Integrating Windows Clients and the SAS® System into the Enterprise

Gary Mehler, SAS Institute Inc., Cary, NC

Abstract

There are several ways to utilize the full potential of the Version 7 SAS System on your Windows system. SAS System products for the PC provide many client/server architectures that can be used to access, update and analyze the entire spectrum of enterprise data. Aside from the SAS System itself, two other clients for accessing SAS data on Windows systems are covered: the SAS Universal ODBC Driver (tutorial included) and the SAS System Viewer.

Introduction

By virtue of the SAS System's Client/Server support, SAS application developers can implement a wide range of data management and analysis configurations. In addition to using the SAS System itself on Windows client systems, users needing lighter-weight data access can also use the SAS System Viewer to view SAS data in tabular form. If the data need to be migrated to another application, the SAS Universal ODBC Driver can provide this access without depending on the base SAS System.

MVA ensures that all of the capabilities of the SAS System on larger systems (such as mainframes and UNIX systems) can run on PC hardware as well.

The SAS System for Windows, Version 7

Version 7 of the SAS System for Windows improves both its user interface and its interoperability with other PC applications. In addition, it takes better advantage of Windows NT-specific features, and is easier to administer.

User interface enhancements continue to provide a more native look and feel expected on the Windows platform. Key GUI enhancements include: print preview, more complete printing options support, command bar and toolbar enhancements, Program Editor enhancements, and support for Microsoft IntelliMouse. These features are covered in more detail in the paper, *"The SAS System for Windows, Version 7"*, also in these proceedings.

Windows NT-specific enhancements take more complete advantage of high-end equipment available on Windows NT Server. Key enhancements include: performance improvements, more support for large data (more than two gigabytes) in more procedures, support for more RAM for large-scale computing projects, a new NT service for SAS/SHARE administration, and compliance with the requirements of the "Designed for Microsoft BackOffice" logo. These features are covered in greater detail in the paper, *"Taking Advantage of the SAS System on Windows NT"*, also in these proceedings.

SAS System Client/Server Support

The SAS System for Personal Computers provides a window to your enterprise data and offers cooperative processing configurations that can access the diverse and scattered data formats while making optimal use of the wide range of enterprise hardware. The following SAS System features and products form the basis for these client/server computing solutions.

- MultiVendor Architecture™ in the SAS System is available for many types of computer hardware and operating system combinations. The compatibility in terms of SAS programs, user interface, and data transportability make inter-system use of the SAS System on various platforms an attractive foundation for enterprise architecture.

- Remote Libraries allow SAS System client applications on one hardware/OS platform access to remote SAS data libraries on any other hardware/OS platform even when the machines use different internal data representations. RLS can also provide gateway access to external DBMS systems that the server can access using SAS/ACCESS® engines. You can also use RLS with all three types of SAS data views.

- SAS/SHARE® software is a multi-user server that can be accessed from any SAS System client. The SAS/SHARE server manages libraries of SAS data sets and catalogs in shared read/write mode.

- SAS/CONNECT® Data Transfer Services allow data downloads and uploads to your PC for local processing. These transfers are performed through PROC UPLOAD and PROC DOWNLOAD, and are useful when local PC processing power is plentiful.

- SAS/CONNECT Compute Services allow SAS statements and programs to be forwarded to a SAS System running on another computer. As remote processing progresses, the program listing and any graphical output are presented on the local system as

if the program had executed locally. This allows compute or data intensive processing to occur on the most capable computing resource or at the location where large volumes of data reside.

- SAS/ACCESS engines provide the ability to access many different DBMS and PC file formats. When used by a local ODBC server or a SAS/SHARE server, the ODBC and SAS clients on one platform can use the server as a gateway to access external database systems. The database systems can be local or remote to the SAS application or SAS server that is using the SAS/ACCESS engine.

- SAS/ACCESS to ODBC allows the SAS System on Windows open access to numerous DBMS formats from a wide range of vendors. This means that SQL queries and other transactions to be executed on databases such as dBase, Paradox, Access, Excel, ORACLE and SYBASE.

- SAS ODBC Driver allows other applications to directly access data from a running SAS session. This driver provides read/write capabilities to SAS data to ODBC-compliant applications. This driver allows a separate application to act as the client with which the user primarily interacts. When used with SAS/ACCESS products, this driver also allows other types of data to be accessed by ODBC-compliant applications.

- SAS Universal ODBC Driver is a standalone driver that can be distributed with SAS data to provide read-only access to SAS data. As a standalone program, it does not require that the SAS System be present or running to provide data to ODBC-compliant applications. This driver allows ODBC-compliant applications to operate on PCs with lesser capabilities, since the SAS System doesn't also have to be operational to provide data access. A usage scenario with Microsoft Excel is included later in this paper.

- The SAS System Viewer is a lightweight standalone program that provides viewing and printing capabilities for SAS data. It provides a read-only view of SAS data in a spreadsheet-like layout, and supports minimal data functions, including subsetting with WHERE clauses, sorting, and partial viewing of large files. The Viewer is discussed in more detail.

Application Configurations

When using the SAS System in a large enterprise, the number of ways applications and databases can be organized is large. Also, many applications require access to very large amounts of data, and may have heavy computational requirements as well. While the number of configurations can seem daunting, keeping a few key factors in mind can help during the decision process. The type of configuration chosen may depend on a few factors:

- Network topology. Particularly when operating across a Wide Area Network (WAN), the bandwidth of networking equipment can be a severly limiting factor. When operating upon large amounts of data, keep in mind how much of that data needs to be transported across network. Even on fast (100Mbps) networks, latency delays can impact processing throughput. On WANs, even that fastest currently available modems (56kbps) may require that very little data travel across the network.

- Server capabilities. If the data are stored on a pure fileserver (unable to also run application software), client processing is about the only option. The data must be accessed across the network and processed on the local desktop workstation. Larger Windows NT servers may be capable of running applications such as the SAS System while also performing file service operations without significant impact to either.

- Data location. If data are stored on central server systems, it may make sense to perform processing on the same system. If the central server is capable of running the SAS System (for instance, mainframe, UNIX server, or NT server), any type of SAS processing can occur on the server.

- Data size. Large data warehouses (on the order of 50GB) are impractical to transfer over a network repeatedly and reliably. Subsetting the data before retrieving it or processing it on the system on which it is stored make the most sense in most cases.

- Computational resources. If the system acting as a data or fileserver has limited computational resources (RAM and CPU time availability for application use), local processing makes the most sense. If the client systems are relatively more capable than the server, processing on the client (usually a personal computer) should be considered. Particularly with newer, high-end PCs, RAM availability of 64MB and beyond as well as CPU speeds in excess of 200MHz make for very capable desktop analytical machines.

- Third-party data requirements. If the data are stored in another store such as Microsoft SQL Server, Lotus Notes, or an ODBC-compliant application, a SAS/ACCESS product may be required. If SAS/ACCESS software can operate on the same machine as the data source, this may improve performance.

- Data processing requirements. The use of the data by the client software may have give some flexibility on the client side. The light-weight SAS System Viewer, while limited in capabilities, is a very lightweight client.

These factors given, the ways in which an application can be designed utilizing the SAS System are briefly described:

- Client/Server applications allow processing to occur on both the client and the server. Data is most likely stored on the server, but it could be stored on both the client and the server. The server also acts as a computation server in addition to also being a data server. The user interface runs on the client, and SAS/CONNECT software (remote computing services) enables communication between the client and the server. Computing on the server is as easy as inserting **RSUBMIT** in a SAS program to cause execution to occur on the remote server.

- File Server use results in processing occurring only on the client system, particularly if the fileserver is incapable of running general software packages. All data is stored on the server, and is moved across the network to client systems for processing. Common on Novell fileservers, or in large organizations in which applications are not allowed on designated fileserver machines, this puts heavy demands on networking capabilities as well as requiring that client systems be capable processing machines.

- Data server, in which SAS/SHARE software manages concurrent user access to the data. This is similar to the previous Fileserver configuration, except that SAS/SHARE handles concurrent read/write sessions on the same data sets. A major benefit of this approach, though, is that the SAS/SHARE server can perform subsetting on the data before transmitting it over the network to client systems.

- Terminal emulation, which keeps all applications operating on the server system, with a terminal protocol delivering the GUI contents across the network to a client PC. In this scenario, the client PC does nothing but render the GUI window contents. While this minimizes the requirements for client system capabilities, it also maximizes the hardware requirements for the server system. In this scenario, all processing must occur on the server. As the number of concurrent users rises, this may become a substantial load on the server. This is a far different situation than supporting a large number of users in file server or data server mode, since individual computing requirements may put a much larger load on the server.

This configuration is currently possible on NT 3.51 using a software package called WinFrame, distributed by Citrix Systems. There is no similar support available for NT 4.0, and as of NT 5.0, this capability will be incorporated into Windows NT Server itself (the support for NT 5.0 is currently being called *Hydra*).

- Third-party program operation occurs when the SAS System is used in conjunction with a custom user interface program. The user interface uses OLE Automation, or ODBC data access (or both)

In these scenarios, the requirements for computational power, data storage capabilities, SAS System licensing requirements, and network bandwidth vary. Similarly, performance and customizability may also need to be considered when deploying a new application in a large, complex enterprise.

SAS ODBC Driver

The SAS ODBC Driver for the SAS System allows third party ODBC compliant Windows applications, such as Microsoft's Excel or Visual Basic, open access to SAS data sets. The SAS ODBC Driver is a multi-tiered driver meaning that it does not directly access SAS data sets but rather communicates with either a local SAS ODBC server or a remote SAS/SHARE server.

SAS System servers transparently handle all cross architecture issues such as floating point data translation between different hardware/OS platforms. The local SAS ODBC server comes with the Base SAS System and allows access to local SAS data. To access SAS data remote from the local machine, you use the SAS ODBC Driver to connect to a SAS/SHARE server that has the SAS/SHARE*NET® software installed.

When used in conjunction with SAS/ACCESS software on the local or remote server machines, the SAS ODBC Driver serves as a read/only gateway to the wide variety of DBMS data supported by the various SAS/ACCESS engines. The SAS ODBC Driver supports both local and network DDE and TCP/IP for connection to local ODBC or remote SAS/SHARE servers. SAS Institute will expand the available access methods for the SAS ODBC Driver in the future.

SAS Universal ODBC Driver

New since the 6.12 production release, the Universal ODBC Driver opens up SAS data to ODBC applications without requiring access to the SAS System itself. This driver handles data that has been generated on the PC, UNIX or on OpenVMS. It permits the distribution of data generated by the SAS System for use with other ODBC applications.

The SAS Universal ODBC Driver allows the (read-only) opening and querying of the following types of data sources from Windows applications that support ODBC: standard SAS data sets, CFO Vision Financial Database (FDB) files, SAS multi-dimensional database (MDDB) files, SAS catalog tables of contents, and JMP files. Data sources can be setup for files stored locally, as well as those accessed through FTP or HTTP protocols on remote internet servers.

This driver makes SAS data more accessible, since ODBC applications can access SAS data formats through this small, inexpensive driver. For applications requiring read/write access to SAS data formats, the original SAS ODBC Driver should still be used. Both drivers provide SQL query capabilities.

SAS Universal ODBC Driver Tutorial

The SAS Universal ODBC Driver currently supports ODBC machine data source names (DSNs), but does not support file DSNs.

An ODBC-compliant application may choose to support the use of machine DSNs, file DSNs, or both. If an application (such as Microsoft Excel 97) only supports use of a file DSN, a workaround is required to get to SAS data from the application.

We will use Microsoft Excel 97 in this example to illustrate the workaround.

Creating the Machine DSN

(If you already have a machine DSN created (i.e- you can already import the desired data from another ODBC-compliant application), then you only need to link the machine DSN to a file DSN. Skip this section)

First, we must create the machine DSN using the ODBC Data Source Administrator.

1. From the Windows Control Panel, select the ODBC (or 32bit ODBC) icon to run the ODBC Data Source Administrator.

2. Select the User DSN tab and click Add.
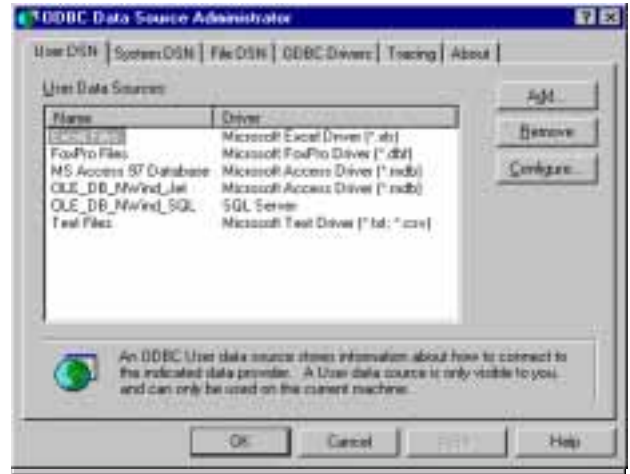
3. Select the SAS Universal ODBC Driver and click Finish.

4. Enter the name of your data source and the location of the SAS data that you would like to access. Click OK, and you should see a new user data source, My SAS Data, appear in the list of user DSNs.

Linking File and Machine DSNs

In order to allow Excel to get to SAS data, we must create a file DSN which 'links' to the machine DSN just created.

1. From the ODBC Data Source Administrator, select the File DSN tab:

(Look for the name of your machine DSN in the list of file DSNs. If it exists, then it was automatically created for you when you created the machine DSN. You can skip the rest of this section.)

2. Click the Add button to add a new file DSN. Select the SAS Universal ODBC Driver and click Next.

3. Enter the name of the File DSN you would like to create (you can use the same name as your machine DSN if you'd like—append '(not sharable)' to the end of the name). Click Next, then Finish.

(Steps 4 and 5 are necessary because the Universal ODBC Driver does not support File DSNs at this time. This results in a generic File DSN getting created without a data file listed in it. Out purpose is to get this file created, so we can edit it later):

4. An Open dialog will appear. Click Cancel to close it.
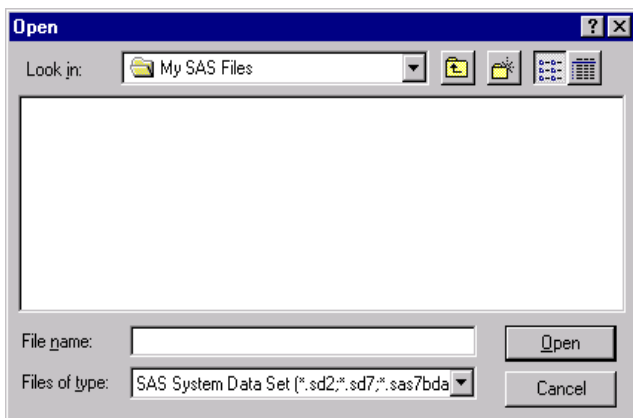
5. The following message will appear. Click Yes to save the file DSN anyway.

You have just created a file on your system named *My SAS Data (not sharable).dsn*. This file DSN does not point to a data file (as a result of steps 4 and 5 above). This is desired. The file does, however, incorrectly contain a DRIVER line.

The last step in linking the file DSN to the machine DSN is to edit this file. Since ODBC installations vary between machines, you will likely need to do a Start Menu/Find/Files or Folders and search for *.DSN in order to find *My SAS Data.dsn*. (A common place for File DSN files is:

C:\Program Files\Common Files\ODBC\DataSources).

6. Edit the *My SAS Data (not sharable).dsn* file as follows:

> Delete this line:
>
> DRIVER=SAS Universal ODBC Driver
>
> Add this line, which contains the name of the machine DSN to link the file DSN to:
>
> DSN=My SAS Data

Importing the SAS System Data into the Application

At this point you should have a file DSN created which points to a machine DSN which points to the data you desire to import.
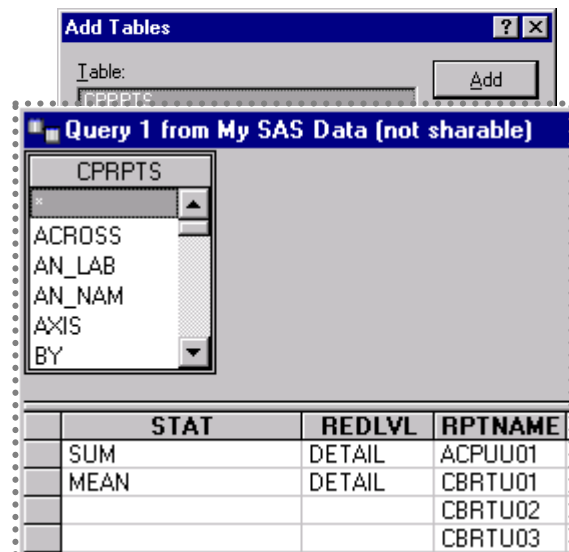
To import the data into Microsoft Excel 97, do the following:
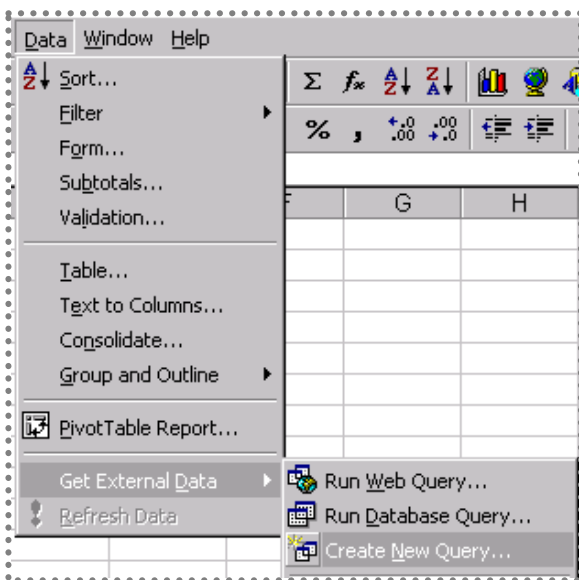
1. From the Excel *Data* menu, choose *Get External Data,* then *Create New Query.*

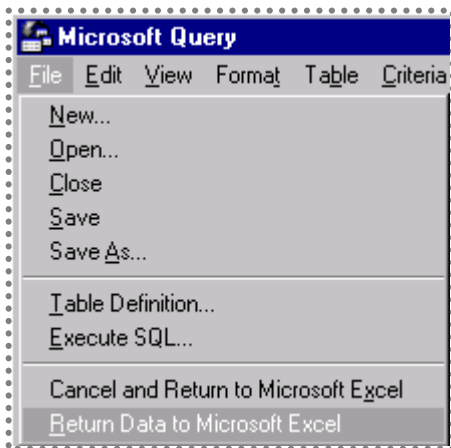2. Select *My SAS Data (not sharable)* from the Data Source listing.

3. Microsoft Query will be launched. Select the tables you are interested in querying, and click *Add*. When finished selecting tables, click *Close*.
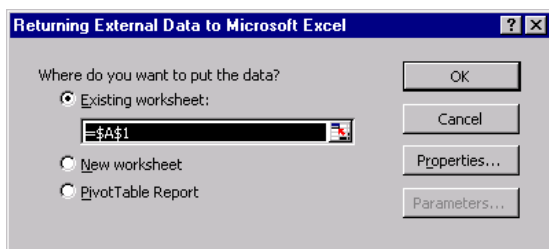

4. Use MS Query to construct a query to get the desired data from the tables.

5. From the MS Query *File* menu, select *Return Data to Microsoft Excel*.

6. Excel will prompt you for where you would like to place the data. Make the appropriate selections, and click OK.

Excel will be populated with the data you have selected. Of course, once the DSN has been setup, subsequent ODBC imports into an application like Excel are simpler.

Interoperability using OLE

The SAS System for Windows fully exploits Microsoft OLE functionality which is actually a very sophisticated form of client/server computing. The SAS System for Windows supports the following OLE features: drag and drop, visual editing, automation (controlling other programs or components), custom controls (embedding other controls in an application), and an automation server (controlling the SAS System from another application).

For a more complete discussion of the OLE support available in the SAS System, please refer to the paper, "*OLE and the SAS System for Windows Release 6.12*."

Other SAS System Interoperability Support

Several other methods are available in the SAS System to allow inter-program operation and data sharing.

The SAS System's support **Lotus Notes** enables users to populate Notes with SAS output including attachments such as data sets, graphs, and reports.

Electronic mail can be sent from within the SAS System either by using the Send Mail dialog (available from the main SAS System **File** pull-down menu), or by using the **EMAIL access method** from a DATA step or SCL code. The SAS System supports MAPI and VIM e-mail client packages and supports SMTP mail delivery directly to central mail servers.

Three access methods provide increased cross-platform data interoperability. The communications protocols can replace normal external file processing, so they can be used anywhere an external file would be accessed from the SAS System. The FILENAME statement can be used to indicate that you want to use an alternate external data source access method. These access methods include: the **FTP access method,** the **TCP/IP socket access method** and the **URL access method.**

SAS System integration with the Windows Explorer allows Windows 95 and Windows NT 4.0 to understand different types of SAS System files (including SAS data sets, catalogs, and SAS System source code), which are represented by unique icons. Each file type is also associated with default actions that can be accessed through the Explorer's pop-up menus.

The SAS System's ability to access external dynamic link libraries (DLLs) from within the SAS System can extend your processing power. Functions contained in custom DLLs developed in-house, or system or third-party DLLs can be called from within the SAS System by experienced programmers. The external DLL can be loaded, routines invoked, and the DLL unloaded upon completion.

For a more complete discussion of the new features available in the SAS System, please refer to the paper, "*The SAS System on Windows, Version 7*," also in these proceedings.

The SAS System Viewer

The SAS System Viewer has been enhanced in a number of areas since the 6.12 Release. These are summarized below.

- Active Document Support. The SAS System Viewer can be embedded within other applications (such as a web browser) to provide a more consistent interface across applications. This support (formerly known as DocObject support) means that the Viewer can run within a different container application. So, if a SAS data set is encountered as a hyperlink on a web page, the SAS System Viewer will bring itself up within the web browser when that link is activated. When the browser "back" button is selected, the Viewer exits

and the previous state of the browser returns. The figure below shows the Viewer operating within
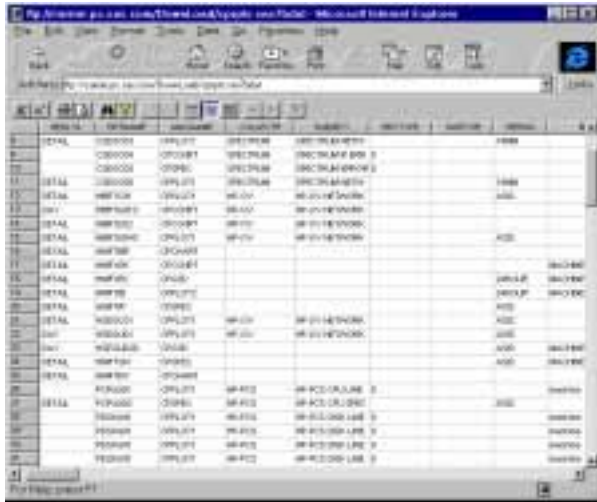
Microsoft Internet Explorer.

- **Improved Printing Support.** Several improvements were made in the area of printing and print preview. The Viewer now correctly interprets pagination embedded in SAS Log and List output files that have been saved to disk. When retrieved for viewing in the SAS System Viewer, pagination information in the saved file is honored both for viewing on-screen and for subsequent printing through the Viewer. With pagination information visible, a user can more accurately determine how their output will be printed. In the example shown below, the SAS System Viewer is displaying a SAS output (.LOG) file, which shows that the correct font and pagination information is being used.

- **Workbook View Support.** The Viewer supports three views on data: Single document interface (for "pinned" files), multiple document interface (the normal view), and workbook view. In the single document interface, only one file can be active at any one time. Multiple document interface supports multiple windows within the Viewer, each of which can be sized/resized individually. The new workbook view is like the multiple document interface in that multiple documents can be active in the Viewer at any given time, and adds a tab selection bar at the bottom of the Viewer to handle movement among the files.
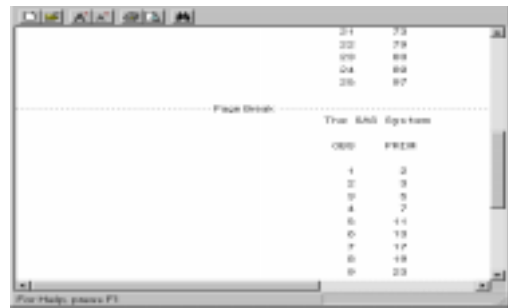
- **Data Subsetting Support.** To facilitate viewing of very large SAS data sets while maintaining a fast response time, the Viewer can be configured to subset the number of observations that are read and viewed
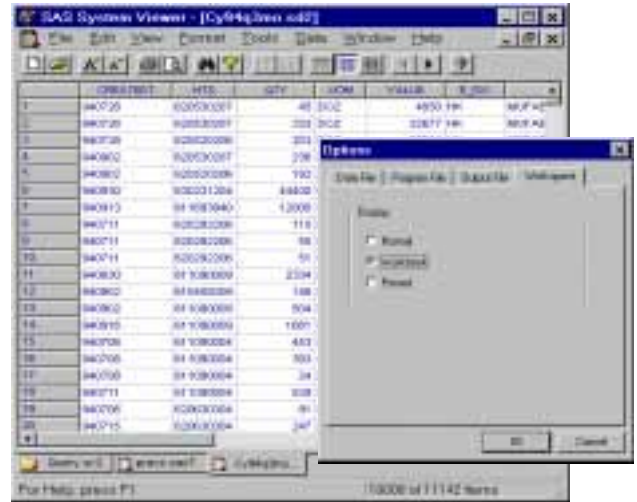
at any one time. For example, the Viewer can be set to read only 1000 observations initially, and the user can select to move to the next range of observations. .In the following example, note the range selection dialog as well as taskbar arrows (left and right arrows) that support sequencing through ranges of data.
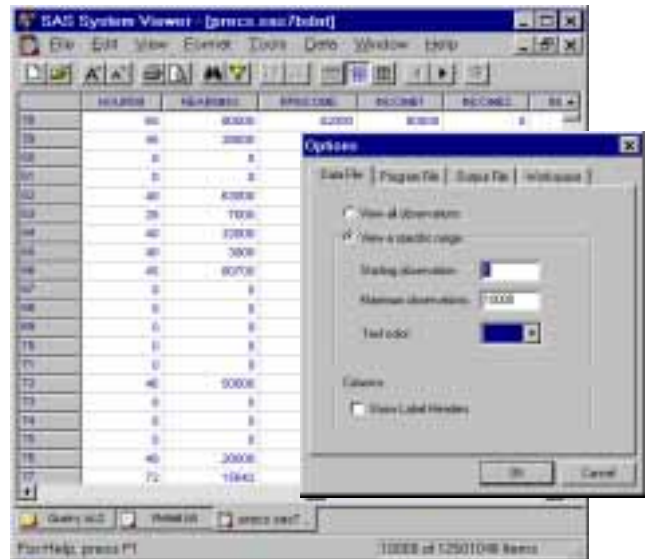
- **Variable display support.** The viewer supports viewing information about dataset variables in a

spreadsheet-like display, as shown in the following example. This information can be printed as well.

- <u>Where Clause History.</u> To assist users who frequently enter complicated SQL queries in the Viewer, a history drop-down listing is now available.

- <u>Go To Observation Support.</u> When browsing very large data sets (and also using data ranging support), it is also convenient to be able to jump to a specific observation. This is even more useful since the Viewer supports reading very large files (up to 2 gigabytes).

Conclusion

In summary, the SAS System provides a wide range of computing and data access options that meet just about any system requirement from desktop analysis to data warehousing. The desktop PC running the SAS System can play an important role in your enterprise computing strategy and will increase both your productivity and your capability. If you require standalone operation, the SAS System Viewer and the SAS Universal ODBC Driver may be what you need.
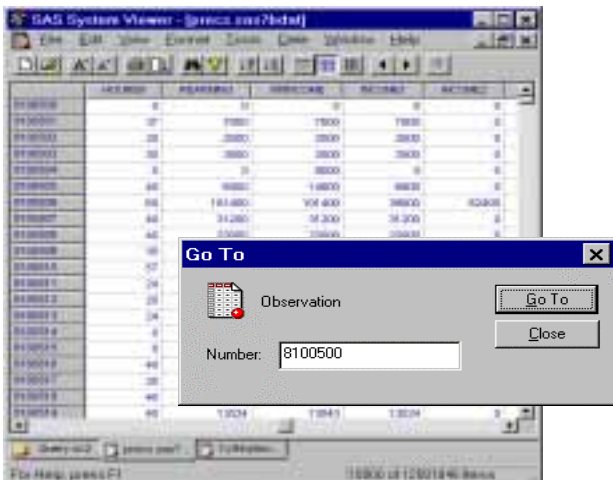
Acknowledgements

Trademarks

SAS, SAS/ACCESS, SAS/CONNECT, SAS/SHARE and SAS/SHARE*NET 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.

References

Clegg, Jennifer and Rigsbee, Carol (1997) "OLE and the SAS System for Windows Release 6.12," *Proceedings of the Twenty-second Annual SAS Users Group International Conference*

Clegg, Jennifer and Mehler, Gary (1998) "The SAS System for Windows, Version 7," *Proceedings of the Twenty-third Annual SAS Users Group International Conference*

Mehler, Gary, (1997) "Integrating the SAS System for Personal Computers into Your Enterprise," *Proceedings of the Twenty-second Annual SAS Users Group International Conference*

Mehler, Gary, (1998) "Taking Advantage of the SAS System on Windows NT," *Proceedings of the Twenty-third Annual SAS Users Group International Conference*

SAS Institute Inc. (1995) "Client/Server Computing with the SAS System: Tips and Techniques," SAS Institute Inc.

Author Information

Gary Mehler
SAS Institute, Inc.
SAS Campus Drive
Cary NC 27513
Phone: (919) 677-8000
e-mail: gjm@wnt.sas.com