

## SAS® For Road Warriors And Telecommuters: How To Avoid The Floppy Shuffle

Michael C. Harris, Amgen, Thousand Oaks, CA

### ABSTRACT

Sooner or later, many people who use computers in their offices will acquire desktop or portable computers so they can have the flexibility to work at home. They may simply desire to be able to work evenings and weekends at their convenience, or employers may have formal work-at-home programs to help meet local environmental regulations. Thanks to the proliferation of TCP/IP packages for personal computers (PCs), SAS users have several options for solving the problem of remote data access without carrying data sets around on disks. This paper describes the hardware, software and system configurations necessary to make remote computing effective. Character based terminal emulation is discussed briefly, then more advanced and exciting approaches, including X Windows over serial lines and creative use of SAS/CONNECT® are presented. Finally, potential problems and performance issues are addressed.

### INTRODUCTION

Until recently, TCP/IP (Transmission Control Protocol/Internetworking Protocol) software, the *lingua franca* of Unix family networks was not widely available for PCs and required a high level of technical expertise to configure and maintain. The tremendous popularity of the Internet, especially the World Wide Web (WWW) has spawned software options to fit almost any budget and operating system. Concurrently, new high speed modem standards have enabled data transfer at rates that make it practical to do remote computing in ways other than simple character mode terminal emulation. This paper focuses on Windows® and OS/2®, the most popular PC operating systems.

### TERMINAL EMULATION

The simplest form of terminal emulation doesn't require a TCP/IP stack at all. On the remote side, all that's needed is a communication program capable of dialing and providing a type of terminal emulation that the host system supports. On the host side, you'll need a modem server and permission to dial in. If you are dialing in to a network, you'll need to contact the network administrator to get the phone number, a password, and whatever other information you'll need to make a connection. Lastly, the SAS System must be available on the host system. See Figure 1 below for an example of SAS in a Kermit session.



Figure 1. SAS in a Kermit session

The communication software options across all PC operating systems are too numerous to mention. One of the best known

commercial packages for both MS-DOS® and Windows is Procomm Plus®, by Datastorm. At the other end of the spectrum, implementations of the venerable Kermit terminal emulator and file transfer protocol are available at no cost for virtually all operating systems. Like its commercial siblings, it offers a variety of terminal emulations and keyboard mappings if you are willing to search bulletin boards for the necessary files. Kermit is well documented, so if you have special requirements it is perfectly reasonable to attempt to create your own keyboard mapping files.

Simple character mode terminal emulation will allow you to run the SAS System and will give acceptable performance with any modem speed from 2400 bit per second (bps) on up. Depending on the host system, you may encounter keyboard mapping problems that are difficult to resolve and your predefined keys within SAS may not function as expected. As result, you may find the SAS Program Editor awkward to use and system editors such as vi on Unix systems may be even worse. Lastly, there is no hope whatsoever of displaying graphics in character mode sessions. Users accustomed to running SAS from the command line will feel at home with this simplest of approaches and may not need the flexibility offered by TCP/IP.

### WHY TCP/IP

TCP/IP is a set of protocols designed to allow the interconnection of networks running dissimilar protocols on virtually any hardware.<sup>1</sup> By extension, networks themselves can consist of computers running a variety of operating systems on almost any hardware. For example, a Novell Netware server running its native IPX protocol can coexist with a Unix server on the same network if a TCP/IP protocol stack is also running. Today's computers are capable of running multiple protocols on a single network adapter without conflicts. SAS/CONNECT supports a variety of communication access methods, but the one that is available on PCs running Windows, OS/2, Windows and Unix is the TCP method. When we make a dialup TCP/IP connection, our remote computer becomes a client on the network with the potential to access all services available to local clients.

### SYSTEM REQUIREMENTS

The hardware necessary to deliver acceptable performance is not at all unusual by today's standards. A 486-33 desktop or portable PC with 8 megabytes of random access memory (RAM), a VGA monitor and compatible video adapter, a 9,600 bit per second modem, and a hard drive of 320 megabyte capacity would be an entry level machine for running a TCP/IP stack under either Windows or OS/2. If you plan on running SAS locally and using SAS/CONNECT, 12 to 16 megabytes of RAM will be required. An enhanced system could include a faster modem, an accelerated video adapter, a 17 inch monitor for home use, and a fast 486 or Pentium® CPU.

Your software choices will be driven by two factors: What else you plan to use your connection for, and your budget. On the Windows side, Peter Tatum's Trumpet Winsock 2.0b is a shareware package that provides a dialer and a Windows Sockets-compliant stack that will function with SAS connect. Many shareware and freeware client programs are available through the Internet, CompuServe, and various bulletin board systems. hobbes.nmsu.edu is an ftp site worth investigating if

you are interested in freeware or shareware for Windows. If you prefer commercial software, Internet In A Box<sup>®</sup> by Frye Inc. ships with a Winsock-compliant TCP/IP stack and an assortment of client programs.

OS/2 users will find everything they need to make a dialup connection included with the latest version of the operating system, Warp<sup>®</sup> Version 3.0. In addition to support for native OS/2 clients, a Windows sockets interface is provided so you can choose clients which best suit your taste. Note that SAS/CONNECT will run in Windows sessions under Warp.

To use TCP/IP over telephone lines you must run an underlying protocol. The two available are Serial Line Internet Protocol (SLIP) and Point to Point Protocol (PPP).<sup>2</sup> PPP is the newest, and in my experience is more robust than SLIP. Most commercial and shareware packages include both, so your choice will be based on what is supported by the system to which you will connect. Regardless of which you use, you'll need the cooperation of a system administrator to receive a set of Internet Protocol (IP) addresses for your machine, the default router, and the domain name server if you are assigned a static IP address. You may need other addresses if full Internet access is available. If your software doesn't automatically handle prompts from your particular modem server or if your network dynamically assigns IP addresses, you'll need to write or modify a login script. Configuring the software is usually straightforward if you have all the necessary information. If you don't or incorrectly enter any required parameters, it is a certainty that you won't be able to make a stable connection. If you lack a required IP address, resist the temptation to make up your own. Other users of the network may experience bizarre problems and you might make new enemies.

## TERMINAL EMULATION UNDER TCP/IP

Virtually all PC implementations of TCP/IP include a version of *telnet*. *telnet* is a protocol which runs over TCP/IP and is also the name of a program which allows remote login to other computers on the Internet.<sup>3</sup> SAS running in a *telnet* session looks quite a bit like a non-TCP/IP terminal session. A crucial difference does exist, though, which may make this an attractive way to run SAS remotely. You can run more than one TCP/IP client at a time. For example, you could have your favorite character mode text editor running in another *telnet* session. Or you could use an *ftp* (File Transfer Protocol) client to retrieve output files to your local machine. Or you could use your favorite editor to write code, then use *ftp* to transfer your programs to the remote system for execution. The ability to have multiple active, visible windows opens up some interesting possibilities.

Yet another kind of terminal emulation running over TCP/IP exists: You can, with software known as an X server, run X Windows programs including SAS in graphics mode with moveable resizable windows, full use of a mouse or other pointing device, and the ability to display high resolution SAS/GRAPH output. The X Windows specification is hardware independent and doesn't mandate a particular look or feel for graphical applications. It provides a set of programming tools that enable creation of portable applications. The look of X programs on a particular platform is determined by a software component called a *window manager*. The Motif window manager is one of the more popular ones in the Unix world. On a PC running Microsoft Windows, it could be the Graphics Device Interface, and under OS/2, Presentation Manager. SAS running on a Unix host but displaying on a remote machine running an X server would appear and act very much like a native application for the operating system running the X server.

SAS users accustomed to X terminals or workstations will feel quite at home running remote sessions on PCs with X servers. It is possible to have multiple windows open, such as *telnet* or *xterm* sessions for issuing system commands, or a native X editor. Any mix of character and graphical user interface applications is possible. Figure 2 below illustrates a SAS session, an X editor and a *telnet* session running simultaneously. You can closely mimic a workstation environment by using the PC as an X terminal.

The number of X servers for Windows and OS/2 is growing. Hummingbird's *eXceed*<sup>®</sup> is available for both platforms and is one of the better known packages. IBM markets *PMX*<sup>®</sup>, a reasonably priced X server, as an add-on to their TCP/IP base kit, now bundled with the OS/2 in a version known as *Warp/Connect*. *PMX* works with the Internet Access Kit software included with *WARP*. At present there are no shareware or freeware X servers for OS/2. A fully functional server (*XWindemo*) was available as freeware for Windows, but has disappeared from popular *ftp* sites.

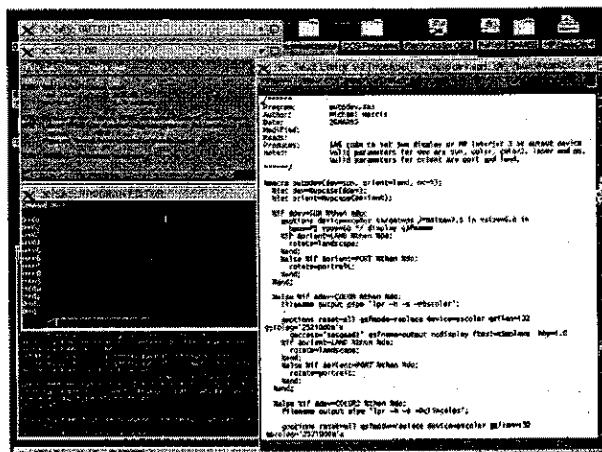


Figure 2. SAS, an editor and a telnet session

## USING SAS/CONNECT

The same functionality that SAS/CONNECT provides through a direct network connection is present through SLIP or PPP. To initiate a session, you use your dialer to log in to the network. Then start SAS on your local machine and issue the *SIGNON* command to run a special script. This script will log you on to a remote host and start a special version of SAS that runs on the host and returns the results of all processing to your local log and output windows. The latest version of SAS/CONNECT features Remote Library Services. You can assign a *libname* to a SAS data library on a remote host and use the data within your local SAS session transparently. There is no need to copy data sets back and forth and no problem with cross-platform translation. You can create local versions of remote data sets with *PROC DOWNLOAD* and remote versions of local data sets with *PROC UPLOAD*. Or you can run *DATA* steps to create local or remote data sets by using appropriate *libnames*.

If you are working with large data sets, you can perform all processing on the remote host by *remote submitting*<sup>4</sup> your programs. Both text and graphical output will be returned to the local session. You have the option of writing code in the local Program Editor Window and remote submitting it, or running an editor in a *telnet* or X window and remote submitting *%INCLUDE* statements pointing to your program.

Perhaps the most difficult part of setting up SAS/CONNECT is writing and debugging a script. The SAS System ships with sample scripts for each supported operating system. You may be able to modify one to suit your needs. Alternatively, you can use the program MKSCRPT.SAS, found in x:\sas\sample (where "x" is the logical drive containing the SAS System) to aid you in writing a script. Last but not least, SAS/CONNECT must be present on the network to which you're connecting. It may require cooperation from the SAS System administrator of the remote system to achieve trouble free operation.

### PERFORMANCE CONSIDERATIONS

Under simple character mode terminal emulation the main performance issue is the speed of screen scrolling. That can be optimized by using the highest speed connection supported at both ends and running full-screen rather than windowed sessions. Maximum speed can be realized by zooming the Program Editor Window during program execution. There will then be no visible scrolling at all.

telnet sessions offer more options for performance tuning. TCP/IP is packet based, with packets consisting of TCP headers followed by data. The telnet protocol sends and receives one character at a time, but a header is required for every character transmitted. Van Jacobson (VJ) compression can greatly reduce the overhead imposed by packet headers. Adjusting the Maximum Transmission Unit (MTU) size to match your host system's can also make a difference. Your SLIP or PPP software will probably include documentation describing other settings you can adjust to get optimal performance.

Everything you can do at the software level to optimize telnet applies to X Windows and SAS/CONNECT as well. In addition, you won't want to run X Windows at less than 9,600 bps. At that speed it is usable but slow. 14,400 bps will yield noticeable performance improvements and 28,800 bps is quite usable. Finally, regardless of how you choose to use your connection, pay attention to the setup string used to initialize your modem. In particular, make sure that you are set up to negotiate the highest connection speed line conditions permit, that auto fallback and fall forward are enabled, and that the appropriate hardware compression scheme is enabled. Modem manufacturers invariably quote maximum effective throughputs for their products that are four times higher than the modem speed. Those high data transfer rates are based on an assumption of highly compressible data. To maximize your effective throughput, select a modem-to-computer (DCE) data transfer rate that is realistic for your connection speed. In general, that will be four times the speed at which you actually connect. If you are fortunate enough to connect at 28,800 bps, you may need special serial port drivers to run the port at 115,200 bps. There are literally hundreds of models of high speed modems and on a recent visit to a computer supermarket I counted no fewer than a dozen software packages promising easy Internet access. If your software doesn't explicitly support your modem, there is no substitute for clear, comprehensive manuals to guide you in building a customized setup string and setting up your operating system.

Finally, there is one more important hardware issue to address in a multitasking environment. If your modem is external, make sure that your serial port adapter uses a buffered UART (Universal Asynchronous Receiver-Transmitter) chip, such as a 16550. Older 8250 style UARTs generate one processor interrupt per character, resulting in dropped characters and resultant retransmissions on a busy system. Buffered UARTS dramatically reduce the CPU load associated with serial port activity. If your

modem is internal it should have a buffered UART for the same reasons.

### USEFUL WEB AND FTP SITES

The latest Trumpet Winsock shareware is available at [ftp.trumpet.com.au](http://ftp.trumpet.com.au).

Winsock applications can be found on the WWW at <http://coyote.csusm.edu/cwis/winworld/winsoc.html>.

TCP/IP applications for OS/2 are available at <http://wc62.residence/gatech.edu/sorenson/tcpip.html>.

Windows and OS/2 applications can be found in archives at [hobbes.nmsu.edu](http://hobbes.nmsu.edu). This ftp site has quite a lot to offer.

The latest version of Kermit (plus lots of useful information about it) is at WWW site <http://www.columbia.edu/kermit/cko191.html>. Its companion ftp site is [kermit.columbia.edu](http://kermit.columbia.edu). Look in /archives.

For SAS support, point your WWW browser to <http://www.sas.com>.

### CONCLUSION

Remote computing with the SAS System has never been more flexible than it is today. Not only do you have a choice of user interfaces, but you can distribute processing power in ways that make the most sense for your unique applications.

### REFERENCES

1. Derr, Daniel P., (1994) *The Internet Guide For New Users*, New York, McGraw-Hill, Inc., 11.
2. Derr 44.
3. Derr 251.
4. SAS Institute, Inc. (1994) *SAS/CONNECT Software, Usage and Reference, Version 6, Second Edition*.

### ACKNOWLEDGEMENTS

SAS, SAS/GRAPH, and SAS/CONNECT are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. IBM, OS/2, and Warp are registered trademarks of International Business Machines Corporation. ® indicates USA registration.

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