

"Using the SAS® System When Transitioning to a New Windows® Workstation"

Orion V. Hanger, Nielsen Media Research

ABSTRACT

When transitioning from one Windows workstation to another, the author found it very useful to write a SAS program to read PC hard drive directory listings for both machines. With this data, it was possible to compare files and directories to identify those that should be moved to the new machine and those that should not be overwritten there (preinstalled software, newer release of the operating system, etc.). This paper presents the basic operations and code created to do this and shows how the SAS System of Software for Windows can be useful for general PC based hard disk file/directory analysis.

CREATING THE RAW DATA

The first step of this process is to create a text file containing a complete listing of all directories and their related files. In the PC/DOS environment this is done by directing the output of the DIR command to a file:

```
DIR C:\ /S > THKPDOLD.TXT
```

This command tells the operating system to display a list of all files and directories in the root directory on the 'C' drive (the 'C:\' parameter) and for all subdirectories (the '/S' switch). Instead of displaying that list to the default output (typically the computer screen), the output of the command (via the '>' redirection character) is written to a file named THINKPOLD.TXT in the current directory. Additional information on the DIR command, available switches and DOS output redirection is available in the DOS user's guide or via the DOS 'HELP' command. Depending upon the version of DOS that is being used, there may be some differences in the format of the output and DIR command capabilities.

READING THE INPUT FILES

In order to compare and summarize the disk organization and configuration of the old versus the new workstations, a file was created for each of them in the manner shown above. A review of the created files indicated that the desired data (directory name, filename, extension, number of bytes, date and time) is in a consistent format:

```
Volume in drive C is KWAEBUS
Volume Serial Number is 1930-17F7
```

Directory of C:\CIVIC

```

.           <DIR>      05-18-95   3:52p
..          <DIR>      05-18-95   3:52p
SASWORK     <DIR>      05-18-95   3:53p
SASUSER     <DIR>      05-18-95   3:53p
CIVMAIN SC2    147,712 01-11-95   3:39p
SOURCAT SC2    160,000 12-09-94   7:45p
CONTROL SD2     4,352 11-06-94  11:10a
CIVCAUTO SAS     577 08-08-94   5:03p
          10 file(s)      312,641 bytes
```

The above is an example of a small file created for just one directory. The 'report' created for the entire disk drive will contain a 'Directory of' section for every directory, as well as, a 'Total bytes' section (total files, total bytes used and total bytes available) at the very end.

Next a SAS routine is needed to read the raw data for each workstation into a separate SAS dataset:

```

data thkpdold;
  length directory $ 44; retain directory;
  infile 'c:\thkpdold.txt' length=len missover;
  keep directory filename fileext bytes1
      datel time1;
  length filename $ 8 fileext $ 3 ind $ 1;
  format datel mmdyy8. time1 time5. ;
  input @1 @; if len<5 then delete;
  input @1 testline $varying80. len @ ;
  if index(testline,'Volume in drive')>0 or
     index(testline,'Volume Serial')>0 or
     index(testline,'<DIR>')>0 or
     index(testline,'bytes')>25 or
     index(testline,'Total files')>0
  then delete;
  else if index(testline,'Directory of')>0
  then do;
    xxx=index(testline,'Directory of');
    input @xxx+13 directory ; return;
  end;
  input @1 filename $8. @10 fileext $3.
        bytes1 :comma12. datel mmdyy8. ;
```

```

time1=input(substr(testline,len-5,5),time5.);
ind=substr(testline,len);
if ind='p' & hour(time1)<12
    then time1=time1+'12:00't;
output;
run;

```

The above code was used to read the raw data file for the old workstation. The same code (with different input file and output designations) was used to obtain the data for the new workstation. It demonstrates some of the many capabilities of the SAS System for reading data from a wide variety of data formats. Each record is checked to determine if it contains text strings or characteristics indicative of non-data lines and, if found, is eliminated from further processing. Some allowances were made in the code to account for some slight DOS version related differences in the location of data or text strings on the output from the DIR command (note the input code for directory name). In addition, time is displayed as 'am' (a) and 'pm' (p). When developing this code, some testing was done to determine the maximum necessary record and directory variable name length.

ANALYZING THE DATA

The information that was desired from this data for the task at hand was necessary to provide the technician with a list of files and directories that needed to be copied from the old workstation to the new workstation. The new workstation had come with it's own set of pre-installed and pre-configured software. To simply copy everything from the old machine to the new machine could potentially mix or overwrite newer versions of software components in ways that are undesirable. To manually review and compare the contents of the two hard drives could be very time consuming. In addition to helping with the migration process, this exercise also provided a means to summarize disk space utilization by major directory names.

First, the two data sets are merged by directory, filename and extension:

```

data newonly(drop=bytes1 date1 time1) both
    oldonly(drop=bytes2 date2 time2);
merge thkpdold(in=old) thkpdnew(in=new);
by directry filename fileext;
if old and not new then output oldonly;
else if not old and new then output newonly;
else output both;
run;

```

The 'oldonly' dataset was further modified to eliminate data for directories that should not be copied:

```

data oldonly; set oldonly;
if directry='C:\BCWIN' then delete;
else if directry='C:\BFWIN' then delete;

output oldonly;
run;

```

A similar process was performed for the 'both' dataset to eliminate data for directories where the file(s) on the old workstation should not replace the ones on the new. Both of these subsetting processes brought the amount of data (files to be copied) down to a level where it was relatively easy to print, review and mark those files that should or should not be copied.

In addition, the author found it very useful to create and read an input file for the new workstation and generate a summary report of total bytes used by 'main' directory name (root and the subdirectories immediately under it):

```

data temp; set thkpdnew;
length maindir $ 12 ; keep maindir bytes1;
maindir=substr(directry,1,3) ||
    trim(scan(substr(directry,4),1,''));
output temp;
run;

```

The output of this code could then be summarized using PROC SORT and PROC MEANS to get totals (to help to determine whether or not some pre-installed software is really needed in lieu of it's disk space requirements).

CONCLUSION

Considering the growth in the amount of processing power, software and disk space that is available to today's users on their personal workstations, managing and configuring that resource can be a difficult task. This paper has presented an example of how the SAS System of Software for Windows can be used to obtain, analyze and present data/information on PC disk space usage and configuration. For the technician who may be responsible for moving the user to a new workstation, this can help them better understand the user's unique needs. And for the user, it helps them better understand and manage their own environment and to communicate their needs to the technician who may be assisting them.

SAS is a registered trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Windows is a registered trademark of the Microsoft Corporation in the USA and other countries.