

SAS® Under OS/2 The User's Perspective

Andrew Parks
U.S. International Trade Commission
Washington DC

ABSTRACT

SAS for OS/2 is one of the handful of 32-bit applications available today. SAS Institute was one of the first software developers to write specifically for OS/2. Their first entry, Release 6.06, performed admirably under OS/2 versions 1.2 and 1.3, but like those releases of the operating system, it was only 16-bit. The introduction of the 32-bit version of the operating system, OS/2 2.0 (and more recently 2.1) and SAS Institute's 32-bit releases 6.08 and 6.10 for OS/2, made running SAS on an Intel-based desktop system much more appealing. Not only is performance excellent, but the multitasking capability and configuration options combine to produce a very workable operating environment.

INTRODUCTION

This paper looks at OS/2, providing an overview including installation and using the object oriented desktop shell. It then turns to the particulars of running SAS under OS/2, describing the workplace shell, and some of the ways that SAS can be run.

System requirements -- For running SAS 6.08 or better under OS/2 2.1 or better, your system should have the following resources:

CPU	386SX or better, 486 recommended
Clock speed	The faster the better
Math coprocessor	Not required, but recommended
Mouse	A must
Display	VGA
Disk storage	OS/2: 18 to 30 Mb SAS: depends on which products. From 23 MB to 160 MB! Make sure to allow for your \SASWORK area!
RAM	8 minimum, 16 better
Swap file	Up to 40 MB! The less RAM, the bigger the swap file needed.

Regarding RAM and disk storage, it is well to remember the American proverb (c 1980s):

"You can never be too rich, too thin, or have too much RAM and disk storage."

OS/2

OS/2 got off to a bad start with its DOS compatibility, or more accurately, DOS incompatibility. Earlier versions (1.x) gave limited support, allowing one DOS session. With version 2.0 and 2.1 came the ability to run an emulated version of DOS in a separate window, and to run most standard DOS applications seamlessly on the desktop. In fact, you can run many DOS sessions simultaneously. These DOS sessions are called Virtual DOS Machines, or VDMs. You can also run a real (not emulated) DOS session by opening a window that boots from drive A:, or you can run a real version of DOS from an image of the DOS boot disk stored on your hard drive. This gives you the ability to run specific versions of DOS, if an application requires it.

Some of the DOS applications that you will have trouble running under OS/2 are those that require DOS extenders, although some of the developers are designing the applications to run under OS/2, either by modifying the DOS extender, or by re-writing the application for OS/2.

In addition to the ability to run seamless DOS sessions from OS/2, you can set up your system to allow you to boot directly into DOS without loading OS/2 at all. This is the safeguard that many people find they need before they experiment with OS/2. Personally, I found it quite useful when I first installed OS/2 1.21. In order to use my CD-ROM drive or to log onto our local area network, I had to boot into DOS using the "Dual Boot" feature. But since using releases 2.0 and 2.1, I find less and less need to boot into DOS. I can run all the applications I need from OS/2.

OS/2 also runs most Windows applications as well, and

with an important advantage: You can run separate Windows sessions for each application, thereby protecting the applications from each other.

Installation options

By far, the **dual boot** option is the easiest way to install OS/2 on a machine with DOS already on it. With it, you can always boot back into DOS to run your applications. You can't take advantage of OS/2's High Performance File System (HPFS). Applications that might make this an attractive alternative include any networks that don't have full OS/2 support (Novell and Banyan do have fairly stable drivers and applications), DOS applications that use DOS extenders, and some applications that require certain drivers or devices.

Installation options

Dual Boot:

- > FAT only
- > Installs right on top of existing OS without replacing it.

Boot Manager:

- > Requires reformatting hard disk
- > Can have HPFS and FAT
- > Can have different OSs: OS/2, DOS, & other

A lot will depend on your system. For instance, OS/2 supports a good variety of CD-ROM drives, but if your system has one that is not supported (usually a proprietary non-SCSI driver), then you'll want to boot into DOS to use your CD-ROM drive. (There are ways to use CD-ROM drives that are not supported by OS/2: boot into a real DOS session under OS/2 where the DOS CD-ROM drivers are loaded. This will give you access to that drive, and all other drives on your machine in that DOS session, but no other session or application on the machine will be able to access the CD-ROM drive.)

Boot Manager is a little more civil way to manage DOS and OS/2 on the same machine. In addition, Boot Manager will handle more than two operating systems. You could have SCO Unix, DOS, and OS/2 on one machine. This installation allows you to use HPFS, which can speed access on large volumes. Finally, it is easier to shut down properly than the dual boot allows. That is, it is easy to boot from OS/2 to DOS without shutting down properly under dual boot, and it is a little easier to do it properly

under Boot Manager.

Boot Manager gives you a choice of operating systems when you boot up. You can design it to default to one after a certain length of time, or you can design it to require user input.

The importance of SHUTTING DOWN

The preferred way to leave OS/2, whether to turn off the machine or to switch to another operating system, is to **SHUT DOWN**. What shutting down does is saves your desktop, and finishes writing from cache to file if you have lazy caching in effect. The real danger in powering down without doing the shut down sequence is that data files might end up corrupted.

In addition, shutting down will preserve changes you have made to your desktop.

It is more difficult to install. You **HAVE** to run FDISK and create a small boot manager partition. If you want DOS as an alternative, you have to create a primary partition for DOS. OS/2 then would be installed on a logical or separate drive. Because you are creating partitions, this installation is harder to install on a machine that already has applications, data, and an operating system on it. Be sure to back up your data before you install the Boot Manager!

Installing OS/2 as the only operating system might seem radical, but given the good record of running DOS and Windows applications, and considering applications that might be run on dedicated OS/2 machines, it's not that unusual.

There are other considerations when installing OS/2, not the least of which is what file system to install. OS/2 will use either the File Allocation Table (FAT), as well as it's own improved file system, the High Performance File System (HPFS). FAT is old, originally designed for use on small volumes. It is prone to fragmentation and is relatively slow. With later versions of DOS, it has been made to support larger volumes however. HPFS on the other hand, is designed to be more efficient (faster) with larger volumes, and be less prone to fragmentation. So if you have large hard disks (500 Mb or more) and your applications open many temporary files, you might

consider HPFS. But there is also the issue of compatibility. Unless you have a backup system that supports HPFS, you should think twice about it. And if you have any need to boot to native DOS and access the information on the HPFS volume, you should consider FAT.

Characteristics of OS/2's Object Oriented Environment

An object is something that you work with to perform a task. There are objects for data files, such as documents, data sets, and programs. Program objects represent executable files on your system. Folders are analogous to subdirectories on your disk. In addition, there are various objects representing devices, such as printers, disk drives, etc.

Data-file object	Document, data, or program.
Folder	Container for files: directory.
Program object	Executable file.
Device object	Printer, disk drive, etc.

All the objects are represented by icons and can be opened or run with the double click of the left mouse button. Objects can be associated with one another. For instance, a data file object that represents a SAS program can be associated with the SAS executable program object so that double clicking on the data file object will invoke the SAS System.

SHADOWS are links between duplicate objects. Whereas a **COPY** of an object does not exchange data between the original and the copy, a shadow does. Shadows are very helpful in organizing your desktop. Instead of actually moving objects from their logical folders, make a shadow and place it wherever you want it.

Each object has its own **settings notebook** that is accessed by clicking the right mouse button to bring up the pop-up menu, then using the left mouse button to select the **ARROW** on the **OPEN** line, then clicking on **SETTINGS** with the left mouse button. The settings notebook for program objects differs from that of data objects, and so on. Each type of object has its own set of attributes, most of which can be controlled by the user.

Multitasking

OS/2 also offers a robust platform that allows **multitasking**, or the ability to have more than one active application at one time.

On the surface, there seems to be little difference between the ability to run several applications at once under OS/2 and the same ability under Windows. But there is a difference. Multitasking under Windows is termed "Cooperative." What that means is that the system relies on the applications being designed to run simultaneously, thereby cooperating and sharing resources. We all know there are many applications that are being used that have no such cooperation designed into them. Consequently, under Windows one sometimes encounters "General Protection Faults", which occur when two applications conflict over a given system resource.

OS/2 offers true multitasking, termed "Preemptive." The operating system itself allocates system resources to applications, thereby resolving any potential conflicts and making the system operate more smoothly.

Multitasking may not seem like much if you are not accustomed to it, but once you use it, you don't know how you ever worked without it. There are times when you really only use one application at a time, but there are plenty of times when you can take advantage of the ability to have more than one application open at a time, and after that you find yourself running active applications in the background while you have other applications open at the same time. It can be somewhat baffling, particularly if your separate applications are set up to run in full screen. You may not realize you still have other applications open. I suggest users find a font size and window size that allows them to display several windows on the desktop at the same time. This also facilitates in the use of the clipboard, i.e., the ability to cut text from one application and paste it into another.

OS/2 References

Shrink-wrapped OS/2 comes with two manuals, an installation guide and "*Using the Operating System*." It is worthwhile having them. In addition, you can install OS/2 with on-line help in the form of a *Tutorial*, *Command Reference*, *Glossary*, and a *Master Help Index*. These online resources take a little getting used to -- the old paradigm of a well worn, dog-eared manual that you can have open on your desk as you work on your computer is changed. But there are advantages to the on-line material: hypertext links, bookmarks, and instant access.

Besides the IBM documentation, there are several third-

party references:

Your OS/2 2.1 Consultant, by Herb Tyson, 1993, Sams Publishing, ISBN 0-672-30317-5

Inside OS/2 (Periodical), The Cobb Group, ISSN 1063-3146

OS/2 Professional (Periodical), I.F. Computer Media, Inc., ISSN 1069-6814

There are several electronic bulletin board services (BBS) devoted to OS/2 and there are some Internet discussions devoted to OS/2:

OS2-L@HEARN -- Unedited discussion on OS/2;

OS2USERS@MCGILL1 -- OS/2 Users discussion;

UTOS2-L@UTKVM1 -- OS/2 discussion.

A very rich resource for OS/2 information, freeware and shareware is the Walnut Creek CD-ROM and FTP server. The CD-ROM offers some 600 megabytes of shareware and sells for about \$30 per disc, and the FTP server can be accessed at the address [FTP.CDROM.COM](ftp://FTP.CDROM.COM).

SAS

Performance and capacity -- One of the major attractions of the OS/2 system is the 32-bit processing speed that the operating system offers. Applications that are designed for 16-bit processing cannot take advantage of this, but the SAS System for OS/2, release 6.08 and since, is designed for 32-bit processing. Of course, performance is relative. If you are coming from mainframes or Unix environments, then you won't be particularly impressed with OS/2. On the other hand, if you are coming from the DOS or DOS/Windows world, you undoubtedly will see speed and capacity improvements.

Performance should be broken down into at least two components: Speed, which is most commonly connected with performance, and capacity, or the ability to handle large data sets, large numbers of variables, etc. Using the DOS version of SAS as a reference point, everyone will agree that it was a poor performer in terms of speed. The capacity of this version also was limited, although I did manage to produce some datasets with upwards of some 1,800 variables.

When considering speed, anything that can take advantage of 32-bit processing will be faster than anything that can't. Therefore, OS/2 offers faster processing. But for applications that perform a lot of reading and writing, the determining factor becomes disk speed and bus speed. EISA systems and systems using 32-bit I/O controllers will help performance, but the actual disk speed will be a

slowest part of the system. For cpu-intensive tasks however, SAS running under OS/2 will perform much better than its DOS and Windows counterparts.

Capacity is another strength of SAS under OS/2. You'll find The SAS System for OS/2 has a great deal more capacity than its DOS counterparts, releases 6.03 and 6.04. Since there is no 640K memory limit to the operating system, as there is with DOS, programs can use data sets with many more variables than they could in the DOS version. Variables are defined in the Program Data Vector (PDV), which is held in conventional memory in the DOS versions. Because OS/2 addresses as much memory as is on the system, and will swap to disk if necessary, the PDV can be much larger.

The Look and Feel of SAS 6.10

If you are familiar with the display manager system, you'll find it somewhat familiar under Release 6.08 and 6.10. The 6.10 release has some significant refinements to the display manager system, and we'll focus on them for this discussion. **NOTE:** the discussion here is based on work with the beta release of 6.10 (May, 1994). While I am confident that nothing I describe here has been changed in the production release, you should be aware that I was not working with the production version.

When SAS is invoked with the display manager system in effect, the user sees several different parts. There is a **title bar** across the top, with a bar containing a series of commands that bring pull-down menus to the screen: **File, Edit, View, Globals, Options, Window, and Help**. The commands on this bar give you all the commands you need for most operations, and this bar is a major step towards a standard interface. The commands are context sensitive, that is, you can use the **File -> Open** command sequence to open a file if the active window is the program editor, but not if the active window is the log or the output window. Under the pull-down menu bar is a bar that contains two areas: a command area where you can issue command line type commands, and an area where the tool bar can be located, if so desired. Below this bar is the application workspace, containing by default the **program editor**, and the **log** and **output** windows. One major improvement found in these windows is in the title bar for each window, which reports the name of the file called in (to the program editor) or the name of the file the contents have been stored under. Other windows appear here too, as called upon by the user. Below these windows is one more bar, containing a message area, and a section that indicates the default directory. The message area contains messages such as how many lines were submitted, saved, etc.

The individual windows no longer have their own pull-down command bar in 6.10 -- these commands are all accessible through the pull-down menu bar at the top, or through the command line option, or through the popup menu option. The pull-down menu at the top is context-sensitive, that is, it displays only the options available for the window that is currently active.

To run a program in the default display manager system, you need to either enter the program into the program editor, or call in a stored program. Calling the program is done by clicking the left mouse button on the **File** pull-down menu on the top menu bar. Using the left mouse button, choose **FILE**, **OPEN** and **READ FILE**. A dialog box opens that allows you to move from drive to drive, from subdirectory to subdirectory, giving you a list of all available files. By default, the system searches for programs with the .SAS extension, but you can change that to your liking if your SAS program names don't have that ending. When you see the program you want, either highlight it and click on OK, or point to it and double click on it. This brings the code into the program editor.

Next, when you have made any changes you need to the program, submit it by again clicking the left mouse button on the **Locals** pulldown menu item, then point to **SUBMIT** and choose it with a click of the left button.

You may use the same method to call up the **LOCALS** menu to **recall** and resubmit your SAS program. To save your work, use the click on the **File** pull-down menu and choose **SAVE** or **SAVE AS**.

The same actions can be accomplished with either the command line, or the popup menus, which are activated by clicking the right mouse button on the text area of one of the windows.

Alternatives to the SAS Display Manager System

You might prefer not to use the Display Manager system. Perhaps you want to conserve system resources -- the DMS does take some resources. Perhaps you simply prefer a batch approach, letting your programs run in the background while you get on to other tasks. Be sure you check your logs!!

There are three ways to launch your programs in a batch mode: Command Line; Click on data-file object; and Drag-and-drop. In each case, a dialog box appears to inform you what program is running, and where the log and output files will be when the job ends. The dialog box also has two radio buttons allowing you to minimize the process into the minimized viewer or to cancel the job. Any DMS

command in your programs will generate warnings.

Double Clicking a data-file object to launch program:

Define default action in data-file object settings:

Open settings notebook for program file object

Click on "Menu" tab

Click on "~Open" in "Available menus"

Click on "Settings"

Click on Popup for "Default action",

Select SAS

NOTE: If SAS is not listed there, you can click on "Create another" while "Primary pop-up menu" is highlighted under "Available menus", and you can add an action to take by naming it and specifying the fully qualified data set name.

You can launch your programs from the **command line**. Open an OS/2 command line window, switch to the directory where SAS is installed, and issue the command "**SAS program_name**". The command line session disappears until the program terminates. It is freed when the task completes. Note that you can have multiple command lines.

Double clicking on the data file object provides an object-oriented method of launching your SAS program. It requires that your data file object have the correct default action specified in the settings notebook. This setting should also assign the SAS icon to the data file object.

Drag-and-drop¹ is another object-oriented method of launching applications in batch mode. Using this technique, you simply point to the data-file object which contains the SAS program, press and hold the right-mouse button, and drag the object over to the program-file object

¹ You can also use the drag-and-drop technique to launch a program into the display manager. Here's how: size the SAS AWS to less than full screen, and open a folder with the program you want to run. Click the right mouse button on the data file object, and drag it onto the text area of the program editor window, then release the mouse button, dropping it on the program area. Immediately there will be a message that one line was submitted, and if you observe the log, you'll see that SAS is treating your program with an "%INCLUDE" statement. The log will show the results of your code, but you won't see the source code. Recalling your text will recall only the one line submitted.

(where the SAS executable is found), then release the mouse button, thereby "dropping" the SAS program on the SAS system. This method requires the string "%*" (inclusive of the double quotes) in the parameter line of the settings notebook of the program object.

Running Multiple SAS sessions

One of the things that you can do with SAS under OS/2 that you can't under Windows is run simultaneous sessions. Of course, the more you ask your cpu to do for you the slower performance will be in general for anything you do. The more performance is important to you, the less you'll want to multitask. But even so, sometimes multitasking can increase productivity even when it hurts performance, for instance, by allowing you to work on something else while a task runs in the background. Sometimes both those tasks involve the use of SAS.

We just discussed how to run SAS in the background batch style.

You can certainly launch several programs to run at the same time. You can also have more than one SAS display manager session running at the same time. The key to running more than one session from the same SAS program object is to specify that the system is to open a new window (session) when the object is chosen again. See the box below on how to do this.

Multiple sessions from one object:

Right mouse button click on SAS object
Left mouse button click on ARROW on OPEN line
Left mouse button click on settings
Click on Window tab
Push "Create new window" radio button
Double click on upper left corner to close the settings notebook

Only the first session will use the PROFILE.SC2 catalog that customizes the user's application work space (AWS). The trick to running more than one session from your SAS object is to tell the operating system what to do when you click on the object again. By default, the system will go to the open object if you attempt to open the object again. Open the setting notebook for the program object (SAS.EXE) and select the Window tab, then the "Create new window" button. This setting is possible for almost any application you can run under OS/2, unless the

application itself limits the number of sessions.

How to run multiple SAS sessions from different objects

You can have copies of your SAS object which each run the same code, but differ only by having different PROFILE.SC2 catalogs associated with them. They can then be configured separately, or their configurations could be copied and used by other sessions.

Why would you want to do this? Perhaps you want the option of using the different DMS options (pop-up menus, or command line, or menu bar). You might have some products running off an external device, and some products loaded on your local disk, and you want to have both configurations without having to edit your config.sas. Maybe you share your PC with someone else with completely different tastes and preferences regarding the AWS. Running from separate objects allows you to do this.

To run different sessions from different objects, you need to first copy the object and create a new config.sas. Specify the new config.sas in the settings notebook of the new object. The new config.sas can exist in the same directory as the old one, but under a different name, or it can reside in a completely different subdirectory. Refer to the box above for more details.

Specify Config.sas for specific configuration:

Copy config.sas to separate file
Edit file, specify different subdirectory for SASUSER. It need not exist (yet) -- SAS will create it if it does not find it.
Create a new object for the SAS executable (Copy the existing object... don't do shadows here !!!).

Open Settings notebook: On the Program page, under Parameters, specify the desired config.sas. Example of a parameters line that points to a different config.sas and allows drag-and-drop:

```
"%*" -config d:\sas\comline.sas
```

Remember, running separate SAS sessions does not require taking the above steps, as long as the object has the

"open new window" property described above. The first session will use the profile as set up, displaying any customized features of the application work space. Any subsequent sessions use a default configuration, which cannot be changed and saved.

SASWORK

SAS under OS/2 allows you to run multiple sessions, each with its own SASWORK area. Unlike the SASUSER notes above, SAS automatically sets up a unique SASWORK area for each session, so that temporary files do not get confused between sessions.

For instance, if I have three separate SAS sessions running, I can look on my disk and find in my SASWORK subdirectory three more subdirectories, one for each session.

Running SAS from multiple drives

Let's face it: the SAS system is huge! If you loaded the entire suite of products, you'd need upwards of 160 megabytes! And most machines I've encountered have lots of other applications on them. Squeezing SAS onto some machines offers quite a challenge.

You can run SAS over a network. Individual machines might have their own SASWORK and SASUSER areas defined, along with their own config.sas and profile.

I have not been entirely pleased with performance for the SAS System when it is run over a network. I'll admit, I did not devote a lot of time and energy to this. I saw it basically as a bandwidth problem: large modules coming over the network. The performance hit could have also have had to do with the network server. But it can also be viewed as a sacrifice of performance for disk space.

One thing that does work is to run SAS from both the local drive and an external drive. Frequently used products, such as the CORE and BASE portions, can be loaded on the local drive, while less frequently used products could be loaded on a network drive or a CD-ROM drive. The performance hit from coming over the network or from a relatively slow CD-ROM drive is reduced by the fact that calls for these modules are made less frequently than calls for data step processing. Just what to load where would best be determined on a case-by-case basis, based on what is being done.

Just how do you go about loading the SAS System on multiple drives? Two things can be done. First, at

installation, the drives can be chosen. Installation then might need to be done in a two-phase operation, installing some of the products on a local drive, then later installing other products on a network drive.

What about CD-ROM? As a practical storage and retrieval medium, the standard optical discs have been criticized for being too slow to run applications from. But witness the growth of the CD-ROM field. Titles abound! And with the introduction of double, triple and quad speed CD-ROM drives, this medium starts looking more attractive for applications and for distribution. Industry leaders like IBM, Microsoft, WordPerfect and SAS Institute offer their products on CD-ROM now. Not only is it more efficient to produce and distribute products on CD-ROM, but it is so much more easy to install from the optical disc.

The SAS System, unlike OS/2, can also be run from the CD-ROM. In fact, you can run the entire SAS System from the optical disc. Not that you'd necessarily want to -- here's where the multiple drives comes in and here's where I learned a trick. If you've ever inspected your config.sas, you no doubt noticed a forbidding warning at a certain point for the user not to edit the file below a certain point. The trick to getting SAS to run certain products off of CD-ROM when the CORE (and other products) are installed on the local drive is to go ahead and edit below that line. Go ahead! Challenge authority! Cross against the light! Edit your config.sas below the NO EDIT line! (But be careful. I checked with tech support on this, and they pretty much said to go ahead, but take care to match the pattern that is already there. I'd make a backup copy of the file first, and make sure that I keep the file in the same structure as when I started.)

Basically, all you're changing is where the system looks for certain products, and that is done by first creating an environment variable in the top section (where you CAN edit the file), then using that environment variable to help the system find the products. I recommend using a config.sas from a full install, so that all the products will be listed properly. Specify your environment variable, then refer to it in the section you're warned against editing.

In the example in the box (next page) I have taken portions of the config.sas which accesses SAS/GRAPH and MAPS from CD-ROM (drive e:) to show how the environment variables are set up and then an example of how they are referred to. Note that the same sorts of changes can be made to run some products off a local area network without actually doing an install.

Standardizing The Look and Feel of SAS

As we've seen, SAS (and OS/2) are highly configurable, allowing the individual user to choose the environment that is most productive. Not only can a single user configure SAS in several different ways on their machine, you'd expect to find different configurations on different machines as well.

The ability to run separately configured SAS sessions extends beyond the realm of multiple simultaneous sessions. At my agency, we have a production oriented shop in which it is not uncommon for any user to sit down at any machine. In order to be productive, they expect SAS to have the same look and feel from one machine to another. While colors and window position are not critical and in fact are quite superficial, things like menu type and F-key programming is necessary to standardize in this type of situation.

SAS from local disk and CD-ROM:

Portions of CONFIG.SAS

At top:

```
/* SAS environment variable for CD-ROM */
-set cdrom e:\sas

/* SASROOT PATH setting */
-SET SASROOT d:\sas608

...stuff deleted...

/* SAS System load image search paths */
-PATH (
    !sasroot\core\sasexe
    !SASROOT\STAT\SASEXE
    !SASROOT\SHARE\SASEXE
    !SASROOT\INSIGHT\SASEXE
    !CDROM\GRAPH\SASEXE
    !SASROOT\FSPS\SASEXE
    !SASROOT\BASE\SASEXE

...stuff deleted...

/* SAS System Maps Library Definition */
-maps !cdrom\maps
```

What's a manager to do? Put their foot down, lay down the law? Make and enforce a standard?

Ironically, we found that the very thing that allows the

personal choice in system configuration also makes a compromise possible. Without anyone being authoritative, the office in question allowed each user to have their own configuration as a separate object, but also insisted that they also maintain a standard SAS configuration as well. A standard can be established, but not one that overwrites personal freedom in choosing the most productive configuration for each individual. You can have your cake and eat it too!

Working with the SAS profile catalog entries for a particular platform, one finds that it is always easy to get back to the default settings: simply delete the appropriate PROFILE.SC2. That file can be found in the appropriate \SASUSER area, and that area in turn can easily be found by either inspecting the config.sas file, or by opening the LIB window under SAS.

To copy a particular profile, either use PROC COPY or just copy the profile catalog PROFILE.SC2 from the SASUSER area to another storage device: diskette, lan file server, etc.

Toolbox

One welcome addition to the Display Manager System is the toolbox. It is an option -- you don't have to have it unless you want it. It can be a floating toolbox, or it can be a toolbar in a fixed location. The default toolbox offers the following features, all in a graphical presentation: **Submit contents of program editor, Open file, Save file, Print, Cut, Copy, Paste, Undo, Set default directory, Lib, Command prompt, and Help.**

These are all commands that are available through the command set and/or the Display Manager System. The toolbox makes these commands a little more accessible to the user.

The toolbox under 6.10 is very configurable, allowing the user to add tools and select from a range of icons. You can design your own commands and have access to them with the click of your mouse button.

Finally, the System command prompt seems to me superfluous. Bear with me, but I thought we were running in a multitasking environment. That means that anytime I need to address the system to run a program, check the status of a file, or whatever, all I have to do is switch from OS/2 in the foreground to open an OS/2 command prompt window. What is more, I can do this while SAS is running some data steps and proc steps.

SAS Help

The SAS System has a fairly extensive help system available online, with several points of entry. The system offers an online facility to describe features and procedures, including syntax and hypertext links. The facility is useful, but it does not offer bookmarks and references to written documentation.

There are several ways to get into SAS Help. One way is to use the HELP menu bar choice on the main menu bar (described above). Another way is through the toolbox, and yet another way is the command line.

Help was been improved in 6.10 and is worth familiarizing yourself with, so you know some of the resources that are available.

As I pointed out before, there are no references to written documentation. That would be a useful enhancement, allowing a user to get the basic information online, but pointing them to the manuals for further information. More examples could be useful in the online documentation also.

A design shortcoming reveals itself when you change your user interface preference from POP-UP or MENU BAR to the COMMAND LINE. Where the help system was designed for "point-and-click" with the GOBACK, EXITHELP and HELP buttons, these are not available with the command line. You can still use the pointer to select items, but you must issue the appropriate commands to move about in the help system. Seems a little inconsistent, and it can certainly be annoying. There is a work-around: add a tool to the toolbox that sends the cursor home and exits from the current screen with the END command. Works for closing other windows too, even when the command line option is not in effect!

CONCLUSION

So if OS/2 is so good, why aren't more people running it? It does present a bit more of a challenge to install than does Windows 3.1, and that, coupled with the prevalence of Windows in the industry, makes the OS/2 users mavericks. In addition there seems to be a certain momentum to the applications in the "Windows world." For instance, the implementation of ODBC provides some additional attractions to running SAS in a Windows environment.

The bottom line for many of us has been that OS/2 offers a mature 32-bit system that delivers performance, capacity, and stability NOW, and has for several years.

The adventurous users who want to get more out of their systems often have been the ones looking at alternative operating systems, but at the expense of having to provide their own support. Even if it does cost something, there are tangible benefits to it. Running SAS under OS/2 provides 32-bit processing and preemptive multitasking, an object-oriented user interface, and a great deal of flexibility. User preferences both at the operating system level and the SAS system let the individual user shape the workspace to enhance their own productivity. Running DOS applications in a multitasking environment changes the way you work. Running Windows applications in separate protected sessions frees you from worrying about General Protection Faults that would bring the entire system down under Windows.

Where will the future lead us? NT? Chicago? OS/2? The computer industry, and the bewildering array of operating systems is constantly changing. But those who want to take advantage of some truly empowering technology can run SAS under OS/2 right now.

References

Recommended for further reading and reference is SAS Institute Inc., *SAS® Companion for the OS/2® Environment, Version 6, Second Edition*, Cary, NC: SAS Institute Inc., 1993. 318 pp.

Acknowledgments

SAS® is a registered trademark or trademark of SAS Institute Inc. in the USA and in other countries. IBM and OS/2 are registered trademarks or trademarks of International Business Machines Corporation. ® indicates USA registration. Other brand or product names are registered trademarks of their respective companies.

For more information about this paper, contact the author:

Andy Parks
USITC
500 E St SW
Washington DC 20436
202-205-3250
andyp@digex.com