

Migration from OpenVMS to Windows NT: Issues and Considerations for Programmers

Adrien Ndikumwami, Social & Scientific Systems, Bethesda, Maryland

ABSTRACT

OpenVMS has been the operating system of choice for deploying SAS® system in many educational and research institutions. The advent of Windows NT, coupled with uncertainties surrounding the future of OpenVMS, is propelling organizations to migrate SAS System applications from OpenVMS to Windows NT. This migration challenges administrators and SAS programmers. An appreciation of the differences and similarities operating systems as they relate to SAS System facilitates the transition process. File structure, file editing, program execution, data sorting, numeric data representation, invoking host-specific commands, magnetic tape processing, and application integration are discussed.

INTRODUCTION

The advent of Windows NT, coupled with uncertainties surrounding the future of OpenVMS, is propelling organizations to migrate SAS System applications from OpenVMS to Windows NT. Migrating SAS system applications across platforms presents challenges to administrators as well as SAS programmers. Programmers who understand the differences at the operating system level can make more efficient use of the SAS system. Failure to do so may result in wasted resources such as disk space and processing time. These inefficiencies particularly become more pronounced when dealing with large data sets. This paper explores the differences and similarities of SAS systems under OpenVMS and Windows NT platforms. The topics discussed include file structure, file editing, program execution, data sorting, numeric data representation, invoking host-specific commands, magnetic tape processing, and application integration.

FILE STRUCTURE

There are subtle differences in file structures between the two operating systems. The files are organized in a hierarchical structure under both operating systems, but noticeable differences exist in the syntax of their corresponding operating system commands (DOS and DCL commands). Under OpenVMS, a file is accessed using the format

- *node::drive:[directory]filename.extension;version.*

Example:

```
PLATO: :DISK1: [PROJECT1] SUGI26.TXT;2
```

Under Windows NT, a file is addressed as *drive:\directory\filename.extension.*

Example:

```
D:\PROJECT1\SUGI26.TXT
```

An OpenVMS migrant will find it easy to adjust to the Windows NT file structure. However, one of the features that a programmer under OpenVMS takes for granted is automatic file versioning. This feature is not available under Windows NT. Creating new data sets with the same name as the existing one would result in overwriting of the old data set. But under OpenVMS, a new version of the file is created, and the existing file remains intact and could be accessed. This feature is good for instant backup under OpenVMS, but can be costly in terms of space when one is writing large files. In case a programmer desires to make use of this instant backup feature, a new option beginning with the SAS 7 version enables the user to keep generation data sets (i.e., to keep multiple copies of a SAS data set). The copies represent versions of the same data set, which is archived each time it is replaced. To request generation data sets, use the GENMAX= data set option.

Example:

```
DATA ONE (GENMAX=3) ;
  X=1 ;
RUN ;
DATA ONE ;
  X=2 ;
RUN ;
DATA ONE ;
  X=3 ;
RUN ;
DATA ONE ;
  X=4 ;
RUN ;
```

In the above example, SAS creates four data sets. The most recent version (also called the base version) is called ONE, and the two backup versions (also called historical versions) are called ONE#002 and ONE#003, respectively. ONE#001 is deleted as SAS writes the fourth data set because the maximum generations was set to only 3.

To reference a specific version of a data set, one can use the GENNUM= data set option.

Example:

```
DATA TWO ;
  SET ONE (GENNUM=-1) ;
RUN ;
```

In the above example, data set TWO is created by setting ONE#003, that is, the 1st most recent version before the base version ONE (also called generation zero).

Using the GENMAX option, one can keep up to 999 historical versions.

FILE EDITORS

A file editor is very important not only for editing programs but also for editing data files. Under OpenVMS, most programmers are used to OpenVMS Text Processing Utility (TPU) editor, EVE, EDT or just the default SAS Text Editor. Under Windows NT, a programmer may use either the enhanced text editor that comes bundled within SAS System or other text editors such as Notepad, TextPad, Kedit, PFE, or UltraEdit. With most advanced editors, the programmer can set them up to launch from within the editor. For example, with UltraEdit, a programmer can configure the editor to run SAS without exiting the editor (Advanced-Tool Configuration). In addition, UltraEdit enables the programmer to set up syntax highlighting (i.e., the ability to recognize predefined SAS key words and display them in a different color[s]). An example of SAS entry to be included in *wordfile.txt* is available at [ftp://ultraedit.com/wf/sas.txt](http://ultraedit.com/wf/sas.txt). However, syntax checking has many limitations when compared to SAS Enhanced Editor.

PROGRAM EXECUTION

In an OpenVMS environment, programmers may execute SAS programs in a variety of ways. These include running programs in interactive line mode, non-interactive mode, spawn mode, or in a batch mode.

In SAS interactive line mode (invoked by \$SAS8/NODMS on OpenVMS), a programmer enters SAS statements line by line in response to prompts that are issued by the SAS System. While running SAS in this mode, the SAS session retains exclusive use of the terminal or terminal window. There is no equivalent execution mode under Windows NT. However, one can highlight a block of

SAS code in the SAS Program Editor and submit it at once for immediate execution.

In SAS non-interactive mode, OpenVMS retains exclusive use of the terminal when the non-interactive SAS job executes, except when programs invoke windowing procedures that require user input. The SAS program is read from a file that is specified by including the filename in the SAS command.

Example:

```
$SAS8 PROG1.SAS
```

This method is available under Windows NT. From Windows Explorer, chose SUBMIT from the right mouse click pull-down menu. The same can be achieved through a less convenient method using the DOS prompt.

Example:

```
>C:\SAS\V8\SAS.EXE PROG1.SAS
```

In a spawn mode (SPAWN=system option), one can submit a program under OpenVMS to run the SAS System in a SPAWN/NOWAIT subprocess. This method of running SAS is similar to batch mode except that in the batch mode, SAS runs in its own process rather than in a spawned subprocess. There is no equivalent of the spawn mode under Windows NT.

The last option under OpenVMS is the batch environment. Unlike a spawned process, a batch job is independent of other processes; it is not linked to an OpenVMS parent process. In an OpenVMS environment, this is accomplished by creating a batch file that consists of DCL commands that invoke SAS and then submitting the batch file to an appropriate queue.

Example:

```
$ SUBMIT/NOTIFY/LOG/QUE=SASQ MYBATCH.COM
```

Batching is available under Windows NT. However, it is done differently. In Windows NT, SAS programs can be submitted in batch mode from either Windows Explorer or from DOS. From Windows Explorer, right click on the SAS program file and choose SUBMIT BATCH under the pull-down menu. Under the DOS prompt, create a DOS batch file that includes the statement to run a SAS program.

Example:

```
REM MYBATCH.BAT
C:\SAS\V8\SAS.EXE -SYSIN D:\PROJ1\PROG1.SAS
```

To run the batch file, just type MYBATCH at the DOS prompt.

In both cases, SAS creates a LOG and LST (LIS for OpenVMS) in the directory where the batch file resides.

Under Windows NT, it is possible to run a program by just dragging SAS programs from other Windows applications onto an open SAS session and submitting them. In addition, dragging files that contain SAS code and dropping them on an open SAS session will submit them.

Under OpenVMS, users can schedule a batch job to run at a particular time in the future using *SUBMIT/AFTER={time}*. Under Windows NT, scheduling a batch job may be done using a third party software package or by issuing the "AT" DOS command.

Example:

```
AT\NTS1 23:59/I C:\SAS\V8\SAS.EXE -SYSIN
D:\PROJ1\PROG1.SAS
```

The above command tells Windows NT to run PROG1.SAS just one minute before midnight. The program will run in a batch mode on server NTS1. The AT command will not run if the server is omitted or if the user does not have appropriate security privileges to run a batch program.

To stop a program, OpenVMS users are accustomed to using Control-Y or just deleting the queue for batches. Under Windows

NT, one can stop a program execution by closing the SAS session from the pop-up menu for SAS on the *Windows Task Bar* or by *End Task* from within the Windows NT Task Manager. For each of these methods, the SAS System displays a dialog box to allow the user to verify the selection.

SORTING

Sorting is one of the most time- and resource-consuming procedures. This is especially true when processing large transaction data sets (retail, hospital discharges, insurance claims, etc.). A good understanding of how SAS system uses PROC SORT is critical to efficiently sorting large data sets. Under OpenVMS, sorting can be done by the host or by the SAS system. Under Windows NT, sorting is only performed by the SAS system. Although the SORTPGM= option can take the value HOST, it is for compatibility reasons, because SAS (not the NT) will perform the sorting. SAS SORT can be supplemented by third-party software like SYNCSORT.

The sorting area also differs in both platforms. When processing large files, one has to make sure there is enough space to process all data steps and procedures. Under OpenVMS, sorting physically happens wherever the logical SYS\$SCRATCH is pointing. Under Windows NT, intermediate files for the sorting process are stored in the SAS work directory. This is a big constraint under Windows NT, because the sort space cannot be dissociated from the workspace, a feature that exists under OpenVMS. The partial remedy to space constraints created by PROC SORT is to specify the OUT= option. This way a programmer can direct the sorted output to a disk with enough space. On both systems, it is important to have enough space to hold three to four times the size of the file to be sorted. Under OpenVMS, one can specify up to 10 sorting locations so that sort space that spawns multiple disks may be used, if required. That option is not available under Windows NT. Therefore a programmer has to direct the SAS work directory to a disk that has enough space. To accomplish this, modify the -WORK statement by editing SASV8.CFG.

Example:

```
-WORK "D:\SAS8TEMP\"
```

In this example, SAS uses D:\SAS8TEMP as a work area, the equivalent of the directory specified by the logical SAS\$WORK under OpenVMS.

NUMERIC DATA REPRESENTATION

Numeric data sometimes takes more space than needed. That is why it is important to understand the difference in terms of numeric data representation to be able to judge how to use the LENGTH statement, which is an important option to help reduce the size of a file. The default length of numeric variables in SAS data sets is 8 bytes. However, certain numeric data values, especially integers, can be stored in less than 8 bytes without compromising on precision. Under OpenVMS, the data representation depends on the architecture. In the SAS System under OpenVMS VAX, the data type of numeric values is D_FLOATING. The precision of this type of floating-point value is 16 decimal digits. In the SAS System under OpenVMS Alpha, the data type of numeric variables is IEEE double precision or T_FLOATING. In addition, with Alpha T_FLOATING numbers, you can specify a minimum length of 3 bytes using the LENGTH statement in the DATA step. However, with VAX D_FLOATING numbers, the minimum length you can specify is 2 bytes. In the SAS System under Windows NT, the Windows data type of numeric values that have a length of 8 is LONG REAL. The precision of this type of floating-point values is 16 decimal digits, almost the same as Alpha OpenVMS. The following tables indicate the largest integer that can be displayed by the three systems. Please note that the column for Alpha OpenVMS is not included because the largest OpenVMS values are one unit less than Windows NT values (8191, 2097151, etc.).

