
Amos Shu, INC Research, Inc., King of Prussia, PA

ABSTRACT
Average SAS programmers should be familiar with the regular SAS programming environment regardless of the Windows, UNIX, or z/OS (OS/390) Mainframe operating environments. SAS Drug Development (SAS DD) is a relatively new SAS system that is designed especially for managing, analyzing, reporting, and reviewing clinical research information. SAS DD resides on the UNIX operating environment, which is maintained by SAS Institute in Cary, NC. Its SAS program code editor is provided within the Process Editor, which is a Java application. Performing SAS programming work on the Java powered interface is very different from the regular SAS environment. This paper discusses the differences in all programming aspects (such as macro facility, flexibility, stability, and time consumption) between the regular SAS programming environment and SAS DD.

INTRODUCTION
The pharmaceutical industry has high expectations for managing clinical research information content in accordance with government regulations, including 21 CFR Part 11, good industry practices, and sound business practices. SAS Institute has designed the SAS DD system especially to try to meet full compliance with the pharmaceutical clinical research standards, specifically in managing, analyzing, reporting and reviewing clinical research information [1]. SAS DD not only contains the data/files/documents repository functions, but also the SAS programming function. Unlike regular SAS, SAS DD uses a SAS program code editor containing a parameter table powered by a Java process application to perform SAS programming work. Programming behaviors in SAS DD differ significantly from conventional SAS programming.

This paper first discusses the differences in macro programming between the regular SAS programming environment and SAS DD, then in flexibility, stability, validation, and finally in time consumption. All programs presented in this paper were developed in PC SAS 8.2 in the Windows environment and SAS DD Process Editor 3.4.

THE MACRO FACILITY
The way to reference a macro variable is to precede the variable with an ampersand sign “&” in the regular SAS environment, i.e. “&xyz” is a macro variable. However, in SAS DD, a variable name preceded by an ampersand sign “&” is either a macro variable (such as &abc and &pgmpath in the following figure) or a substitution parameter (such as &reportname, &mysetup, &ds, &rows, and &vars) which is not, but behaves in a similar manner to a macro variable. For example, a period can end a macro variable reference but cannot do the same to a substitution parameter. Due to the use of substitution parameters and other reasons, SAS DD programs are not able to be run directly in regular PC SAS, or vice versa. Even though a substitution parameter can be ignored and treated as a standard SAS macro, it is really confusing to average SAS programmers. It is also difficult to tell that a variable preceded by an ampersand sign “&” is a macro variable or a substitution parameter in a program. The values of substitution parameters are set up using ‘point and click’ in the parameter panel. The total setup process of all substitution parameters in a program usually is tedious and requires substantial time to be completed. This greatly decreases the efficiency of programming and forces programmers to change their programming behavior. Almost all SAS programmers resist changing from the convenient traditional macro programming approach to the confusing and inconvenient substitution parameter programming method.
FLEXIBILITY

As mentioned above, the setup process of a substitution parameter in a program usually is cumbersome. Managing substitution parameters in SAS DD is not as flexible as managing macro variables in regular SAS even though they behave similarly.

SAS DD is a closed system that is maintained solely in Cary, NC by SAS Institute. To access SAS DD, users need to be authorized. Although it increases the security of the system, it is much less flexible to use than the regular SAS system. There is a limited time (the default time is 60 minutes) for each session. SAS DD automatically shuts down after periods of inactivity and does not save your work. Re-login is required for each new session. Navigating multiple folder levels is often needed to access your destination files.

Since SAS DD resides on the UNIX operating environment, it is case-sensitive. The UNIX operating environment is also unable to utilize the character map in windows available to display special characters. PC SAS does not have this issue.

Partial submission of programs is very helpful for coding, debugging, etc. programming work. However, partial submission of programs in SAS DD does not work as well as in regular SAS. When you highlight a portion of code and then submit it, SAS DD sometimes submits other portions of code in addition to those you highlighted, or simply does not work. In addition, SAS DD often does not undo highlighting of that same code after it has been submitted.

In regular SAS, some procedures such as Proc Print and Proc Freq will have the output windows available right away after submission. In contrast, SAS DD does not have this feature. It takes several steps of navigating through multiple levels of folders to see the output in SAS DD.

Some procedures such as Proc Gdevice are unavailable in SAS DD. SAS DD has an interior program which is device.sas that contains all device information, but requires navigating several folders to access it. It is not as convenient as regular SAS.

High-resolution graphics using png cannot be produced in SDD. We received the following error: ERROR: Unable to allocate sufficient memory.

Unlike regular SAS, nested macros do not work in SAS DD. This could make code much longer than regular SAS. Utility macros cannot be called in the same way as in regular SAS.

SAS DD is unable to get data from Oracle Clinical in an automated manner. It requires PC SAS to connect with Oracle Clinical and then transfer data to SAS DD. Performing batch processing is an advantage of PC SAS. However, SAS DD is not capable of this.
STABILITY
By nature, the interface of a Java powered application in the internet is not as stable as regular SAS. The Process Editor often freezes and forces the programmer to close it down and restart the system. This frustrates programmers. Refreshes are constantly needed during the programming process. Heavy internet traffic makes SDD’s behavior unpredictable and leads to unexpected results. These results make programmers often doubt if the system is sufficiently validated. In addition, when programmers attempt to stop a submitted program, the ‘Abort’ program function does not work well.

EFFICIENCY
In general, generating a program in SAS DD takes much longer than in regular SAS with the same code. To start a program, SAS DD takes at least 30 seconds to open while regular SAS probably needs just a half second. As mentioned above, the total setup process of all substitution parameters in a program usually requires substantial time to be completed while regular SAS probably needs just a few seconds. In addition, the editing, debugging process takes longer than regular SAS.

CONCLUSION
The biggest defect of SAS DD for populating its market is the use of the substitution parameter table, which is an inconvenient way for programming and has changed SAS programmers’ programming habits that have been accumulated for years. Supporting SAS DD requires additional personnel such as IT professionals and more SAS programmers. SAS DD needs substantial improvement in flexibility, stability and efficiency.

ACKNOWLEDGEMENTS
A Special thanks to Thomas Kinghorn and other colleagues in Biostatistics Analysis group at former GCD of MDS Pharma Services, now INC Research for their inputs.

REFERENCES

CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author at:
Amos Shu
INC Research, Inc.
2200 Renaissance Blvd.
King of Prussia, PA 19406
Email: hshu@incresearch.com

TRADEMARK INFORMATION
SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.