

# Better Safe than Sorry: A SAS<sup>®</sup> Macro to Selectively Back Up Files

Jia Wang, Data and Analytic Solutions, Inc., Fairfax, VA

Zhengyi Fang, Social & Scientific Systems, Inc., Silver Spring, MD

## ABSTRACT

SAS<sup>®</sup> programmers know that they need to regularly back up their SAS programs, data sets, and documentation files, but it's not done as frequently or systematically as it should be-- especially if there are files from hundreds of folders or subfolders every week. The best way to ensure routine file backups is to make it easy to do and easy to document.

This paper presents a handy, easy-to-use, SAS tool that automatically backs up files stored in a hierarchical structure. The tool allows you to select files based on folder, file type, file size, time period of creation, and file ownership. Once your selection is made, by running the SAS program, all files that meet your criteria will be copied automatically to a designated location with the same hierarchical structure as the source files. You can also schedule the backup program to run at a pre-set time and send a notification email with a list documenting the folders and files that have been backed up.

## INTRODUCTION

It is always important to regularly back up your files. Your files can be accidentally deleted or overwritten by yourself or your colleagues. Imagine this scene: when you turn on your computer one morning, you are greeted with a blue screen. Oops, your hard drive or the server has crashed. You realize that your SAS programs and SAS data sets created by your hard work in the last week are all gone. You regret that you did not back up your files. It's no use crying over spilled milk.

When you try to back up files from hundreds of folders or subfolders every week, it is time-consuming because you need to pick out all the folders with newly created or revised files, create the same folders and subfolders in the backup destination, select files you want to back up, and copy all of the selected files to the backup folder. We developed a SAS macro that can back up files easily. The macro has the following three features.

- The macro can automatically create the folders or subfolders in the backup location and copy files to the backup folders by using Windows commands.
- This tool is flexible since it allows you to select files based on folder, file type, file size, time period of creation, and file ownership.
- You can schedule the backup procedure by using the Windows Task Scheduler so that backups will be done as regularly as they should be.

## CREATION OF TEXT FILES WITH FILES INFORMATION

To back up files within a hierarchical structure, the first step in the macro is to create a text file, which contains all the detailed information on the files in a folder, including file name, file size, file creation date, and file owner. See the SAS code below.

```
OPTIONS NOXWAIT;  
%LET SOURCEDIR=P:\PROJECT_A;  
X "DIR &SOURCEDIR /S /L /Q > C:\TEMP\DIR.TXT";  
FILENAME DOSDIR ("C:\TEMP\DIR.TXT");
```

The macro variable &SOURCEDIR specifies the folder to be backed up. The X statement is used to execute a Windows command DIR within a SAS session and redirect a list of files and folders in the source folder to a text file. In the above sample code, a text file called C:\TEMP\DIR.TXT is created. Three options, /S, /L, and /Q, are used in the DIR command.

/S: include all subfolders

/L: use lowercase

/Q: display the owner of the file

Note that submitting an X statement will keep the Command window open and suspend your SAS session while waiting for your command. So the NOXWAIT option is used to close the Command window automatically and return to execute the SAS program without stopping. In the FILENAME statement, the fileref DOSDIR is associated with the external text file, DIR.txt, to access file information.

If multiple folders need to be backed up, you can improve the code as follows.

```
%LET N=2;
%LET SOURCEDIR1=P:\PROJECT_A;
%LET SOURCEDIR2=P:\PROJECT_B;

%DO I=1 %TO &N;
X "DIR &&SOURCEDIR&I /S /L /Q > C:\TEMP\DIR&I..TXT";
%END;
FILENAME DOSDIR (%DO I=1 %TO &N;
                  "C:\TEMP\DIR&I..TXT"
                  %END;);
```

The macro variable N is set to be the total number of folders that you want to back up. &SOURCEDIR1- &SOURCEDIR&N are macro variables for the N folders. The first macro %DO loop creates text files (DIR1- DIR&N) for the N folders. The FILENAME statement with another %DO loop concatenates all of the text files and associates the fileref DOSDIR to them.

## CREATION OF A SAS DATA SET CONTAINING FILE INFORMATION

Once the text files are created, the macro retrieves the file information from the text files and creates a SAS data set containing the file information. See the sample SAS code below.

```
%LET BACKUPDIR=C:\BACKUP; /*Location of backup files*/
DATA FILEINFO;
  INFILE DOSDIR TRUNCOVER FIRSTOBS=4 LENGTH=LEN;
  RETAIN DIRNAME;
  INPUT @;
         LEN1=LEN-14;
         LEN2=LEN-62;

  INPUT @1 VAR1 $10. @15 VAR2 $VARYING300. LEN1 @;
  IF VAR1='Directory' THEN DIRNAME=VAR2;
  ELSE
  INPUT @1 DATE $10. @22 FSIZE $17. @40 OWNER $20. @63 FNAME & $VARYING300.
  LEN2;

  IF FNAME NE ' ' AND FSIZE^='<DIR>';

  FULLNAME=COMPRESS(DIRNAME||'\'||FNAME);
  DIRNEW=COMPRESS("&BACKUPDIR"||'\'||SUBSTR(DIRNAME,4));
  FILENEW=COMPRESS("&BACKUPDIR"||'\'||SUBSTR(FULLNAME,4));
RUN;
```

The fileref, DOSDIR, is referenced in the INFILE statement. Each raw data file that has been referenced can be sequentially read into a data set. The INFILE statement option FIRSTOBS=4 tells SAS to read from the fourth line of the file, since the first three lines only contain the disk's volume label and serial number. The option TRUNCOVER prevents SAS from moving to the next line when the line being read does not have all variables.

There are three INPUT statements in this step. The first INPUT statement just holds the line. Then, two variables, LEN1 and LEN2, are created to determine the length of the varying-length variables, VAR2 and FNAME, respectively. The second INPUT statement creates VAR1 and VAR2 to identify if this line starts with "Directory". A single trailing @ is used in this INPUT statement to hold the line. If a line starts with "Directory", then the code assigns the value of VAR2 to DIRNAME and retains the value. Otherwise, the third INPUT statement will be executed to read the values for creation date (DATE), file size (FSIZE), file owner (OWNER), and file name including the extension (FNAME) from this data line. In case there is a single blank in the file name, an ampersand (&) is added so that the system will continue reading the value. The records with FNAME=' ' or FSIZE='<DIR>' are deleted be-

cause they are for folders rather than files.

The variable FULLNAME is created by concatenating the folder and the file name for the source files. DIRNEW is used to combine the specified root folder stored in the macro variable "BACKUPDIR" with the original folder, so that the hierarchical structure of the backup files is maintained. FILENEW is created by adding the file name to DIRNEW.

Below is an example that shows the transition of the text file into a SAS data set.

```

Directory of P:\Project_A\V2\Team2
07/06/2012 03:32 PM <DIR> S-3\jwang .
07/06/2012 03:32 PM <DIR> S-3\jwang ..
07/03/2012 02:58 PM 35,312 S-3\jwang compest08x.log
07/03/2012 02:58 PM 174,554 S-3\jwang compest08x.lst
07/03/2012 02:47 PM 15,739 S-3\jwang compest08x.sas
07/06/2012 11:57 AM 35,489 S-3\jwang compest09.log
07/06/2012 11:57 AM 56,120 S-3\jwang compest09.lst
07/06/2012 11:47 AM 16,060 S-3\jwang compest09.sas
        6 File(s)        333,274 bytes

Directory of P:\Project_B
07/06/2012 03:44 PM <DIR> S-3\jwang .
07/06/2012 03:44 PM <DIR> S-3\jwang ..
07/06/2012 03:43 PM <DIR> S-3\jwang data2000
07/06/2012 03:44 PM <DIR> S-3\FZfang doc
07/06/2012 03:46 PM <DIR> S-3\jwang prog2000
        0 File(s)        0 bytes

Directory of P:\Project_B\Data2000
07/06/2012 03:43 PM <DIR> S-3\jwang .
07/06/2012 03:43 PM <DIR> S-3\jwang ..
01/11/2012 05:07 PM 394,240 S-3\FZfang data1.sas7bdat
01/11/2012 05:07 PM 13,312 S-3\FZfang data2.sas7bdat
01/11/2012 05:07 PM 1,311,744 S-3\FZfang data3.sas7bdat
        3 File(s)        1,719,296 bytes

Directory of P:\Project_B\DOC
07/06/2012 03:44 PM <DIR> S-3\FZfang .
07/06/2012 03:44 PM <DIR> S-3\jwang ..
08/15/2011 01:33 PM 119,600 S-3\FZfang fe023.pdf
08/15/2011 07:48 AM 107,008 S-3\FZfang painconditions_specs.doc
09/23/2011 02:59 PM 180,529 S-3\FZfang sas report - women_paing_11.pdf
        3 File(s)        407,137 bytes

DATE          FSIZE        OWNER        FNAME          FULLNAME          FILENEW
07/03/2012    35,312       S-3\jwang    compest08x.log P:\Project_A\V2\Team2\compest08x.log C:\BACKUP\Project_A\V2\Team2\compest08x.log
07/03/2012    174,554      S-3\jwang    compest08x.lst P:\Project_A\V2\Team2\compest08x.lst C:\BACKUP\Project_A\V2\Team2\compest08x.lst
07/03/2012    15,739       S-3\jwang    compest08x.sas P:\Project_A\V2\Team2\compest08x.sas C:\BACKUP\Project_A\V2\Team2\compest08x.sas
07/06/2012    35,489       S-3\jwang    compest09.log P:\Project_A\V2\Team2\compest09.log C:\BACKUP\Project_A\V2\Team2\compest09.log
07/06/2012    56,120       S-3\jwang    compest09.lst P:\Project_A\V2\Team2\compest09.lst C:\BACKUP\Project_A\V2\Team2\compest09.lst
07/06/2012    16,060       S-3\jwang    compest09.sas P:\Project_A\V2\Team2\compest09.sas C:\BACKUP\Project_A\V2\Team2\compest09.sas
01/11/2012    394,240      S-3\FZfang    data1.sas7bdat P:\Project_B\Data2000\data1.sas7bdat C:\BACKUP\Project_B\Data2000\data1.sas7bdat
01/11/2012    13,312       S-3\FZfang    data2.sas7bdat P:\Project_B\Data2000\data2.sas7bdat C:\BACKUP\Project_B\Data2000\data2.sas7bdat
01/11/2012    1,311,744    S-3\FZfang    data3.sas7bdat P:\Project_B\Data2000\data3.sas7bdat C:\BACKUP\Project_B\Data2000\data3.sas7bdat
08/15/2011    119,600      S-3\FZfang    fe023.pdf       P:\Project_B\DOC\fe023.pdf          C:\BACKUP\Project_B\DOC\fe023.pdf
08/15/2011    107,008      S-3\FZfang    painconditions_specs.doc P:\Project_B\DOC\painconditions_specs.doc C:\BACKUP\Project_B\DOC\painconditions_specs.doc
09/23/2011    180,529      S-3\FZfang    sas report - women_paing_11.pdf P:\Project_B\DOC\sasreport-women_paing_11.pdf C:\BACKUP\Project_B\DOC\sasreport-women_paing_11.pdf

```

## SELECTION OF FILES TO BACK UP

The files you want to back up will be selected based on the file creation date, file owner, and file type.

```

%LET BEGDATE=MM/DD/YYYY; /*Search all files created since MM/DD/YYYY*/
%LET FILETYPE=%STR('sas7bdat', 'sas', 'log', 'lst'); /*Case insensitive. Use
'all' if back up all types of files*/
%LET OWNER=%STR('XXX'); /*All in upper case. Use 'ALL' if back up files re-
gardless of owner. For multiple owners, use %STR('XXX', 'YYY') */

```

```

DATA BACKUP;
  SET FILEINFO;
  IF INPUT (DATE, MMDDYY10.) >= INPUT ("&BEGDATE", MMDDYY10.)
  %IF %UPCASE(&OWNER) NE 'ALL' %THEN %DO;
    AND UPCASE(OWNER) IN (&OWNER)
  %END;
  %IF %UPCASE(&FILETYPE) NE 'ALL' %THEN %DO;

```

```

        AND SUBSTR(FNAME, INDEX(FNAME, ".") + 1) IN (&FILETYPE)
    %END;
;
RUN;

```

The records in the data set FILEINFO are subset to those that meet the criteria that you specified for file creation date, file owner, and file type. You can also add one more criterion based on file size as shown by the code below.

```
IF &FSIZEMIN<=INPUT(FSIZE,COMMA8.)<=&FSIZEMAX
```

## COPY FILES

After the data set BACKUP is created, SAS copies the files that you selected into the designated folder &BACKUPDIR.

```

PROC SQL NOPRINT;
    SELECT COUNT(*) INTO: NUMOBS FROM BACKUP;
    SELECT FULLNAME INTO: FILE1-:FILE&NUMOBS FROM BACKUP;
    SELECT DIRNEW INTO: DIR1-:DIR&NUMOBS FROM BACKUP;
QUIT;

%DO I=1 %TO &NUMOBS;
    %LET RC = %SYSFUNC(FILENAME(FILEREF, &&DIR&I)) ;
    %IF NOT %SYSFUNC(FEXIST(&FILEREF)) %THEN %DO ;
        %SYSEXEC MD &&DIR&I ;
        %PUT NOTE: THE DIRECTORY "&&DIR&I" HAS BEEN CREATED. ;
    %END ;
    X "COPY &&FILE&I &&DIR&I";
%END;

```

First, PROC SQL is used to (1) determine how many files should be backed up. The number of files is stored in a macro variable &NUMOBS; (2) create a set of macro variables FILE1-FILE&NUMOBS, each storing a single FULLNAME value; (3) create macro variables DIR1-DIR&NUMOBS, each with a DIRNEW value.

Next, the macro checks if the folder exists before copying each file. If it does not exist, a new folder will be created using another Windows command MD within the %SYSEXEC statement.

Finally, each file is copied to a specified folder using an X statement to submit a COPY command.

The %DO loop is used to repeat the last two actions for each file.

## SCHEDULE THE BACKUP TASK

Now, suppose you want to back up files automatically.

First, create a one-line batch file (see sample code below). BACKUP.sas is the SAS file that contains the backup macro.

```
"c:\program files\SAS\sasfoundation\9.2\sas.exe" c:\backup.sas
```

Next, set up a task in the Windows Task Scheduler to run the batch file automatically. When the batch file is triggered, the SAS software will be opened and the backup macro is run in batch mode immediately.

If you want to back up files weekly without specifying the macro variable &BEGDATE every time, you can modify the code for selection based on the file creation dates to this:

```
IF INPUT( DATE, MMDDYY10.) >= INPUT("&SYSDATE", MMDDYY10.) - 7
```

If you want to back up files monthly, then just change 7 to 31.

## SEND EMAILS

Once the backup is completed, SAS can automatically send out a notification email to relevant staff with an attachment containing a list of folders and files that have been backed up. See the SAS code below.

```
ODS HTML BODY="C:\BACKUP_&SYSDATE..XLS";
```

```

TITLE1 "Files(&FILETYPE) created by &OWNER have been copied to &BACKUPDIR";
PROC PRINT DATA=BACKUP NOOBS LABEL;
  VAR DATE FSIZE OWNER FULLNAME FILENEW;
  LABEL DATE='Date modified'
         FSIZE='File size'
         OWNER='Owner'
         FULLNAME='Source files'
         FILENEW='Backup files';
RUN;
ODS HTML CLOSE;

%LET YOUREMAIL="XXX@NESUG.COM";
%LET TOEMAIL  ="YYY@NESUG.COM";
%LET CCEMAIL  ="ZZZ@NESUG.COM";

FILENAME OUTPUT EMAIL SUBJECT= "Backup is completed"
                FROM= &YOUREMAIL
                TO= (&TOEMAIL)
                CC= (&CCEMAIL)
                ATTACH= "C:\BACKUP_&SYSDATE..XLS";

DATA _NULL_;
  FILE output;
  PUT "Attached is a list of files which have been copied to &BACKUPDIR.";
RUN;

```

## CONCLUSIONS

Regular backup of files is a necessary but sometimes time-consuming task, especially when you are busy with other responsibilities. In this paper, we present a simple macro routine which can help you quickly back up files within a hierarchical structure. A major advantage of this tool is the flexibility of being able to specify different types of selection criteria. In addition, the SAS macro can be run automatically in the background at a scheduled time and no further user input is required when the code is executed.

## ACKNOWLEDGMENTS

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 registered trademarks or trademarks of their respective companies.

The authors would like to thank Nancy Odaka, Leif Karell, Lita Manuel, Valerie Aschenbach, and Adrien Ndi-umwami of Social & Scientific Systems, Inc. and Paul Gorrell of IMPAQ International, LLC for their helpful comments.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please contact the authors at:

Jia Wang  
 Data and Analytic Solutions, Inc.  
 3057 Nutley Street  
 Fairfax, VA 22031  
 Email: [jwang@s-3.com](mailto:jwang@s-3.com)

Zhengyi Fang  
 Social & Scientific System, Inc.  
 8757 Georgia Avenue, 12th floor  
 Silver Spring, MD 20910  
 Email: [zfang@s-3.com](mailto:zfang@s-3.com)