Using SAS® to Locate and Rename External Files

Lu Gan, Pharmaceutical Product Development, LLC, Austin, TX

ABSTRACT

When reading in external data into SAS, a program needs to specify the data file location and file name in the related importing statements. If the raw data file is named with a timestamp, the import program will require an extra step which is updating the file name every time before it's run. This will be a hassle if the import is needed on a routine schedule and/or there are multiple external data files. Therefore, a SAS program can be written to eliminate the repeating manual update. This approach includes two steps, first searching for the latest data file in the specific file directory, then renaming it by removing the specific timestamp. After execution of the above steps, the import program will be able to read in the data files directly.

The examples and syntax illustrated are in SAS 9.2. This paper assumes that the reader has a basic understanding of DATA step programming and the macro language.

INTRODUCTION

Time-saving is always a goal programmers like to achieve in daily work. Instead of manually renaming external data files, using SAS to programmatically do it undoubtedly improves efficiency. There are various types of external data files SAS can import; in this paper text file(.txt) is used as an example.

The approach starts with the definitions of several macro variables which are going to be used in the sample code.

```
** 'filepath' is the file directory storing the file(s) which will be renamed;
%LET filepath= %NRBQUOTE(U:\PHARM\);

** 'delimiter' is the delimiter used in filenames;
%LET delimiter= %NRBQUOTE(_);

** 'file_extension' is the file extension of the external data file(s);
%LET file_extension= %NRBQUOTE(.txt);

** 'filename_parts' tells how many part(s) in the filename excluding file
extension. Example: in filename 'Sample_yyyymmdd_hhmm_Filen.txt', there are 4 parts
delimited by the delimiter '_';
%LET filename_parts= 4;

** 'except_file' is a list of file(s) that are with same file extension but don't
want to be renamed;
%LET except_file= %NRBQUOTE(Except1,Except2,Sample_File3);
```

Except1.txt	1 KB	Text Document	9/1/2012 8:28 PM
Except2.txt	1 KB	Text Document	9/1/2012 8:28 PM
Sample_20120901_2027_File1.txt	1 KB	Text Document	9/1/2012 8:27 PM
Sample_20120901_2027_File2.txt	1 KB	Text Document	9/1/2012 8:27 PM
Sample_20120901_2027_File3.txt	1 KB	Text Document	9/1/2012 8:27 PM
Sample_20120901_2027_File4.txt	1 KB	Text Document	9/1/2012 8:27 PM
Sample_20120902_1526_File2.txt	1 KB	Text Document	9/2/2012 3:26 PM
Sample_20120902_1526_File4.txt	1 KB	Text Document	9/2/2012 3:26 PM
Sample_20120902_1526_File5.txt	1 KB	Text Document	9/2/2012 3:26 PM
Sample_20120902_1526_File6.txt	1 KB	Text Document	9/2/2012 3:26 PM
Sample_File1.txt	1 KB	Text Document	9/1/2012 11:18 PM
Sample_File2.txt	1 KB	Text Document	9/1/2012 11:18 PM
Sample_File3.txt	1 KB	Text Document	9/1/2012 8:27 PM
Sample_File5.txt	1 KB	Text Document	9/1/2012 11:18 PM
Sample_File7.txt	1 KB	Text Document	9/1/2012 11:18 PM

Display 1. List of TXT files in directory specified by &filepath macro variable

GET ALL FILE(S) INFORMATION IN THE FOLDER

There are multiple ways for SAS programs to interact with the world outside of SAS. FILENAME statement associates a SAS fileref with an external file or an output device, disassociates a fileref and external file, or lists attributes of external files. It's used here to get all file(s) information in the above folder.

```
FILENAME indat PIPE "dir ""&filepath.*&file_extension"" ";
DATA all_file;
   INFILE indat TRUNCOVER;
   INPUT fldt $ 1-11 fltm $ 12-22 flsiz $ 25-38 flnm $ 39-100;
RUN;
```

SAS System Viewer - [all_file.sas7bdat]

File   Edit   View   Window   Help

	fldt	fltm	flsiz	flnm
1	Volume in	drive U i	ers	
2	Volume Ser	ial Numbe	6501-0A78	
3				
4	Directory	of U:\SCS		
5				
6	09/02/2012	03:26 PM	11	Sample_20120902_1526_File2.txt
7	09/01/2012	11:18 PM	10	Sample_File1.txt
8	09/01/2012	11:18 PM	10	Sample_File2.txt
9	09/01/2012	08:27 PM	9	Sample_File3.txt
10	09/01/2012	11:18 PM	10	Sample_File7.txt
11	09/02/2012	03:26 PM	12	Sample_20120902_1526_File4.txt
12	09/02/2012	03:26 PM	12	Sample_20120902_1526_File5.txt
13	09/01/2012	08:28 PM	9	Except2.txt
14	09/01/2012	08:28 PM	9	Except1.txt
15	09/02/2012	03:26 PM	12	Sample_20120902_1526_File6.txt
16	09/01/2012	08:27 PM	9	Sample_20120901_2027_File1.txt
17	09/01/2012	08:27 PM	9	Sample_20120901_2027_File2.txt
18	09/01/2012	08:27 PM	9	Sample_20120901_2027_File3.txt
19	09/01/2012	08:27 PM	9	Sample_20120901_2027_File4.txt
20	09/01/2012	11:18 PM	10	Sample_File5.txt
21		15 File(	15	0 bytes
22		0 Dir(s	10,319,25	0 bytes free

Output 1. Data set 'all_file'

CLEAN THE FILE(S) INFORMATION

As you might already notice, data set 'All_File' contains some useless observations which are going to be removed. Besides, in macro variable &except_file, there are several text files that we don't want to rename so we also want to exclude them from further processing. The following macro %Clean_filenames_Info is created to do the 'cleaning' work.

```
%MACRO Clean_filenames_Info;

%LET numf=%EVAL(1+%SYSFUNC(COUNTC("&except_file.", ',', t)));

DATA except_files;
    LENGTH flnm $60.;
    %DO i=1 %TO &numf.;
        %LET _flnm= %SCAN(&except_file., &i, ',');
        flnm= STRIP("&_flnm"||"&file_extension");
        OUTPUT;
    %END;
RUN;

PROC SQL;
CREATE TABLE all_file_info AS
SELECT *
FROM all_file
WHERE NOT (INDEXC(fldt,'Check','Directory') > 0 OR MISSING(fldt)) AND
      STRIP(flnm) NOT IN (SELECT flnm FROM except_files);
QUIT;
%MEND Clean_filenames_Info;
```

SAS System Viewer - [all_file_info.sas7bdat]

File Edit View Window Help
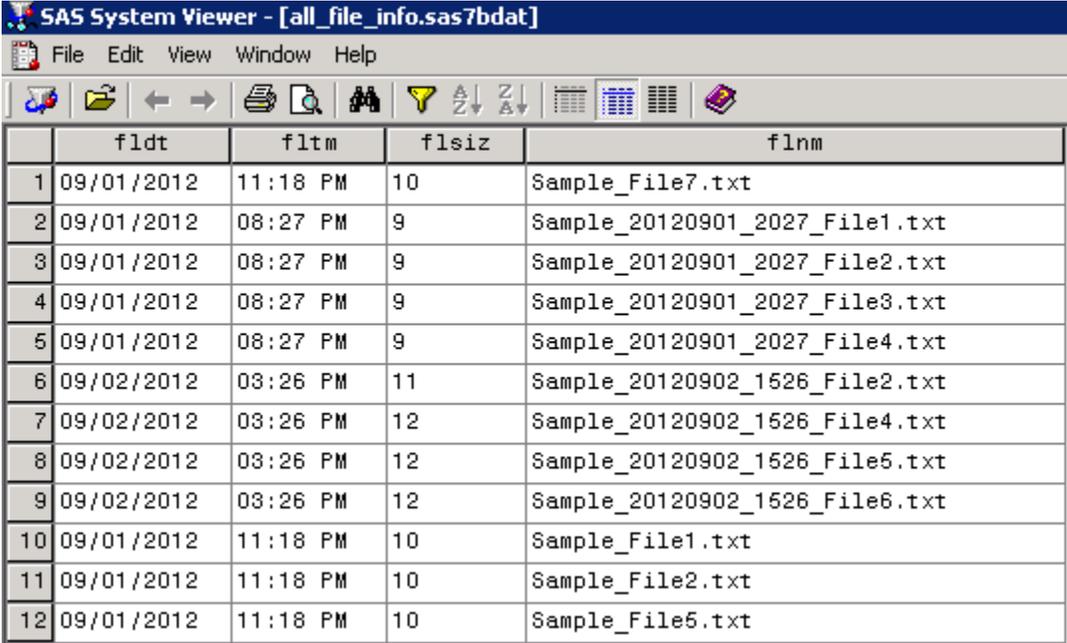
	fldt	fltm	flsiz	flnm
1	09/01/2012	11:18 PM	10	Sample_File7.txt
2	09/01/2012	08:27 PM	9	Sample_20120901_2027_File1.txt
3	09/01/2012	08:27 PM	9	Sample_20120901_2027_File2.txt
4	09/01/2012	08:27 PM	9	Sample_20120901_2027_File3.txt
5	09/01/2012	08:27 PM	9	Sample_20120901_2027_File4.txt
6	09/02/2012	03:26 PM	11	Sample_20120902_1526_File2.txt
7	09/02/2012	03:26 PM	12	Sample_20120902_1526_File4.txt
8	09/02/2012	03:26 PM	12	Sample_20120902_1526_File5.txt
9	09/02/2012	03:26 PM	12	Sample_20120902_1526_File6.txt
10	09/01/2012	11:18 PM	10	Sample_File1.txt
11	09/01/2012	11:18 PM	10	Sample_File2.txt
12	09/01/2012	11:18 PM	10	Sample_File5.txt

Output 2. Data set 'all_file_info'-- contains useful information that will be needed in subsequent steps

PROCESS THE FILE NAME INFORMATION
With all the file information available, we want to follow naming convention to find the filename(s) the file(s) will be renamed to. In this example, we assume original filename 'Sample_yyyymmdd_hhmm_Filen.txt' will be renamed to 'Sample_Filen.txt'.

```
DATA file_info;
```

```
       SET all_file_info;
       FORMAT file_dt YYMMDD10. file_tm TIME5.;
       LENGTH filetime $5. new_file_name $50.;

       CALL MISSING(new_file_name, filetime, file_dt, file_tm);

       file_dt = INPUT(fldt, MMDDYY10.);
       IF INDEX(fltm,'PM') THEN  file_tm = INPUT(SUBSTR(fltm,1,5), TIME5.)+43200;
       ELSE   file_tm = INPUT(SUBSTR(fltm,1,5), TIME5.);
       filetime= COMPRESS(PUT(file_tm, ??TIME5.), ':');

       ARRAY flnames{&filename_parts} $20.;
       DO i=1 TO DIM(flnames);
          flnames(i)= SCAN(TRANWRD(flnm, "&file_extension", ""), i, "&delimiter");
         IF INPUT(flnames(i), ??YYMMDD10.)= file_dt OR flnames(i)= filetime THEN
               flnames(i)='';
       END;

       DO i=1 TO DIM(flnames);
           new_file_name= CATX("&delimiter", new_file_name, flnames(i));
       END;
       new_file_name= STRIP(STRIP(new_file_name)|| "&file_extension");

       KEEP flnm file_dt file_tm new_file_name;
   RUN;
```
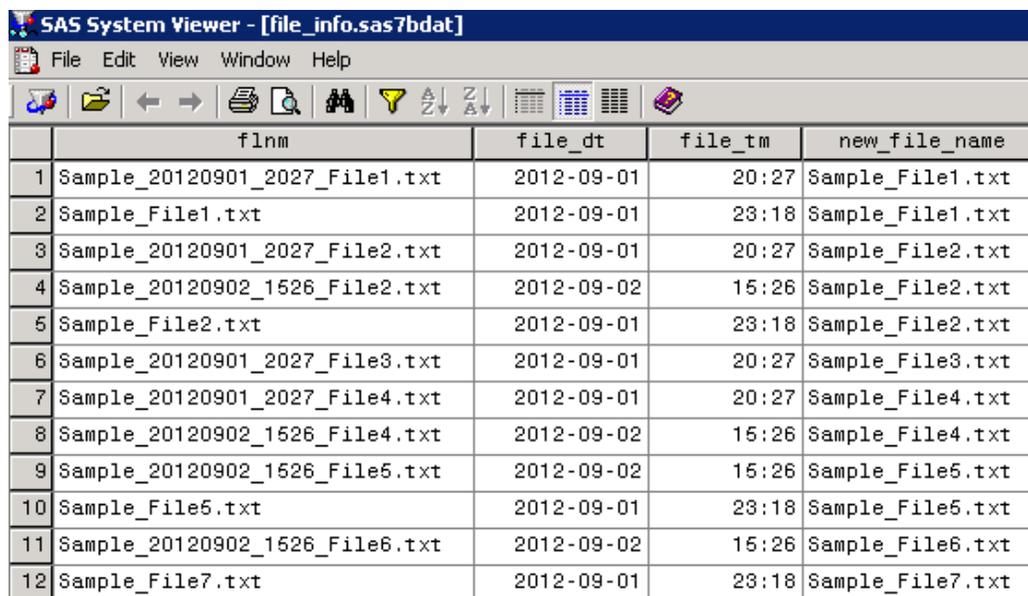
	flnm	file_dt	file_tm	new_file_name
1	Sample_20120901_2027_File1.txt	2012-09-01	20:27	Sample_File1.txt
2	Sample_File1.txt	2012-09-01	23:18	Sample_File1.txt
3	Sample_20120901_2027_File2.txt	2012-09-01	20:27	Sample_File2.txt
4	Sample_20120902_1526_File2.txt	2012-09-02	15:26	Sample_File2.txt
5	Sample_File2.txt	2012-09-01	23:18	Sample_File2.txt
6	Sample_20120901_2027_File3.txt	2012-09-01	20:27	Sample_File3.txt
7	Sample_20120901_2027_File4.txt	2012-09-01	20:27	Sample_File4.txt
8	Sample_20120902_1526_File4.txt	2012-09-02	15:26	Sample_File4.txt
9	Sample_20120902_1526_File5.txt	2012-09-02	15:26	Sample_File5.txt
10	Sample_File5.txt	2012-09-01	23:18	Sample_File5.txt
11	Sample_20120902_1526_File6.txt	2012-09-02	15:26	Sample_File6.txt
12	Sample_File7.txt	2012-09-01	23:18	Sample_File7.txt

Output 3. Data set 'file_info' -- include file name, date and time and the new file name if a file is renamed.

Now we find out which files are going to be renamed. Windows® operating system doesn't allow a file to be renamed if file with same name already exists, therefore, before renaming, we want to identify those existing files with same name and flag them for deletion while keep the rest for renaming.

```
   DATA delete_files rename_files;
      SET file_info;
      BY new_file_name flnm file_dt file_tm;
      IF FIRST.new_file_name NE LAST.new_file_name AND new_file_name= flnm THEN
        DO; CALL SYMPUT('existing_file', '1'); OUTPUT delete_files; END;
      ELSE
        DO; OUTPUT rename_files; END;
```

```
RUN;
```


Output 4. Data set 'rename_files'-- includes all existing files that will be renamed

Output 5. Data set 'delete_files'-- includes all existing files that will be deleted and replaced thereafter

DELETE EXISITING FILE(S) WHICH WILL BE REPLACED

We use CALL EXECUTE to call macro %delete_old_file to delete the files stored in data set 'delete_files':

```
%MACRO delete_old_file(file) ;
   %IF %SYSFUNC(FILEEXIST("&filepath.&file")) %THEN %DO;
      %SYSEXEC DEL "&filepath.&file";
      %IF &SYSRC= 0 %THEN %DO;
       %PUT ALERT_I: File &file was successfully deleted.;
    %END;
    %ELSE %DO;
         %PUT ALERT_R: File &file was NOT successfully deleted. SYSRC=   &SYSRC;
    %END;
    %END;
    %ELSE %DO;  %PUT ALERT_R: &filepath.&file NOT FOUND.; %END;
%MEND delete_old_file;

%MACRO delete_existing_file;
%IF &existing_file= 1 %THEN %DO;
   DATA _NULL_;
    SET delete_files;
    CALL EXECUTE('%delete_old_file(file='||strip(flnm)||');');
    RUN;
```

5

```
%END;
%MEND delete_existing_file;
```

```
INFORMATIONAL: ALERT_I: File Sample_File1.txt was successfully deleted.
INFORMATIONAL: ALERT_I: File Sample_File2.txt was successfully deleted.
INFORMATIONAL: ALERT_I: File Sample_File5.txt was successfully deleted.
```

Display 2. SAS log showing what files have been deleted

|			
Except1.txt	Text Document	9/1/2012 8:28 PM	1 KB
Except2.txt	Text Document	9/1/2012 8:28 PM	1 KB
Sample_20120901_2027_File1.txt	Text Document	9/1/2012 8:27 PM	1 KB
Sample_20120901_2027_File2.txt	Text Document	9/1/2012 8:27 PM	1 KB
Sample_20120901_2027_File3.txt	Text Document	9/1/2012 8:27 PM	1 KB
Sample_20120901_2027_File4.txt	Text Document	9/1/2012 8:27 PM	1 KB
Sample_20120902_1526_File2.txt	Text Document	9/2/2012 3:26 PM	1 KB
Sample_20120902_1526_File4.txt	Text Document	9/2/2012 3:26 PM	1 KB
Sample_20120902_1526_File5.txt	Text Document	9/2/2012 3:26 PM	1 KB
Sample_20120902_1526_File6.txt	Text Document	9/2/2012 3:26 PM	1 KB
Sample_File3.txt	Text Document	9/2/2012 7:05 PM	1 KB
Sample_File7.txt	Text Document	9/1/2012 11:18 PM	1 KB

Display 3. Files in the original file folder after execution of macro `delete_existing_file`

RENAME THE FILE(S)

After the above proceeding steps, finally we are ready to rename the file . Couple of things we want to consider are:
1> If more than one file will be renamed to the same name, we want to use the latest file—we assume the file with the latest timestamp in filenames will be needed. 2> Check again to make sure file(s) in '&except_file' will not be renamed.

```
PROC SORT DATA= rename_files;   BY new_file_name flnm file_dt file_tm;   RUN;

DATA to_rename_files;
    SET rename_files;
    BY new_file_name flnm file_dt file_tm;
    IF LAST.new_file_name;
RUN;
```
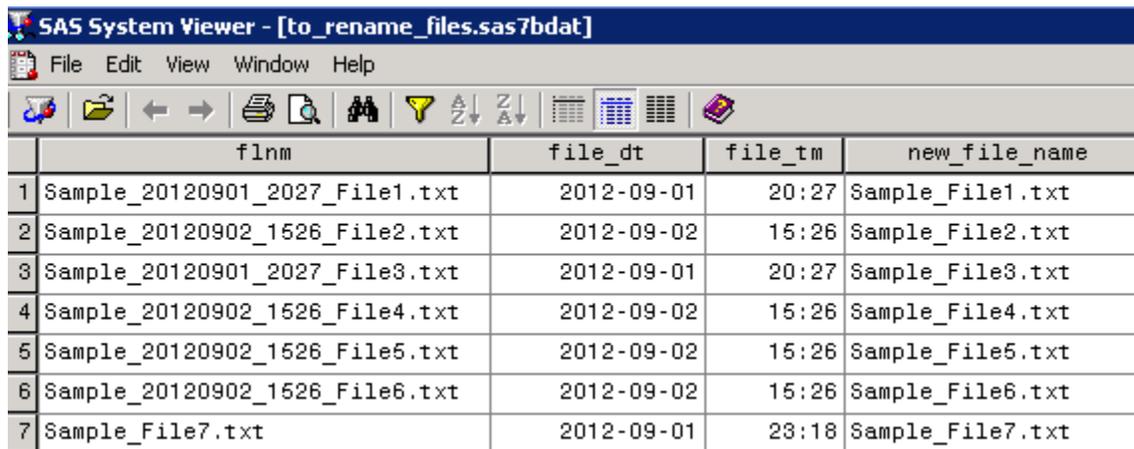
SAS System Viewer - [to_rename_files.sas7bdat]

File Edit View Window Help

	flnm	file_dt	file_tm	new_file_name
1	Sample_20120901_2027_File1.txt	2012-09-01	20:27	Sample_File1.txt
2	Sample_20120902_1526_File2.txt	2012-09-02	15:26	Sample_File2.txt
3	Sample_20120901_2027_File3.txt	2012-09-01	20:27	Sample_File3.txt
4	Sample_20120902_1526_File4.txt	2012-09-02	15:26	Sample_File4.txt
5	Sample_20120902_1526_File5.txt	2012-09-02	15:26	Sample_File5.txt
6	Sample_20120902_1526_File6.txt	2012-09-02	15:26	Sample_File6.txt
7	Sample_File7.txt	2012-09-01	23:18	Sample_File7.txt

Output 6. Data set to_rename_files

```
PROC SQL;
CREATE TABLE to_rename_files2 AS
```

6

```
SELECT *
FROM to_rename_files
WHERE new_file_name NOT IN (SELECT flnm FROM except_files);
QUIT;
```

SAS System Viewer - [to_rename_files2.sas7bdat]

File   Edit   View   Window   Help

	flnm	file_dt	file_tm	new_file_name
1	Sample_20120901_2027_File1.txt	2012-09-01	20:27	Sample_File1.txt
2	Sample_20120902_1526_File2.txt	2012-09-02	15:26	Sample_File2.txt
3	Sample_20120902_1526_File4.txt	2012-09-02	15:26	Sample_File4.txt
4	Sample_20120902_1526_File5.txt	2012-09-02	15:26	Sample_File5.txt
5	Sample_20120902_1526_File6.txt	2012-09-02	15:26	Sample_File6.txt
6	Sample_File7.txt	2012-09-01	23:18	Sample_File7.txt

Output 7. Data set to_rename_files2

Function RENAME() is available since SAS version 9.2. Its purpose is to rename a member of a SAS library, an entry in a SAS catalog, an external file, or a directory. It is used here to rename the file(s).

```
%MACRO file_rename(from_filename, to_filename);
DATA _NULL_;

   rc= RENAME("&filepath.&from_filename", "&filepath.&to_filename", 'file');

   IF rc NE 0 THEN DO;
    PUT "ALERT_" "R:Rename of file &from_filename to &to_filename unsuccessful."
        rc=;
     CALL SYMPUT('err','1');
   END;
    ELSE PUT "ALERT_I: Rename of file &from_filename to &to_filename  successful.";
RUN;
%MEND file_rename;

DATA _NULL_;
    SET to_rename_files2;
    CALL EXECUTE('%file_rename(from_filename='||STRIP(flnm)||', to_filename='||
        STRIP(new_file_name)||');');
RUN;
```

```
INFORMATIONAL: ALERT_I: Rename of file Sample_20120901_2027_File1.txt to Sample_File1.txt successful.
INFORMATIONAL: ALERT_I: Rename of file Sample_20120902_1526_File2.txt to Sample_File2.txt successful.
INFORMATIONAL: ALERT_I: Rename of file Sample_20120902_1526_File4.txt to Sample_File4.txt successful.
INFORMATIONAL: ALERT_I: Rename of file Sample_20120902_1526_File5.txt to Sample_File5.txt successful.
INFORMATIONAL: ALERT_I: Rename of file Sample_20120902_1526_File6.txt to Sample_File6.txt successful.
INFORMATIONAL: ALERT_I: Rename of file Sample_File7.txt to Sample_File7.txt successful.
```

Display 4. SAS Log showing what files are deleted or renamed

Except1.txt	1 KB	Text Document	9/1/2012 8:28 PM
Except2.txt	1 KB	Text Document	9/1/2012 8:28 PM
Sample_20120901_2027_File2.txt	1 KB	Text Document	9/1/2012 8:27 PM
Sample_20120901_2027_File3.txt	1 KB	Text Document	9/1/2012 8:27 PM
Sample_20120901_2027_File4.txt	1 KB	Text Document	9/1/2012 8:27 PM
Sample_File1.txt	1 KB	Text Document	9/1/2012 8:27 PM
Sample_File2.txt	1 KB	Text Document	9/2/2012 3:26 PM
Sample_File3.txt	1 KB	Text Document	9/1/2012 8:27 PM
Sample_File5.txt	1 KB	Text Document	9/2/2012 3:26 PM
Sample_File7.txt	1 KB	Text Document	9/1/2012 11:18 PM
Sample_File4.txt	1 KB	Text Document	9/2/2012 3:26 PM
Sample_File6.txt	1 KB	Text Document	9/2/2012 3:26 PM

Display 5. List of TXT files in the same folder after all proceeding steps

CONCLUSION

This paper described a way to locate and rename external file(s) using SAS data steps, macro languages along with several system functions and commands. With the automation of renaming, users can add other necessary proceeding and succeeding steps to a batch job to fully automate external data import activities. The purpose of this paper is trying to present the idea of expanding the capabilities of the SAS system beyond basic data manipulation so as to achieve more efficient and more automated solutions to the day-to-day programming activities.

REFERENCES

Schacherer, C(2011) The FILENAME Statement: Interacting with the world outside of SAS®

SAS(R) 9.2 Language Reference: Dictionary, Fourth Edition Cary, NC: SAS Institute Inc.

ACKNOWLEDGEMENTS

I'd like to thank my coworker Ranjan Karmakar for discussing with me when I had the idea about this approach. I also appreciate the support from the programming management team at Pharmaceutical Product Development, LLC.

CONTACT INFORMATION

Your comments and questions are valued and welcomed. Please contact the author at:

    Author Name: Lu Gan
    Enterprise: PPD
    Address: 7901 E. Riverside Drive
    City, State ZIP: Austin, TX 78744
    Email: Lu.Gan@ppdi.com
    Work Phone: (512)747-5715