

## File Management Using Pipes and X Commands in SAS®

Emily K.Q. Sisson, Boston University School of Public Health

### ABSTRACT

SAS for Windows can be an extremely powerful piece of software, not only for analyzing data, but also for organizing and maintaining output and permanent datasets. By employing pipes and operating system ('X') commands within a SAS session, you can easily and effectively manage files of all types stored on your local network.

### INTRODUCTION

Not only are SAS programmers responsible for creating, maintaining, and documenting SAS programs, but also the associated output and permanent datasets. SAS can be a very handy tool in managing the various files occupying network folders.

While this paper will display syntax specific to file management on a Windows-based computer, the concepts extend to UNIX- and Linux-based systems.

### INVENTORYING YOUR NETWORK

#### GENERATE A FILE LISTING

The first step in understanding what files need to be managed is to import a directory listing of file attributes into a SAS dataset. Using a pipe device type on a filename statement allows you to invoke DOS commands within a SAS session. By executing the DOS command for a directory listing ("dir"), an inventory of the specified directory can be input into a SAS dataset. Additionally, the "/s" option should be included in the DOS command to recursively process any subfolders within that directory. Since the length of each record in the directory will differ, it is prudent to input the data into a SAS dataset using the \$VARYINGw. format. The LENGTH option will assign the internally stored record length to RECLLEN, which then becomes the length-variable argument to the \$VARYINGw. format. An example of this code is shown here:

```
filename dirlist pipe 'dir "Y:\MyFiles" /s';
data dirlist ;
    length lineinfo $256 ;
    infile dirlist length=reclen ;
    input lineinfo $varying256. reclen ;
run;
```

As seen in Display 1, the resulting DIRLIST dataset is one that contains many observations, but only one column. All the information gathered through the DOS "dir" command is stored in a text string.

	lineinfo
1	Volume in drive Y is Y03 - Y Drive (SPH)
2	Volume Serial Number is E65B-E2D5
3	
4	Directory of Y:\MyFiles
5	
6	02/24/2016 01:39 PM <DIR> .
7	02/24/2016 01:39 PM <DIR> ..
8	08/12/2015 08:38 AM 38,400 1281 ID Report.xls
9	11/03/2015 11:39 AM 31,232 Activity Review data Audit Form.xls
10	12/13/2013 01:21 PM 23,053 AnyDeliveryAnalysisbak.sas
11	10/17/2014 07:26 AM 84,911 assist_v3_english.pdf
12	11/02/2015 10:34 AM 20,344 BoxPlot1.png

Display 1. Excerpt from the DIRLIST dataset

## MANIPULATE A FILE LISTING

Once there exists a dataset containing information about directory locations and file attributes, it can be used to manage files. For illustrative purposes, this paper will consider a scenario where you need to create archive folders within the existing directory and subsequently move files to each based on file date and type. Within the root folder, the intent is to have one subfolder per year and within each subfolder have a folder for each file type (PDF, Word Document, etc.).

Fortunately, the text string stored in the DIRLIST dataset can be manipulated, truncated, and excerpted to suit a variety of needs. For example, you can create variables representing file path, file name, file extension, and date of last modification. With some simple code, this one column dataset can be morphed into a more useful tool for managing files on a network:

```
data dirlist_useful;
  set dirlist;
  /*Path of the directory appears once, use retain statement to assign*/
  length directory $1000;
  retain directory;
  if left(upcase(lineinfo))=: 'DIRECTORY OF' then
      directory = substr(left(lineinfo),14);

  /*Isolate other important information*/
  filename = substr(left(lineinfo),40);
  fileextens = scan(strip(lineinfo),-1);
  filedate = input(substr(lineinfo,1,10),?? mmddyy10.);
      format filedate mmddyy10.;
  filetime = input(scan(lineinfo,4)||" "||scan(lineinfo,5),time12.);
      format filetime time12.;

  /*Categorize file extensions as file types, extract year from date*/
  [lines omitted]

  /*Delete extraneous rows*/
  if lineinfo = '' then delete;
  if index(upcase(lineinfo),'<DIR>') then delete;
  if left(upcase(lineinfo)) =: 'VOLUME' then delete;
  if left(upcase(lineinfo))=: 'DIRECTORY OF' then delete;
  if fileextens in ('bytes' 'Listed:' 'free') then delete;
run;
```

Display 2 shows a selection of the newly created dataset. This will prove significantly more useful than the previous iteration.

	directory	filename	fileextens	filetype	filedate	year	filetime
1	Y:\MyFiles	1281 ID Report.xls	xls	Word and Excel Documents	08/12/2015	2015	8:38:00
2	Y:\MyFiles	Activity Review data Audit Form.xls	xls	Word and Excel Documents	11/03/2015	2015	11:39:00
3	Y:\MyFiles	AnyDeliveryAnalysisbak.sas	sas	SAS Files	12/13/2013	2013	13:21:00
4	Y:\MyFiles	assist_v3_english.pdf	pdf	PDFs	10/17/2014	2014	7:26:00
5	Y:\MyFiles	BoxPlot1.png	png	Other File Types	11/02/2015	2015	10:34:00
6	Y:\MyFiles	BoxPlot2.png	png	Other File Types	11/02/2015	2015	10:34:00
7	Y:\MyFiles	BoxPlot3.png	png	Other File Types	11/02/2015	2015	10:35:00
8	Y:\MyFiles	CDSelf-Efficacy32.pdf	pdf	PDFs	10/17/2014	2014	7:15:00
9	Y:\MyFiles	CESD-10 English.pdf	pdf	PDFs	10/17/2014	2014	7:15:00
10	Y:\MyFiles	Chart Review 6 Contents.xlsx	xlsx	Word and Excel Documents	03/09/2015	2015	14:13:00
11	Y:\MyFiles	Chart Review Audit Form 12 Month partid 1258.xls	xls	Word and Excel Documents	07/14/2015	2015	12:52:00
12	Y:\MyFiles	Chart Review Audit Form 18 Month partid 1003.xls	xls	Word and Excel Documents	08/06/2015	2015	9:41:00
13	Y:\MyFiles	Chart Review Audit Form 6 Month partid 1379.xls	xls	Word and Excel Documents	07/08/2015	2015	10:12:00
14	Y:\MyFiles	CheckEncounterLocationNotRoFReport.sas	sas	SAS Files	12/18/2014	2014	11:55:00

**Display 2. Excerpt from the DIRLIST\_USEFUL dataset**

## MANAGING YOUR NETWORK

### CREATE NEW NETWORK FOLDERS

In addition to pipe devices, SAS can communicate with a network through operating system ('X') commands. When an 'X' command is invoked within SAS, you are placed in a Windows command prompt session, and any subsequent statements are executed as Windows commands. Among other things, these commands can be used to make or delete directories, and move, copy, and delete individual files. The 'X' command *mkdir* can be used to create a new folder in a directory. The *-p* option ("parent") will create any intermediate directories should they not already exist:

```
x "mkdir -p Y:\MyFiles\newfolder";
```

Once in this command prompt session, you must type 'exit' into the command prompt window before returning to the SAS session. Luckily, there is a system option in SAS to avoid having to type 'exit' after each command: *noxwait*. Since this paper is focusing on archiving by both file type and date, using a macro to invoke the *mkdir* command repeatedly is the most efficient means to create your desired subfolders:

```
/*Isolate unique combos of Directory/Year/File Type to create folders*/
proc sql;
  create table dir_yr_typ as
  select distinct directory, year, filetype
  from dirlist_useful;
quit;

/*Using the dir_yr_typ dataset, create a folder for every combo of yr/typ*/
%macro createdir(dir=,yr=,typ=);
  x "mkdir -p &dir.\&yr.\&typ.";
%mend createdir;

options noxwait;
data _null_;
  set dir_yr_typ;
  command = cats('%createdir(dir=',directory,', yr=',year,',
                typ=',filetype,')');
  call execute(command);
run;
```

After isolating unique combinations of directory/year/file type, the call execute function is used within a data \_NULL\_ step to invoke the %createdir macro many times over. Display 3 illustrates the newly defined "command" column in the data \_NULL\_ step. Through the use of the call execute function, the macro is invoked once per observation by processing the macro call stored in "command".

	directory	year	filetype	command
1	Y:\MyFiles	2005	PDFs	%createdir(dir=Y:\MyFiles, yr=2005, typ=PDFs);
2	Y:\MyFiles	2005	Word and Excel Documents	%createdir(dir=Y:\MyFiles, yr=2005, typ=Word and Excel Documents);
3	Y:\MyFiles	2010	PDFs	%createdir(dir=Y:\MyFiles, yr=2010, typ=PDFs);
4	Y:\MyFiles	2013	SAS Files	%createdir(dir=Y:\MyFiles, yr=2013, typ=SAS Files);
5	Y:\MyFiles	2014	Other File Types	%createdir(dir=Y:\MyFiles, yr=2014, typ=Other File Types);
6	Y:\MyFiles	2014	PDFs	%createdir(dir=Y:\MyFiles, yr=2014, typ=PDFs);
7	Y:\MyFiles	2014	SAS Files	%createdir(dir=Y:\MyFiles, yr=2014, typ=SAS Files);
8	Y:\MyFiles	2014	Word and Excel Documents	%createdir(dir=Y:\MyFiles, yr=2014, typ=Word and Excel Documents);
9	Y:\MyFiles	2015	Other File Types	%createdir(dir=Y:\MyFiles, yr=2015, typ=Other File Types);
10	Y:\MyFiles	2015	PDFs	%createdir(dir=Y:\MyFiles, yr=2015, typ=PDFs);
11	Y:\MyFiles	2015	SAS Files	%createdir(dir=Y:\MyFiles, yr=2015, typ=SAS Files);
12	Y:\MyFiles	2015	Word and Excel Documents	%createdir(dir=Y:\MyFiles, yr=2015, typ=Word and Excel Documents);
13	Y:\MyFiles	2016	SAS Files	%createdir(dir=Y:\MyFiles, yr=2016, typ=SAS Files);

Display 3. Excerpt from the DATA \_NULL\_ step

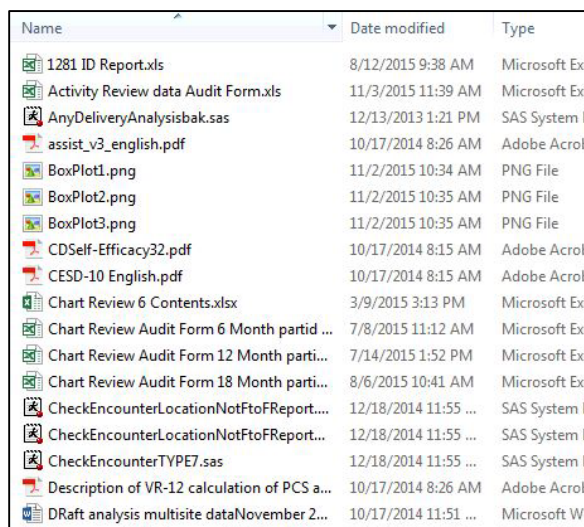
## REORGANIZE FILES

Similarly, a simple macro can be created to move existing files to their new destinations. Here, the 'X' command of choice is the *move* command followed by a file's current location then new location. The DIRLIST\_USEFUL dataset is used in a data \_NULL\_ step to generate a new set of macro invocations, this time for the newly defined %movfil macro:

```
/*Using the original dirlist_useful set, move old files to new
destinations*/
%macro movfil(dir=,name=,yr=,typ=);
  x move "&dir\&name" "&dir\&yr\&typ\&name";
%mend movfil;

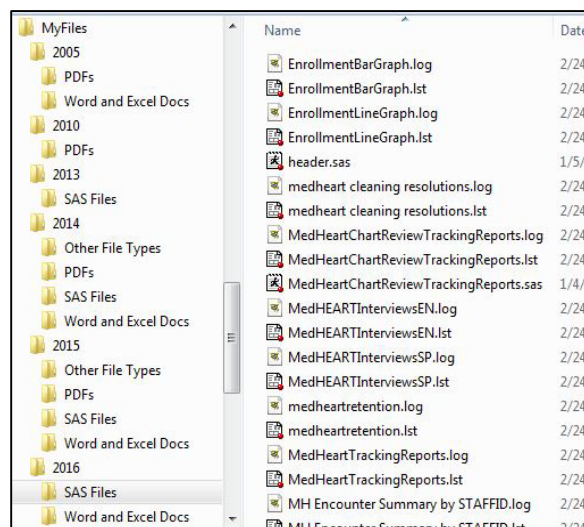
data _null_;
  set dirlist_useful;
  command = cats('%movfil(dir=',directory,', name=',filename,',
                 typ=',upcase filetype)',',yr=',year,')');
  call execute(command);
run;
```

Once the DATA \_NULL\_ step has been executed, the hundreds of files found in the original folder have been moved to their new locations. Having used pipes to inventory your directory and 'X' commands to subsequently restructure it, your documents are much more organized. Displays 4a and 4b demonstrate the original and final layouts of the directory.



Name	Date modified	Type
1281 ID Report.xls	8/12/2015 9:38 AM	Microsoft Ex
Activity Review data Audit Form.xls	11/3/2015 11:39 AM	Microsoft Ex
AnyDeliveryAnalysisbak.sas	12/13/2013 1:21 PM	SAS System f
assist_v3_english.pdf	10/17/2014 8:26 AM	Adobe Acrob
BoxPlot1.png	11/2/2015 10:34 AM	PNG File
BoxPlot2.png	11/2/2015 10:35 AM	PNG File
BoxPlot3.png	11/2/2015 10:35 AM	PNG File
CDSelf-Efficacy32.pdf	10/17/2014 8:15 AM	Adobe Acrob
CESD-10 English.pdf	10/17/2014 8:15 AM	Adobe Acrob
Chart Review 6 Contents.xlsx	3/9/2015 3:13 PM	Microsoft Ex
Chart Review Audit Form 6 Month parti...	7/8/2015 11:12 AM	Microsoft Ex
Chart Review Audit Form 12 Month parti...	7/14/2015 1:52 PM	Microsoft Ex
Chart Review Audit Form 18 Month parti...	8/6/2015 10:41 AM	Microsoft Ex
CheckEncounterLocationNotFtoFReport...	12/18/2014 11:55 ...	SAS System f
CheckEncounterLocationNotFtoFReport...	12/18/2014 11:55 ...	SAS System f
CheckEncounterTYPE7.sas	12/18/2014 11:55 ...	SAS System f
Description of VR-12 calculation of PCS a...	10/17/2014 8:26 AM	Adobe Acrob
DRaft analysis multisite dataNovember 2...	10/17/2014 11:51 ...	Microsoft W

Display 4a. Original Layout of Y:\MyFiles



Name	Date
EnrollmentBarGraph.log	2/24/
EnrollmentBarGraph.lst	2/24/
EnrollmentLineGraph.log	2/24/
EnrollmentLineGraph.lst	2/24/
header.sas	1/5/2
medheart cleaning resolutions.log	2/24/
medheart cleaning resolutions.lst	2/24/
MedHeartChartReviewTrackingReports.log	2/24/
MedHeartChartReviewTrackingReports.lst	2/24/
MedHeartChartReviewTrackingReports.sas	1/4/2
MedHEARTInterviewsEN.log	2/24/
MedHEARTInterviewsEN.lst	2/24/
MedHEARTInterviewsSP.log	2/24/
MedHEARTInterviewsSP.lst	2/24/
medheartretention.log	2/24/
medheartretention.lst	2/24/
MedHeartTrackingReports.log	2/24/
MedHeartTrackingReports.lst	2/24/
MH Encounter Summary by STAFFID.log	2/24/
MH Encounter Summary by STAFFID.lst	2/24/

Display 4b. Restructured Layout of Y:\MyFiles

## CONCLUSION

This paper highlights the value of being able to execute Windows operations within a SAS session. Using pipes and operating system commands, you can inventory, manage, and create a complex hierarchy of varying file types very easily. With thoughtful use of the call execute function, hundreds, if not thousands, of files can be manipulated in a matter of seconds. Archiving files is just one of many applications possible with these tools.

## ACKNOWLEDGEMENTS

A special thanks to Clara Chen, Jamie Collins, and Joseph Palmisano for their thoughtful review of this manuscript.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Emily K.Q. Sisson  
Boston University School of Public Health  
Data Coordinating Center  
[eq@bu.edu](mailto:eq@bu.edu)

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.