Paper 054-2007

Functionsize: Process your External Files

Author, Linda Libeg, Westat, Rockville, MD

ABSTRACT

There are many instances where all files in a directory need to be processed in exactly the same manner. If the files in the directory are not assigned using an incremental naming convention, or if the number of files in the directory change with each request, processing the files can be problematic.

This paper helps to resolve this problem by using SAS® external file functions. These functions can open a directory, determine the number of members of the directory, identify the file names that can be used to process the data, and close the directory. File processing can be done dynamically without ever knowing the physical file name.

INTRODUCTION

Functions are predefined procedures that have zero or more arguments and return a value.

```
Syntax: Value = function name;
                or
        Value = function name(argument<+argument>);
```

Similarly, SAS external file functions perform some action using an argument and return a value to the user. This paper focuses on using the SAS external file functions to process data files.

THE TASK

Data are collected on respondents who request survey information and saved in an Excel spreadsheet. The client has requested that this information be combined into one SAS dataset and delivered to them. Although these files are date stamped, they were not created every day and the date stamp was not always consistently created. Therefore, I am not able to use a macro and loop through the process 'x' number of times. I could assign each file to a macro call. However, this is cumbersome. Below is an example of some of the file names (mail 011006, mail010206, and mail032006) that need to be processed.

```
%macro probmac(exfile);
        proc import…
        run;
%mend;

%probmac(mail010206)
%probmac(mail032006)
%probmac(mail02152006)
%probmac(mail 011006)
%probmac(mailapr2006)
```

THE SOLUTION: SAS EXTERNAL FILE FUNCTIONS

A SAS external file function returns a value after performing some action involving the assigned arguments. %SYSFUNC and %QSYSFUNC are macro functions that can be used to call the SAS external file functions in open code or within a macro. The difference between these functions is that %QSYSFUNC conceals special characters and mnemonics and %SYSFUNC does not.

The following SAS external file functions were used to process the files in my program and %SYSFUNC was used to call these external file functions.

DOPEN:
This external file function opens a directory using a fileref as the argument and returns a numeric directory identifier value. If the value returned is a 0, then the directory was not found. Otherwise, a returned value greater than 0 identifies the opened directory and can be used in other SAS external file functions.

```
/* "dirid" holds the value that identifies the directory */
filename exdir 'c:\saspaper\sgf';
%let dirid = %sysfunc(DOPEN(exdir));
```

DNUM:
This external file function returns the number of members in a directory using the directory identifier returned by the DOPEN function.

```
/* "dircnt" contains the number of files in the directory */
%let dircnt = %sysfunc(DNUM(&dirid));
```

DREAD:
This external file function uses the identifier returned by the DOPEN function and returns a directory member name. By using the value of the total number of files in a directory returned by DNUM, all files in a directory can be processed in a loop.

```
/* "dirread" holds the name of the "&ith" file in the directory */
%do i = 1 %to &dircnt;
        %let dirread = %sysfunc(DREAD(&dirid,&i));
        %put dirread = &dirread;
%end;
```

DCLOSE:
This external file function closes the directory opened previously by DOPEN. Zero is returned if successful. Otherwise, if unsuccessful, a value other than zero is returned.

```
%let rc =%sysfunc(DCLOSE(&dirid));
```

The SAS program that uses these SAS external file functions is included in the appendix of this paper. The macro opens the directory specified in the filename statement with DOPEN and uses DNUM to identify the number of files in this directory. In a loop based on the number returned by DNUM, the code opens each file name in the directory returned by DREAD, imports the data, and appends the data to a master file. Finally, DCLOSE is used to close the directory.

ADDITIONAL SAS EXTERNAL FILE FUNCTIONS

DCREATE:
The DCREATE SAS external file function creates a new external Directory in the program code.

```
%let Newdir = %sysfunc(DCREATE(directory-name<,parent directory>));
%let newdir = %sysfunc(DCREATE(NEWDIR,c:\saspaper));
```

DOPTNUM:
The DOPTNUM SAS external file function returns the count of information items available for an opened directory. Using the Windows XP operating system and processing my code in SAS 9.1.3, I only received one information item (DOPTNUM returned a value of 1). This count may vary depending on the operating system that is being used.

```
/* &dirid is defined by the returned value of the DOPEN SAS external file function */
%let infocnt = %sysfunc(DOPTNUM(&dirid));
```

DOPTNAME:
The DOPTNAME SAS external file function returns the directory attribute for an information item. Using the Windows XP operating system and processing my code in SAS 9.1.3, DOPTNAME returned the attribute "Directory".

```
%do i = 1 %to &infocnt;
   %let doption = %sysfunc(DOPTNAME(&dirid,&i));
   %put Directory information = &doption;
%end;
```

DINFO:
The DINFO SAS external file function returns the directory information pertaining to one of the available information items for the directory. Using the Windows XP operating system and processing my code in SAS 9.1.3, DINFO returned the full directory path name.

```
%do i = 1 %to &infocnt;
   %let doption = %sysfunc(DOPTNAME(&dirid,&i));
   %let dirinfo = %sysfunc(DINFO(&dirid,&doption));
%end;
```

FILEEXIST:
The FILEEXIST SAS External Function verifies the existence of an external file by its physical name. A fully qualified physical filename must always be used. If the external file exists, a value of 1 is returned and if the external file does not exist, a value of 0 is returned.

```
%if %sysfunc(FILEEXIST("c:\saspaper\054-2007.doc")) %then %do;
```

CONCLUSION
SAS external file functions provide a convenient way to process data files systematically without worrying about physical file names. This approach helps to reduce errors that may occur when manually processing data. This code can easily be adapted to other work assignments. In fact, this approach was also used to FTP files to a secure account.

REFERENCES
SAS Institute Inc. 2006. **SAS OnlineDoc® 9.1.3.** Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENT
I would like to thank Duke Owen, Mike Rhoads, and Michael Raithel for their guidance and review of this paper.

DISCLAIMER
The contents of this paper are the work of the author(s) and do not necessarily represent the opinions, recommendations, or practices of Westat.

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

Linda Libeg
Westat
1650 Research Blvd – RA1252
Rockville, MD 20850
Work Phone: (301) 738-3542
Email: LindaLibeg@westat.com

TRADEMARKS

APPENDIX: FUNCTIONSIZE.SAS

```
/* Directory containing EXCEL files to process */
filename exdir 'c:\saspaper\SGF';

/* Directory containing SAS datasets and the appended file */
libname sasdir 'c:\saspaper\sasdata';

Options mprint validvarname=upcase;

/* Macro opens a directory, returns the number of files, and processes each file */
/* For easier code readability, Error Handling is not included in this code */
%macro sysmac;
    %if %sysfunc(FILEEXIST("c:\saspaper\054-2007.doc")) %then %put FILE EXISTS;
    %let dirid = %sysfunc(DOPEN(exdir));
    %put dirid =&dirid;
    %let dircnt = %sysfunc(DNUM(&dirid));
    %put dircnt = &DIRCNT;
    %do i = 1 %to &dircnt;
        %let dirread = %sysfunc(DREAD(&dirid,&i));
        %put dirread = &dirread;

        PROC IMPORT OUT= mail&i
                    DATAFILE= "c:\saspaper\SGF\&dirread"
                    DBMS=EXCEL2000 REPLACE;
             GETNAMES=YES;
        RUN;

       data sasdir.mail&i(keep=id_number lastname firstname address city state
                              zipcode telephone lettertype);
           set mail&i(rename=(city=oldcity address=oldaddr
                             telephone_number=telephone));
           length firstname lastname city lettertype $25 address $40;
           lastname = left(last_name);
           firstname = left(first_name);
           city = left(oldcity);
           address = left(oldaddr);
           lettertype = left(kind_of_letter);
        run;

        %if &i=1 %then %do;
            data sasdir.mailall;
                set sasdir.mail&i;
            run;
        %end;
        %else %do;
            proc append base=sasdir.mailall data=sasdir.mail&i;
            run;
        %end;

    %end;

    %let rc = %sysfunc(DCLOSE(&dirid));
    %put rc = &rc;
%mend;
%sysmac

proc contents data=sasdir.mailall;
run;
```