MAKING YOUR SAS® SOFTWARE SKILLS PORTABLE

Kirk Paul Lafler, Software Intelligence Corporation
Marjorie R. Bedinger, Health Care Financing Administration

ABSTRACT

Over the years the SAS® System has evolved from a single environment to a multi-environment, multi-version system. Consequently, major new advancements have been introduced to simplify and improve the way we interact with the software. It may be a sign of the times, but our desire to access information residing on a variety of environments is on the rise. The level of consistency between the different environments and versions is extremely high. Still, confusion exists, especially since some statements, commands, and/or conventions may not work when transported between environments. This paper presents areas causing concern when transferring skills between a multi-environment, multi-version SAS System. In addition, guidelines are presented to minimize this confusion.

INTRODUCTION

This paper discusses some of the stumbling blocks when trying to port one's SAS software skills between the environments of mainframe OS, Release 5.18 and micro, Release 6.03. It is particularly oriented to those users who will be constantly popping back and forth between the two worlds and linking the two worlds together via the SAS Micro-to-Host link.

The level of code consistency between the various releases and environments is high enough that an individual migrating from one system to another has a relatively easy time picking up on the differences and continuing. The confusion arises for the user working in and/or attempting to run identical programs in multiple environments and for the technical support staff who try to be "fluent" in multiple environments simultaneously.

We decided we could best discuss the issues involved in making your SAS software skills portable by taking a functional approach, beginning with SAS invocation and the interactive environment in the form of Display Manager, then moving on to the program statements in the form of Data step, Procedure step, and Use Anywhere statements. Finally, we will address linking the environments with the Micro-to-Host link. Due to time and space limitations, this presentation will cover only the primary elements involved.

If you are currently a mainframe user submitting all your SAS programs in batch, you may find that after you become familiar with PC SAS, you will want to use SAS interactively on your mainframe. Some of the information which we present, while being unfamiliar to you now, may save frustration later on.

INVOKING SAS

In the beginning there was the invoking of SAS. Under OS this is usually accomplished by keying SAS with any special options. On the PC the same method is used. However, the way you specify the options differs.

For OS 5.18 you specify:

```
SAS OPTIONS('option1 option2 . . .')
```

For PC 6.03 you key:

```
SAS -option1 -option2 . . .
```

Also, as you might expect, there are also differences in the options you can specify. These will be discussed below.

PRIMARY WINDOWS

The interactive Display Manager session appears. Now you are going to see either one more or one less window than you are accustomed to seeing, depending on which environment is new to you. OS SAS Display Manager contains two visible primary windows--Log and Program Editor--with the third primary window--Output--hidden behind these two. The Output window comes to the forefront automatically when a procedure is executed or it can be called forward on command.

In PC SAS one sees all three primary windows at once, with Output on top, Log in the middle, and Program Editor on the bottom. Any one of them is really too small to see or do very much. You can redefine the shape and placement of these windows in various ways by using the WGROW, WSHRINK, WMOVE, and WDEF commands and save the new 'look' with WSAVE.

DEFAULT FUNCTION KEYS

Perhaps the biggest shock when you test drive whichever version is new to you is that your function keys don't work right! The default keys in one version bear no relationship to the default keys in the other. Also, on the PC you can combine the Shift, Alt, and Ctrl keys with the basic function keys to create additional user-assignable key combinations (not to mention combining Alt and Ctrl with 26 letters of the alphabet).

Your SAS site representative or consultant could specifically configure the basic function keys to match across versions. The drawback to reconfiguring one system to match the other as a site-wide default is that documentation, templates, and classes all use the SAS Institute-defined defaults. This could lead to more confusion for new users.

Someone who is adept at one operating system or the other will probably either change the default function keys or start keying all commands and quit using function keys altogether. When you change your own key assignments, the information is stored in your profile catalog. At Health Care Financing a profile is automatically created and retained the first time a person uses OS SAS interactively so most of our users never even realize they have a profile.

In OS 5.18 each window--Log, Program Editor, and Output--has its own set of key definitions. In 6.03 all windows share the same set of key definitions.

For technical support folks, the location of the Display Manager default keys differs on the mainframe and on the PC. Under OS, they are found in SASUTL.SAS while on the PC they are found in \SAS\SASHELP\CORE.SCT. There is no SASUTL directory on the PC.

KEYBOARD CHARACTER DIFFERENCES

Keyboard character differences are an area where we frequently receive calls from a new user. Vertical bars, broken bars, logical not symbols, and square brackets are all characters present on one but not the other keyboard.

	3270 Keyboard	PC Keyboard
Logical Not	¬	^
Vertical Bar	|	¦ or |
Brackets	N/A	[ ]

Fortunately, these differences do not present any significant problems with the interface between versions OS 5.18 and PC 6.03. We have submitted code from the PC to execute on the mainframe via the Micro-to-Host link. We have moved code via Proc UPLOAD and also used flat file transfer directly with terminal emulation software.

The only difficulty we have had is with the brackets - [ ]. They translate to the EBCDIC brackets, but executing the code on the mainframe gives a SAS system error. The broken bar translates to a vertical bar, and the caret translates to the logical not symbol.

If, for some reason, you do experience difficulties, we make the following suggestions. Use NE for Not Equal instead of using the symbols ^= . Use parentheses () instead of braces. Definitely do not use brackets. You could also use the CHARCODE option which allows you to substitute a two-character string for the symbol giving you problems. For example:

Logical Not	?=
Vertical Bar	?/

Both operating systems will recognize these characters and you should have no communication difficulties.

SPECIAL FILES

SASEXEC/AUTOEXEC.SAS - Under OS a file allocated with a DDname of SASEXEC can contain a multitude of Display Manager command line commands which are automatically executed when SAS is invoked. On the PC a file named AUTOEXEC.SAS performs a similar function. If AUTOEXEC.SAS is found in the directory from which the SAS System is invoked the commands will be automatically executed. However, in Release 6.03 these are NOT Display Manager command line commands. They ARE regular program statements. To issue a Display Manager command you must preface it with the letters DM.

Personally, we prefer the 6.03 approach because it enables you to put both Display Manager commands and program statements in the same file. In OS 5.18 you must use the INCLUDE command in the SASEXEC file to reference a second file name which contains the actual program statements. Then you must issue a SUBMIT command to cause the execution of the program statements. This requires keeping track of two files and confuses novice users.

CONFIG.SAS - The PC SAS System requires that the user take more responsibility for the product. One of the ways that you do this is with the CONFIG.SAS file. Under MVS the site rep sets all the options under which the product normally operates at your site when he or she installs the product. To some extent this can be done on the PC by having the site representative provide the user with a company-standard configuration file.

However, a PC environment within a company can be so variable that it is almost imperative that the user be familiar with the configuration file.

For example, this file specifies on which drive and directory to find the SAS System executable files. This can easily vary from machine to machine depending on its configuration with hard drives, Bernoulli boxes, etc. This file also contains information to the SAS System about the type of monitor being used. If this is not specified properly with the -FSDEVICE option, you may not see anything but a cursor on your screen once the SAS software is loaded.

While we are on the subject of user responsibility with PC SAS, we might as well tell you that it is also your responsibility to keep the product running when your new license year begins. On the mainframe the site rep applies the new dates and codes to keep the products operational, but on the PC you must apply the dates and codes yourself using the -SETINIT option.

GLOBAL OPTIONS

Let's talk about options in a little more detail. Certain options that you may customarily use could also cause consternation as you move between environments. How and where you specify these options might also cause you some confusion.

In both environments there are certain options which can only be specified as the SAS System is being loaded. These are called configuration options. We have already talked about this type in our discussion of CONFIG.SAS. In both environments these options can be pre-configured or you can specify them when you invoke the SAS System as we showed at the beginning of our presentation.

The second type of options are called system options. It is this group that will probably give you the most trouble, particularly if you are already a whiz with Release 6.03. The PC release allows you to specify system options in four places: the user profile catalog, the SAS command, an Options statement, or the Display Manager Options Window. OS 5.18 allows only for the SAS command and the Options statement.

There are two options in particular which we believe will give you fits if you are constantly switching back and forth between interactive environments. Under OS you specify the linesize and pagesize you desire in your interactive Display Manager session by using OPTIONS TLS= and TPS= for terminal linesize and terminal pagesize. To reset them to the defaults, you can use TLS=0 and TPS=0. Release 6.03 does not recognize TLS and TPS and will tell you there is no such option. It requires that you use OPTIONS LS= and PS= as you would under OS batch.

The second one is the PAGENO option which allows you to reset output page numbering in Release 6.03. This option does not exist in OS 5.18.

BASIC DISPLAY MANAGER COMMANDS

Probably the most frequently issued Display Manager commands are those that deal with including and saving code and printing the contents of the windows and also popping out to TSO/DOS for miscellaneous reasons. A feature of the PC version which makes it easier for new computer users in this respect is the ability to directly reference the PC file name you are using; such as, 'a:prog1.sas' wherever a fileref is required. Under OS you MUST use the TSO ALLOCATE command to link a fileref to the name. While you can also use a fileref on the PC, you don't have to do so.

Before we start discussing actual working commands, we want to take a minute to talk about a very important aspect of any software package -- HELP! This is also an area where Releases 6.03 and 5.18 differ. In OS 5.18 you can key HELP and a keyword on the command line or submit HELP keyword; as a program statement. In 6.03 the HELP program statement is not available. You can either key HELP and a keyword on the command line or just key the word HELP to open a Help Window from which you proceed to select the topic you want. On-line help frequently reveals interesting and otherwise undocumented information.

CLEAR - The CLEAR command is used in both operating systems to empty a window of its contents. In OS you can use the commands CLEAR OUTPUT, CLEAR LOG, or CLEAR PGM to empty the selected window from the active window or you can execute CLEAR from within the window you want to clear. (Exception: You cannot clear the Output window when you are in that window.) In Release 6.03 the parameters on the CLEAR command are not accepted. You must be in the window you want to clear and issue the command from there.

The alternative to actually keying the commands to move to the window you wish to clear, clearing it, and then moving back to where you were is to assign the command sequence to a key combination; for example,

CTRL L ===> LOG;CLEAR;PGM

This works just fine -- as long as you can remember all the key assignments. We hope SAS Institute will make these parameters a part of the next Version 6 release.

INCLUDE - This command is used to bring the contents of an external file into the Program Editor window, both the OS and PC SAS Systems. As stated above, on the PC you can directly reference the PC file name you wish to INCLUDE, while on the mainframe you must have previously allocated the dataset name to a fileref. We will discuss this command again in conjunction with the FILE command.

PRINT/SAVE/FILE - Now we come to the various ways to save and print the contents of your Display Manager windows. In OS 5.18 you can use the PRINT command to send the contents of a window directly to a system printer or to a fileref. In addition, you can specify which window you want printed. (A number of other options are also available.) In 6.03 the PRINT command is not recognized.

Under OS, you can SAVE the contents of any window to an external file by specifying SAVE [windowname] fileref. Once again this is a command the PC version does not recognize.

If you PRINT the contents of the Output Window to a fileref, the carriage control characters will be placed in the file. If you SAVE the contents of the Output Window, carriage control characters are not included. The SAVE command has a MOD option which allows you to append additional information to the end of a physical sequential file. The PRINT command always replaces the existing file.

Then there is the FILE command. This command is not documented for OS 5.18, and there is no information about it under the OS SAS Help facility. However, it is there; and it does work. On the PC you must use the FILE command to place the contents of a window in a file or send directly to a printer.

Under OS the FILE command appears to work the same as the SAVE command; that is, you cannot use it to send a window's contents to a printer; it does not place carriage control characters in a file; it does accept the MOD option.

On the PC you can use the command FILE 'PRN' to direct a window's contents to the printer. If you direct the contents of the Output window to a file, the carriage control characters are placed in the file. The FILE command does not have the MOD option. You will always overwrite an existing file.

As with the CLEAR command discussed above, OS 5.18 allows you to specify which window you want to PRINT or SAVE or FILE. In 6.03 you must be in the particular window or set up a key combination to accomplish the task.

All of this leads us next to a discussion of the interaction between the INCLUDE and FILE (PRINT/SAVE) commands. On the PC when you do an INCLUDE with a fileref (or actual file name), a subsequent FILE command with no fileref specified will automatically replace the file you included. Under OS 5.18 this is not the case. The automatic replacement does not occur UNLESS you specify the REP option on the INCLUDE command. Normally you would have to specify a fileref on the FILE command.

As if all of this were not enough, there is a further complication between OS and PC when using INCLUDE and FILE/PRINT/SAVE commands. This complication deals with the fact that under OS you can work with two types of datasets--physical sequential and partitioned. The PC has only the equivalent of the OS physical sequential. The discussion up to this point has dealt with the use of these commands with physical sequential files. The following deals specifically with OS partitioned datasets.

You cannot specify the MOD option on the SAVE or FILE commands. If you do not use the REP option on your INCLUDE command, you will need to specify both a fileref and the REP option on your FILE or SAVE command. However, this does not hold true for the PRINT command. The PRINT command will overwrite a member of a partitioned dataset without the REP option being present.

The moral of this story is: Be Careful. You might accidentally overwrite your file.

NEXT - On the PC you can hop around from window to window to window by using the NEXT command. NEXT is not recognized by the OS version.

SPLIT/ZOOM - Under OS, in order to change the positioning of the top of your Program Editor window, you can position the cursor at the location you wish on your screen and press the SPLIT function key (or issue the SPLIT command). You cannot totally eliminate the other window, however. There is no SPLIT command on the PC. The closest you can come is to use the ZOOM command assigned to a function key to blow the window up to full screen. You can use Window Grow and Window Shrink commands, but we have not found that to be as satisfactory as a quick ZOOM.

SUBTOP - On the PC SUBTOP is a handy command. If you are keying in some code and decide you should have designated a libref or fileref first, you can stick a LIBNAME or FILENAME statement in as the first line of your code and then SUBTOP it, that is, submit the top line. You cannot do that under OS.

X/TSO - Now you want to issue a TSO or DOS command. Under OS you can use the letter X or TSO. On the PC you must use the letter X. You cannot use DOS. This does not present much of a learning curve. Just use X in each environment you work in. When you leave the TSO or DOS environment to return to the SAS System, however, you can use END, RETURN, or EXIT under TSO while you must use EXIT under DOS.

While we only have time here to touch upon a couple of the major commands used, we encourage all of you mainframe users to investigate the wealth of new commands in the PC version, such as AUTOADD, AUTOFLOW, AUTOWRAP, the new COLOR commands, the CUT & PASTE commands and the special windows. For you PC users who are working up enough courage to tackle the mainframe, sorry, but you are going to lose all these neat features, at least for the duration of Version 5.

PROGRAM STATEMENTS

In general you need to be aware that there are some data step and procedure step statements that are available in one environment that are not available in the other. Also, the options for, and capabilities of, statements that exist under both operating systems may be different.

DATA STEP STATEMENTS

A couple of what we consider to be the most significant differences involving the DATA step are the ARRAY, LENGTH, RUN, and WHERE statements and the WHERE= data set option.

The ARRAY statement is an example of where the capabilities of a statement which is present in both versions are different--Release 6.03 allows for multidimensional explicitly subscripted arrays while OS 5.18 only allows for single dimension explicitly subscripted arrays. This has caused anguish to a number of people who developed programs on the PC and then tried to port them to OS.

Another noteworthy difference is in the LENGTH statement. Under OS the minimum length which can be specified for numeric variables is 2. On the PC the minimum length is 3.

The feature that makes the RUN statement on the PC interesting is its PGM= option. Using this option you are able to store compiled SAS code in a SAS dataset with a .SSP extension. You cannot store compiled code under OS 5.18. Also, this is an instance of where using the on-line Help facility of PC SAS is a real benefit because you will not find this feature documented in the SAS Language Guide for Personal Computers.

In Release 6.03 the dataset option WHERE= (used with the SET, MERGE and UPDATE statements) and the data step WHERE statement can be used to subset your data prior to observations being brought into the program data vector. WHERE is not available in OS 5.18.

As mentioned earlier, on the PC you can explicitly specify the location and name of any input and output files used in your INFILE and FILE statements; for example, 'A:PATIENTS.DAT'. This is not possible under OS. In that case you must use a file reference.

STATEMENTS USED ANYWHERE

The two statements of this class that we want to discuss are the DM and LIBNAME statements. We mentioned the DM statement briefly earlier when we talked about the SASEXEC/AUTOEXEC files. We feel that it is significant to mention again that you can mix program statements and Display Manager commands in a single input stream using this feature which is available on the PC.

The LIBNAME statement we believe deserves mention to draw your attention to the fact that it works the same way and performs the same action under OS and PC systems. It can be used in OS as an alternative to a TSO ALLOCATE statement or a JCL DD statement. However, an important consideration should be given to its use under OS. The LIBNAME statement performs dynamic allocation of a data set. This means that the

dataset you are requesting will not be allocated to your program until the LIBNAME statement is encountered whereas with the JCL DD statement the dataset is allocated prior to the invocation of SAS.

Obviously on a standalone PC you will not have contention with anyone else for the use of a dataset.

However, under OS if there are multiple users of a dataset and it is possible that one or more users (or your job) might require exclusive use of the dataset for updating purposes, use of the LIBNAME statement runs the risk that you might get part way through your job and then error out because the allocation cannot be performed. With the JCL DD statement, the system will wait for the dataset to become available for a certain length of time and notify the system operators of the condition. Your job will not start without being able to gain the access you require to the dataset. Under TSO you can also detect a failure of the allocation and take whatever steps you desire again prior to the execution of the SAS command.

From a systems resource point of view and assuming that access to the dataset is critical for a successful run, we believe it makes more sense to perform your allocations through the operating system prior to the initialization of SAS.

PROCEDURE STEP STATEMENTS

The WHERE statement mentioned above as a DATA step statement in Release 6.03 is also available as a PROC step statement and as such it can improve the efficiency of your code by often negating the necessity of using a data step at all. The WHERE statement performs subsetting of your incoming SAS dataset as it is being brought into the procedure. Again, this is a statement which we are certain you will want to use on the PC, and it is not available under OS 5.18.

With Release 6.03 the Institute created a new group of procedures called 'interactive procedures'. This term may be a bit confusing because any time we are running SAS using a method not termed 'batch', we call it 'interactive'. However, these procedures, which currently consist of CATALOG, DATASETS, and PLOT, execute in what is referred to as 'run groups'. In order to terminate these procedures, there is another new procedure statement known as QUIT.

Normally a RUN statement forces you to a DATA or PROCEDURE step boundary. But with these 'interactive procedures', RUN simply causes the execution of a 'run group'. The procedure then sits there and waits for you to either execute another 'run group' or to QUIT. Fortunately, performing another DATA or PROC step will also terminate these procedures. If you attempt to exit the SAS System using the BYE command while one of these procedures is executing, you will be asked if you are sure you want to terminate the SAS session, to which you will have to reply Y or N.

The only one of these procedures we want to discuss here is PROC DATASETS because it gets a large amount of use, and there are some significant differences between OS 5.18 and 6.03.

In both environments PROC DATASETS allows you to list, rename, and delete all types of members in a SAS data library. You can also alter variable names, labels, formats, and informats. However, under OS PROC DATASETS has two formats--full screen processing and 'no full screen processing', which is accessed by using the Proc statement option NOFS. On the PC there is no equivalent of the OS full screen processing mode. So you can't just scoot your cursor around the screen interactively, making the changes you want.

With release 6.03 you specify the statements you wish to use such as Modify, Format, Delete, etc., as you would with 5.18 NOFS processing. Still, there is a difference between this and 5.18 NOFS processing in that the dataset list which is produced does not show you how many observations the dataset contains or how much disk space it takes. We particularly miss having the number of observations associated with each dataset in the library since one then has to do a CONTENTS to obtain this one little bit of information.

PC Release 6.03 also differs in other ways from OS 5.18. It provides the capabilities of the APPEND, CONTENTS, and COPY procedures as statements within the DATASETS procedure.

ERROR MESSAGES

The last item we want to discuss related to program statements is error messages. Running identical programs in the two versions produces different error messages. Generally speaking, the error messages on the PC are more cryptic and do not give you as good an idea of what your problem is. Error messages on the PC do not have error code numbers, and there is no PROC SASMSG which lists the error codes and messages in numeric order. This lack of order would make it difficult to document the errors in a meaningful way.

MICRO-TO-HOST LINK

The aspects of the PC SAS System which makes it most interesting to many users are the Micro-to-Host link capability and the ability to convert files to and from Lotus 1-2-3® and dBase III®. The micro-to-host link is used to subset large mainframe SAS files and download the data to the PC. There the analysts and statisticians can make their choice of using PC SAS, 1-2-3, or dBase for further work.

We suggest that users of the Micro-to-Host link include in their AUTOEXEC.SAS file the basic statements necessary before the link can be started. These statements are:

```
FILENAME RLINK '\path\script file name';
OPTIONS REMOTE = device;
```

substituting their own script names and the device appropriate to their site.

Generally speaking, we recommend that people start up the communications link and logon to the mainframe themselves rather than have it done through the script since it is much easier to see when something is not working right this way. We teach the use of TTY.EXE provided by SAS Institute as a standard method of establishing the link since it is distributed as part of PC SAS and therefore a constant that everyone will have.

Procs UPLOAD and DOWNLOAD are procedures used with Micro-to-Host processing which we feel deserve attention. These procedures provide for transferring flat files and SAS datasets between environments. Earlier we stated that whenever a fileref was required for interactive SAS under OS, it was necessary to use the TSO Allocate statement to make this association. These two procedures now allow for dynamic allocation of datasets on the mainframe and, therefore, make the coding more like that which can be used on the PC. When used in this manner, dataset names must be fully qualified and enclosed in quotes.

The procedures can operate only on existing host datasets; that is, you cannot create a new dataset using this method. Another point to be aware of is that if you do not specify DCB information (LRECL, RECFM, etc.) on the host when you create the file, that is, you just give it a name and space, PROC UPLOAD will default to a record length of 260 and a record format of Variable Blocked.

CONCLUSION

This presentation has tried to convey some of the basic differences which are encountered as one moves between the environments of OS Release 5.18 and PC Release 6.03. The lesson to be learned is that although a high degree of consistency exists between the releases, you cannot assume that if it works in one environment, it will automatically work in the other! We hope that SAS Institute will work toward eliminating many of these disparities in future Version 6 releases.

Kirk Paul Lafler
Software Intelligence Corporation
P.O. Box 1390
Spring Valley, CA 92077-0220
619-670-SOFT

Marjorie R. Bedinger
Health Care Financing Administration
7131 Rutherford Road
Baltimore, Maryland 21207
301-966-5800