

New Data Step Functions in SAS® 6.12 - Part II

External File Functions

Caroline Bahler, BIOPOP, Radford, VA

Abstract

The Data step in SAS® 6.12 is the recipient of 50+ new functions. These new functions are broken into the following groupings data set functions, external file functions, library and catalog functions, special functions, variable functions, and statistical functions. The objective of this paper is to discuss the external file functions and to illustrate potential uses of the external file functions.

Introduction

The external file functions come originally from the SCL (Screen Control Language) realm and have been available to SCL programmers for years. These functions can be utilized within a macro or Data step and provide the programmer with an additional set of options for accessing information about an external file directory and information stored within flat files. Within a macro, a programmer will reference these functions by using the %sysfunc and/or %qsysfunc macros. Whereas, within a Data step the functions are called in a similar manner used within SCL code¹. Functionally, this group of functions can be divided into seven (7) usage groups - directory, file identifiers, file required, file information, file record identification, file record selection and retrieval and file modifiers.

Usage Groupings

Directory functions consist of the DOPEN, DOPTNUM, DOPTNAME, DINFO, DNUM, DREAD and DCLOSE functions (see table 1 for descriptions). The DOPEN function must be used first before the other functions can be utilized, since it "opens" or identifies the directory. These functions are useful in that they allow a programmer to determine what files are stored within a given directory. In addition, depending on the host system used other pieces of information about the directory can be accessed. The main purpose of "Closing" the directory is to de-assign an identifier number and does not have the same consequences as not closing a SAS data set.

File identifier functions consist of FILENAME and FILEREF (see table 1 for descriptions). The FILENAME function is equivalent to the Filename statement in that a fileref is assigned or de-assigned. However, one additional output of the

function is a return code to indicate correct/successful completion of the assignment. In addition, the options available to the Filename statement are also available as options within this function.

File required functions consist of FOPEN, MOPEN and FCLOSE functions (see table 1 for descriptions). A file must be opened before any other file function can be used. The role of the FOPEN function is to define an identifier to an external file and specify the type of access allowed to the file. In addition, the record format is specified within this function. The MOPEN function works identically to the FOPEN except that the directory identifier is used and the name of the file is specified. "Closing" in this case can refer to either the directory of a file or a specific external file depending upon whether FOPEN or MOPEN was used. The reason a file is closed is to de-assign an identifier number and does not have the same consequences as not closing a SAS data set.

File information functions consist of the FEXIST, FILEEXIST, FOPTNUM, FOPTNAME, FINFO and PATHNAME functions (see table 1 for descriptions). These functions determine whether a file exists and the properties associated with the external file.

File record identification functions consist of the FNOTE, FPOINT, FREWIND, FCOL, FPOS and DROPNOTE functions (see table 1 for descriptions). These functions determine what record will be copied into the File Data Buffer (FDB). The FDB functions in a similar fashion to Dataset Data Vector (DDV).

File record selection and retrieval functions consist of the FGET, FPUT, FREAD and FRLen functions (see table 1 for descriptions). These functions act in a similar fashion to an Input statement within a data step. The FREAD copies values from a record within an external file are into the FDB. The FGET function copies values from the FDB into a macro or Data step variable. The FPUT function copies macro or Data step variable into the FDB².

File modifiers functions consist of the FAPPEND, FWRITE and FDELETE functions (see table 1 for descriptions). The first two (2) functions change the information within an external file by either

appending information at the end of the file or
copying information into a file at a specific record

number and column within the record.

Table 1. Listing of the external file functions grouped by function usage.

Grouping	Function	Description
Directory	DOPEN	Opens directory to allow information retrieval about that directory.
	DOPTNUM	Determines how many pieces of host specific directory information are available for a directory. For example, using Windows 95 operating system only the physical directory specification is available.
	DOPTNAME	Determines the name of a specific directory information item. Name of the information item associated with the number.
	DINFO	Determines the value of a specific directory information item.
	DNUM	Determines the number of files within a directory.
	DREAD	Determines the filename associated with a file number. The combination of DNUM and DREAD allow the compilation of a directory file list.
	DCLOSE	Closes an directory was opened with an DOPEN function and any open members associated with that directory.
File Identifiers	FILENAME	Has the exact same function as the FILENAME statement, it assigns and de-assign a file reference identifier.
	FILEREF	Determines whether a file reference identifier defined within the FILENAME function is assigned.
File Required	FOPEN	Opens an external file to allow both reading and writing to that file depending on the options selected, after a file reference identifier has been assigned.
	MOPEN	Opens an external file to allow both reading and writing to that file by using the filename provided by the DREAD function. Has the same function as the FOPEN function.
	FCLOSE	Closes an open file.
File Information	FILEEXIST	Determines whether a file exists by using the full filename.
	FEXIST	Determines whether a file exists by the file reference identifier.
	FOPTNUM	Determines how many pieces of information are available for a file.
	FOPTNAME	Determines the name of a specific file information item.
	FINFO	Determines the value of a specific file information item.
	PATHNAME	Identifies the physical storage location of an external file.
File Record Identification	FNOTE	Determines the last record copied into the FDB.
	FPOINT	Determines which record will next be copied into the FDB.
	FREWIND	Moves the record pointer to the top of the file.
	FCOL	Determines what is the column number within the FDB.
	FPOS	Moves the column pointer within the FDB to a specific column position.
	DROPNOTE	Removes the marked placed by the NOTE function on an observation number.
File Record Selection and Retrieval	FGET	Copies the values of a record into the FDB.
	FPUT	Copies values into the FDB at the current column position from a data set variable, macro variable or entered value.
	FREAD	Copies the values of a file record into the FDB.
	FRLen	Determines the length of a record copied into the FDB.
File Modifiers	FAPPEND	Appends a record the end of a file using the values with the FDB.
	FWRITE	Writes a record to a file containing the values with the FDB.
	FDELETE	This function is equivalent to executing the system command to delete a file.

The main hallmark of all of these functions is that they are used in conjunction with each other to reach a particular goal. As stated previously, before any of

the other functions are used the external file or directory must first be opened, then functions from the other usage groups can be utilized as required.

Documentation for each of the functions in Table 1

can be found within the HELP area of SAS 6.12, including syntax specifications. To access this information a HELP user needs to click on the HELP icon within the toolbar and select one of the following areas from the main HELP menu- What's New and Base SAS. A listing of all of the new functions occurs within the What's New section under Base SAS - Functions. Or the information about the functions can be accessed within the complete listing of all Base SAS functions under Base SAS Documentation - Language Reference, SAS Functions³.

Utilization of functions

Typically, the following steps should occur within a data step or macro for data retrieval to occur.

Illustration of steps utilizing the %sysfunc functions within a macro:

```
options symbolgen mtrace mprint;
```

```
%macro test;
```

Step 1 - Assign a fileref to an external file.

```
%let fileref=demodata;
%let fn=%sysfunc(filename(fileref,
    'c:\sas\sasuser\demodata.txt'));
```

Step 2 - Determine whether the external file exists.

```
%if %sysfunc(fexist(&fileref)) %then %do;
```

Step 3 - Open the external file specifying the the record layout length and variable delimiter (similar to the lrecl= and delimiter= options within the infile statement).

```
%let fopened=%sysfunc(fopen(&fileref,i,100));
%let rc=%sysfunc(fsep(&fopened,%str(,)));
```

Step 4 - Copy the entire data record into the FDB.

```
%if %sysfunc(fread(&fopened)) = 0 %then %do;
```

Step 5 - Copy the contents of the FDB into macro variables.

```
%let rc=%sysfunc(fget(&fopened,name));
%let rc=%sysfunc(fget(&fopened,address));
%let rc=%sysfunc(fget(&fopened,city));
%let rc=%sysfunc(fget(&fopened,state));
%let rc=%sysfunc(fget(&fopened,zip));
%put &name &city &state &zip;
%end;
%end;
```

Step 6 -Close the external file.

```
%let rc=%sysfunc(fclose(&fopened));
```

```
%mend test;
%test;
```

Illustration of steps using a Data step.

```
data test;
```

Step 1 - Create the variables that the values from the FDB will be copied into.

```
length name address city $30 state $2 zip $5;
name = '';
address = '';
city = '';
state = '';
zip = '';
```

Step 2 - Assign a fileref to the external file.

```
rc=filename("fileref","c:\sas\sasuser\demodata.txt");
```

Step 3 - Determine whether the external file exists.

```
if fexist("fileref") then do;
```

Step 4 - Open the external file specifying the record layout length and variable delimiter.

```
fopened=fopen("fileref","i",100);
rc=fsep(fopened,',');
```

Step 5 - Copy the entire data record into the FDB.

```
if fread(fopened) = 0 then do;
```

Step 6 - Copy the contents of the FDB into variables.

```
rc=fget(fopened,name);
rc=fget(fopened,address);
rc=fget(fopened,city);
rc=fget(fopened,state);
rc=fget(fopened,zip);
end;
end;
```

Step 6 -Close the external file.

```
rc=fclose(fopened);
drop rc fopened;
run;
```

The above two examples differ in that within the Data step example the variables were first defined and then the values from the FDB were copied into them. In comparison, within the macro the variables

were create and the values from the FDB were copied into them at the same time. This is not always necessary, but can be considered to be good practice since the default variable length within a Data step is eight (8) for both character and numeric variables.

The macro below illustrates the use of the directory functions.

```
%macro dirtest;
%let fileref=demodata;
%let fn=%sysfunc(filename(fileref,'c:\sas\sasuser'));

%let dirid=%sysfunc(dopen(&fileref));

%let opts=%sysfunc(doptnum(&dirid));
%let optname=%sysfunc(doptname(&dirid,&opts));
%let dirinfo=%sysfunc(dinfo(&dirid,&optname));

%let filecnt=%sysfunc(dnum(&dirid));
%do i = 1 %to &filecnt;
  %let file=%sysfunc(dread(&dirid,&i));
  %put &file;
%end;
%put &opts &optname &dirinfo &filecnt;

%let rc=%sysfunc(dclose(&dirid));

%mend dirtest;
%dirtest;
```

In conclusion, these file functions put a set of valuable tools in the hands of SAS programmers.

References

1. SAS Institute Inc., SAS Screen Control Language: Reference, Version 6, First Edition, Cary, NC: SAS Institute Inc., 1990. 91-374 pp.
2. SAS Institute Inc., SAS Screen Control Language: Reference, Version 6, First Edition, Cary, NC: SAS Institute Inc., 1990. 47 - 50 & 91pp.
3. Bahler, C. (1997) New Data Step Functions in SAS 6.12 - Part I Data Set Functions. Proceedings of the Fifth SouthEast SAS Users Group Conference.

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

Caroline Bahler
GIV Inc.
7335 Lee Highway
Radford, VA 24141
540-633-3500
cbahler@giv.com