DATA=NET.CM%MMDD INFILE=IMPRFILE;
RUN;
```

The dataset is then merged with network traffic data to identify the traffic generated by each workstation. Various summarizations are done to produce reports to show traffic by financial area, network segment, time of day, day, week, month, etc...

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.

	Creating Table NETACCT.CMS with 8145 rows and 41 columns mm:ss	Modifying Table NETACCT.CMSTX with 7 columns mm:ss	Updating 8145 rows NETACCT.CMSTXHRD mm:ss
486-66 cpu with 16M			
SAS 6.10 - Windows 3.11 with 16-bit drivers	3:35	1:58	6:13
SAS 6.10 - WindowsNT with 32-bit drivers	2:35	1:22	2:48
SAS 6.11 - Windows NT with 32-bit drivers	2:28	1:11	2:32
586-100 cpu with 32M			
SAS 6.10 - WindowsNT with 32-bit drivers	0:56	0:46	1:25
SAS 6.11 - Windows NT with 32-bit drivers	0:54	0:44	1:16

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

SAS Institute, Inc., *SAS/ACCESS Software Changes and Enhancements: SQL Procedure Pass-Through Facility, Version 6*, Cary, NC: SAS Institute Inc., 1994

SAS Institute, Inc., SAS Technical Report P-262, *SAS/ACCESS Interface to ODBC: SQL Procedure Pass-Through Facility, Release 6.08*, Cary, NC: SAS Institute Inc., 1993

SAS Institute, Inc., *SAS Guide to the SQL Procedure: Usage and Reference, Version 6, First Edition*, Cary, NC: SAS Institute Inc., 1989

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