

## Paper 219-26

**SAS® Macros as File Management Utility Programs**

Christopher J. Rook, EDP Contract Services, Bala Cynwyd, PA  
Shi-Tao Yeh, EDP Contract Services, Bala Cynwyd, PA

**ABSTRACT**

This paper provides several SAS macros for file management and file conversion tasks. The SAS product used in this paper is SAS® BASE, with SAS system installed on the UNIX and Windows® operating systems.

**INTRODUCTION**

The SAS macros provided in this paper are for file management utility purposes. Each macro is a stand-alone program to perform a specific task for the platform specified. Such tasks include file concatenation of existing external files, common page concatenation of existing external files and file conversion to HTML for existing external files.

This paper contains several sections. Each section discusses a SAS macro with SAS code and macro invocation example.

**SAS MACROS FOR FILE CONCATENATION:**

The SAS macros *concat1*, *concat2* and *concat3*, presented in this paper, can be used to concatenate existing external files. This type of file management is useful for a task such as printing a large number of existing external files. In this case it may be much more time efficient for you to combine all of the external files into a single file and simply print that file. The SAS macro *concat1* will accomplish this task on the UNIX operating system. The SAS macro *concat2* is for use with the Windows operating system and has extra features such as the option to insert a page feed character between the external files in the concatenated file. The SAS macro *concat3* is also for use on the UNIX operating system and is different from *concat1* because it allows you to select the specific page from each file that is to be concatenated together. As an example, the macro *concat3* could be used when QCing several SAS output files and you would like to print the first page of each to check that column total counts are consistent across tables. You can call the macro *concat3* specifying the first page of each output file and this will result in a single file being created that contains the first page of each output file in the specified directory.

**File Concatenation Macro for UNIX****Macro Name: CONCAT1**

The following SAS macro *concat1* is for file concatenation on the UNIX operating system. This macro is designed to take several existing external files and concatenate them into a single file.

```

/*****
* MACRO PARAMETERS:
*
* prmptcmd: This is the command to be executed at the UNIX
*           prompt that will provide a list of the desired files
*           to concatenate.
*
* catfile: This is the name of the file that will be created to
*           contain all output files concatenated together.
*****/

%macro concat1(prmptcmd=, catfile=);

filename lstfiles pipe "&prmptcmd";

data allfiles;
length fname $50;
infile lstfiles;
input fname $;
call execute('x "cat ' ||fname||' >> &catfile;');
run;

%mend concat1;

```

**Example Invocation of SAS Macro CONCAT1:**

The following SAS code shows an example invocation of *concat1*. All source files are in the same subdirectory. All files with an extension of ".lst" will be concatenated together and saved with the new file name "allout.lst".

```
%concat1(prmptcmd=ls *.lst, catfile=allout.lst);
```

**File Concatenation Macro for Windows****Macro Name: CONCAT2**

The following SAS macro *concat2* is for file concatenation under the Windows operating system. This macro makes use of some added features that are not present in the macro *concat1*. Namely, this macro checks to see if the file specified by the parameter *catfile* exists, and if so, deletes it. In addition, an option to insert a page feed character between concatenated files is included. There is also a fundamental way in which this macro differs from *concat1* and that is that it doesn't execute the concatenation statements directly from the command prompt, but creates a batch file in MS-DOS® that is executed as the final step in the process. The batch file is named "concat.bat" and resides in the current SAS working directory. This will ensure that it remains temporary since it will be deleted as soon as the

SAS session has ended. (The file "concat.bat" is created by specifying the current working directory in the file statement in a data step. Put statements then add the desired content to the file. The last line of the batch file contains the word "exit" to close the DOS-Prompt window and return to the current SAS session. SAS then continues to process the remaining code in the program.)

```

/*****
* MACRO PARAMETERS:
*
* dir: Directory where outputs reside.
*
* prmptcmd: This is the list of options to be added to the "dir"
*           statement in MS-DOS which will provide a list of
*           the desired files to standard output. These are
*           the files that will be concatenated together.
*
* catfile: Name of file to contain all of the concatenated files.
*
* pagefeed: Set to Y if a page feed is needed between files.
*****/

%macro concat2(dir=, prmptcmd=, catfile=, pagefeed=N);

filename filelist pipe "dir %cpress(&dir)&prmptcmd";

/*****
* Delete external file with the same name as the concatenation
* file. (If it exists.)
*****/

filename tmp "%cpress(&dir)%cpress(&catfile)";
%let rc=%sysfunc(fdelete(tmp));

/*****
* If a page feed is requested between each output, create a
* file in the work directory that contains a single page feed.
*****/

%if %upcase(&pagefeed)=Y %then %do;
  data _null_;
  file "%sysfunc(pathname(work))\pagefeed.txt";
  pagefeed=byte(12);
  put @1 pagefeed;
  run;
%end;

/*****
- Create batch file in the current work directory used to
* concatenate files.
* - If a page feed is requested between each file, append the
* "pagefeed.txt" file between each output file.
* - Execute the batch file.
*****/

data _null_;
length filename $200;
infile filelist end=last;
input filename $;
file "%sysfunc(pathname(work))\concat.bat";
cmd = "type %trim(&dir)||compress(filename)||"
      >> %trim(&dir)%left(&catfile)";
put cmd;
%if %upcase(&pagefeed)=Y %then %do;
  if not(last) then do;

```

```

cmd = "type
      %sysfunc(pathname(work))\pagefeed.txt
      >> %trim(&dir)%left(&catfile)";
  put cmd;
  end;
%end;
if last then do;
  put "exit";
  call execute('x "%sysfunc(pathname(work))\concat";');
end;
run;

%mend concat2;

```

#### Example Invocation of SAS Macro CONCAT2:

The following SAS code shows an example invocation of **concat2**. All files with an extension of ".lst" in the subdirectory "u:\concat\" will be concatenated together and saved with the new file name "allfiles.lst". In addition, a page feed will be inserted between each of the concatenated files.

```

%concat2(dir=u:\concat\, prmptcmd=*.lst /b, catfile=allfiles.lst,
pagefeed=Y);

```

#### Common Page File Concatenation Macro for UNIX Macro Name: CONCAT3

The following SAS macro **concat3** is for file concatenation under the UNIX operating system. This macro contains the added feature of selecting only one page from each file and concatenating all of the single pages together. The page common to each output file to be concatenated together is specified by the parameter *pagenum*.

```

/*****
* MACRO PARAMETERS:
*
* dir: Directory where outputs reside.
*
* filetype: This is the extension of the files from the directory
*           specified by the parameter "dir" to concatenate
*           together.
*
* catfile: Name of file to contain all of the concatenated files.
*           (This file will be given the same extension as that
*           specified by the parameter filetype.)
*
* pagenum: Set to the number of the page of each file that
*           is to be concatenated.
*****/

%macro concat3(dir=, filetype=, catfile=, pagenum=1);

/*****
* Create macro to read in an external file, specified by para-
* meter filename, sending the page specified by the parameter
* pagenum to a new file with the same name, located in the
* work directory.
*****/

```

```

%macro getpagex(filename=);

data _null_;
infile "&dir.&filename" missover end=last;
input @1 fstchar $1.;
retain cnt 1;
pagefeed=byte(12);
if (cnt=&pagenum) then do;
file "%sysfunc(pathname(work))&filename";
if (fstchar ne pagefeed) then put _infile_;
if last or (fstchar eq pagefeed) then put @1 pagefeed;
end;
else if (cnt > &pagenum) then stop;
if fstchar=pagefeed then cnt=cnt+1;
run;

%mend getpagex;

/*****
* Create data set containing 1 record per list file.
* Send first page of each list file to the work directory using
* the same name as the file containing all pages. Uses macro
* "getpagex" created previously.
*****/

filename lstfiles pipe "ls &dir.*.&filetype";

data allfiles;
length fullname $100;
infile lstfiles;
input fullname $;
filename=reverse(scan(reverse(fullname),1,'/'));
call execute("%getpagex (filename=||compress(filename)||');");
run;

/*****
* If a file already exists with the same name as the output file,
* then delete it.
*****/

filename allfiles "&dir.&catfile..&filetype";

%if %sysfunc(fexist(allfiles)) %then
  %let rc=%sysfunc(fdelete(allfiles));

/*****
* Concatenate the chosen page of each of the list files together,
* creating a new file using the name of the parameter catfile.
*****/

data _null_;
set allfiles;
call execute('x "cat '||"%sysfunc(pathname(work))"||
  compress(filename)||' >> '||"&dir.&catfile..&filetype"||"'');
run;

%mend concat3;

```

#### Example Invocation of SAS Macro CONCAT3:

The following SAS code shows an example invocation of the macro **concat3**. The first page of each file with an extension of "lst" in the subdirectory "u:\concat" will be concatenated together and saved with the new file name "allfiles.lst".

```
%concat3(dir=u:\concat/,filetype=lst, catfile=allfiles,pagenum=1);
```



#### Notes:

1.) The "pipe" option on the filename statement is used to execute an operating system statement from the command prompt directing the output away from standard output and into a SAS data set through an infile statement. This is all done within the SAS program.

2.) Any temporary files that are created by the above macros are stored in the SAS work directory, which ensures they are deleted once the SAS session is over. This is easily accomplished by first determining the physical location of the SAS work directory, then directing the content to that location using file and put statements. There are many ways of determining the physical location of the current work directory and the method we use in this paper is the macro code: "%sysfunc(pathname(work))".

#### Macro for File Conversion to HTML for Windows

##### Macro Name: CONVHTML

SAS Web Formatters are good tools for producing your SAS output in HTML format. In some regulatory situations, you are not allowed to rerun your old SAS program to regenerate the old SAS output with newer date on the output. The following SAS macro is one solution for the above situation. It reads in the existing SAS output file and puts HTML tags **<PRE>** and **</PRE>** at the beginning and at the end of the output file to convert it to HTML format.

```

/*****
* MACRO Parameters:
*
* filein: This is the entire file location (including the directory)
*         of the SAS output file to be converted to an HTML
*         file.
*
* fileout: This is the entire file name and location where the
*         HTML file will be written to.
*****/

%macro convhtml(filein=, fileout=);

filename readin "&filein";

```

```

data _null_;
infile readin end=last;
file "&fileout";
input;
if _n_=1 then put @1 '<PRE>';

```

```
if not last then do;
  put _infile_;
end;
else put @1 '</PRE>';
run;
```

```
%mend convhtml;
```

**Example Invocation of SAS Macro CONVHTML:**

The following SAS code shows an example invocation of the macro *convhtml*. Select a target SAS output file with its location and assign the information to macro parameter *filein*. Assign a new file name for the converted file with file name extension of "html" to macro parameter *fileout*.

```
%convhtml(filein=c:\ar\file1.lis, fileout=file1.html);
```

```
<PRE>
```

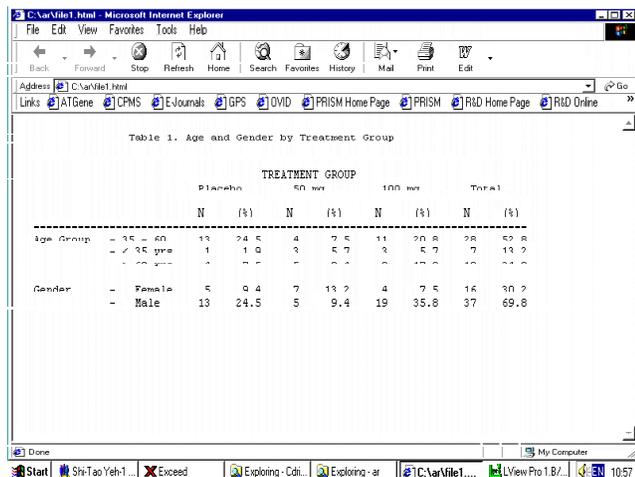
Table 1. Age and Gender by Treatment Group

```
=====
```

		TREATMENT GROUP							
		Placebo		50 mg		100 mg		Total	
		N	(%)	N	(%)	N	(%)	N	(%)
Age Group	- 35 - 60	13	24.5	4	7.5	11	20.8	28	52.8
	- < 35 yrs	1	1.9	3	5.7	3	5.7	7	13.2
	- > 60 yrs	4	7.5	5	9.4	9	17.0	18	34.0
Gender	- Female	5	9.4	7	13.2	4	7.5	16	30.2
	- Male	13	24.5	5	9.4	19	35.8	37	69.8
Total	-	18	34.0	12	22.6	23	43.4	53	100.0

```
</PRE>
```

**Figure 1. Converting an Existing SAS Output File to HTML Format**



**Figure 2. Converted SAS Output File on Browser**



**Note:** In HTML files, white space is usually ignored by the browser. When using the `<PRE>` and `</PRE>` tags, however, white space is preserved and the resulting file is viewed through the browser exactly as it appears in the file. This lends itself rather nicely to viewing SAS tables and listings through web browsers. As an aside, link tags and character styles can also be used between the `<PRE>` and `</PRE>` tags.

**CONCLUSION**

The purpose of this paper is to provide you with 4 SAS macros that function as file management utility programs for the UNIX and Windows operating systems, including example invocations of each macro. To review, these SAS macros are:

- 1.) **concat1** - This SAS macro concatenates all existing output files in a specified UNIX directory.
- 2.) **concat2** - This SAS macro concatenates all existing output files in a specified directory on the Windows operating system. (Includes some added features that do not exist in the macro **concat1**.)
- 3.) **concat3** - This SAS macro concatenates a common page of all existing output files in a specified UNIX directory. (The page number to be concatenated from each file is specified as an input parameter.)
- 4.) **convhtml** - This SAS macro converts an existing SAS output file to an HTML file. (The program that generated the output does not have to be rerun, which preserves any date and time stamps that exist on the pages of the output file.)

While SAS software is widely used in many industries to generate statistical summary tables and data listings in clear tabular formats, it can also be used in place of shell scripting (on UNIX platform) or batch programming (on MS-DOS operating system) to manage these output tables and listings rather effectively.

**REFERENCES**

LeBouton, K. (2000), "Smokin' With UNIX Pipes", *Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference*, 25, 555-558.

SAS is a registered trademark of SAS Institute Inc., in the USA and other countries.

Windows is either a trademark or registered trademark of Microsoft Corporation in the United States and/or other countries.

MS-DOS is either a trademark or registered trademark of Microsoft Corporation in the United States and/or other countries.

® indicates USA registration.

**CONTACT INFORMATION**

Christopher J. Rook  
(610) 917-5089 (W) E-mail: [cjrook@nni.com](mailto:cjrook@nni.com)

Shi-Tao Yeh, Ph. D.  
(610) 917-5883 (W) E-mail: [shitaoyeh@us.sina.com](mailto:shitaoyeh@us.sina.com)