The FILEVAR Option: A Tool for Management of Large Numbers of Raw Data Files

Henry W. Buffum, Computer Sciences Corporation
Jeffrey S. Rosen, Computer Sciences Corporation
Richard W. Latimer, U.S. Environmental Protection Agency

ABSTRACT

When building a SAS ® database system, it is usually preferable to store all data in SAS datasets. However, in many scientific databases it is necessary to store some data in ascii data files. Files produced by laboratory instruments or monitoring devices may need to be kept online in their original raw format. And scientists may prefer to use their own FORTRAN programs or other software to access these raw files. Fortunately, it is not difficult to design a SAS database system that can manage raw ascii data files. A new feature of the INFILE statement, the FILEVAR option, is particularly useful in managing large numbers of raw data files. This paper describes a SAS-based environmental database system that allows easy access to large amounts of data stored in raw ascii format.

INTRODUCTION

The EPA's Environmental Monitoring and Assessment Program (EMAP) measures environmental quality parameters at over a hundred sampling stations in estuaries on the eastern seaboard of the United States each year. Most of the data from these stations are added to SAS datasets that make up the EMAP Estuaries database. These datasets all contain the key field STA_NAME (Station Name) and optionally, a key field VST_DATE (Visit Date). The data can be merged on these key fields in SAS datasteps or in SQL views.

Additional data from sampling stations are stored in raw ascii files. One such type of data file is the vertical profile of the water column for salinity, temperature, dissolved oxygen concentration, light transmission, fluorescence, photosynthetically active radiation and other parameters, which is obtained using the Seabird CTD SeaLogger. As this device is lowered through the water column it stores measurements from an array of instruments in an internal data logger. After the device is retrieved from the water, the data are downloaded to a DOS-based field computer system. Next they are transferred to the central VAX minicomputer where they are stored in individual data files.

These raw files are incorporated into the EMAP Estuaries database by storing their names in the SAS dataset WATRQUAL. This dataset also contains the key fields STA_NAME and VST_DATE, allowing it to be linked to other datasets in the database, utilizing all the data management power of the SAS system. The raw data files can be easily read into temporary SAS datasets when needed, making their data elements part of the larger database. In implementing this system, a new feature of the INFILE statement, the FILEVAR option, proved to be very helpful.

THE FILEVAR OPTION

The FILEVAR option is a new (with Version 6) feature of the INFILE statement. It is used in conjunction with the END option in the following manner:

infile <dummy> filevar=<var1> end=<var2>;

where <dummy> is an arbitrary, dummy file name that is never referenced elsewhere, <var1> is a variable that contains the names of files to be read by the infile statement, and <var2> is a logical variable that will be set "on" when the end of the file being read by the INFILE statement is reached. The variable <var1> must be created before the INFILE statement is executed. The variable <var2> is created by the INFILE statement and will be referenced in code that follows.

When using the FILEVAR option, the name of input file to be read in a data step does not have to be hard-coded into the INFILE statement. Instead, the file name is passed to it through the variable specified in the FILEVAR clause.

In our example, the water profile data file for selected sampling stations must be read to obtain summary statistics. These profile data files all consist of several columns of data in the same fixed format:

Columns	Variable	Description
1-11	seconds	Seconds in Water
12-22	meters	Water Depth (meters)
23-33	temp	Temperature (degrees C)
34-44	salinity	Salinity (ppt)
45-55	cond	Conductivity (s/m)
56-66	oxy	Dissolved Oxygen (mg/l)
67-77	ox_temp	Oxygen Temperature (Degrees C)
78-88	pH	Ph (units)
89-99	light	Light Transmission (%)
100-110	lightpar	Light PAR (micro-einsteins/m**2/s)
111-121	fluores	Flouresence (units)

Note that the raw files contain no information about the time or place that the data were collected. To obtain this information the data files must be related to the SAS database.

The dataset WATRQUAL contains the following variables (for this example the actual dataset structure has been simplified):

Variable	Format	Description
STA_NAME	$9.	Sampling Station Identifier
VST_DATE	YYMMDD.	Date of Sampling
CTDFILE	$30.	Raw Data File Name

The variable CTDFILE contains the full physical file name of the raw profile data file. This dataset can be merged on the key field STA_NAME with another dataset containing the exact locations of sampling stations in latitude and longitude.

The following SAS code accesses the dataset WATRQUAL, selects two sampling stations of interest, and passes the filenames to a datastep that reads the raw files data. The other variables from WATRQUAL are also passed to the final dataset, where they are combined with the data read from the raw files. Summary statistics are then produced.

Sample Code:

```
data step1;
set est.watrqual;
/* select sampling stations of interest*/
if sta_name = "VA91-123" or sta_name="VA91-456";

proc sort; by sta_name vst_date;
proc print;

data step2;
set step1;
infile dummy filevar=ctdfile end=eof;
do until(eof);  /* read all selected ctd files */
     input seconds     1-11
           meters      12-22
           temp        23-33
           salinity    34-44
           cond        45-55
           oxy         56-66
           ox_temp     67-77
           pH          78-88
           light       89-99
           lightpar    100-110
           fluores     111-121;
     output;
     end;

proc means;
by sta_name vst_date;
```

This program first creates the temporary dataset STEP1, with 4 observations (each sampling station selected has 2 observations in WATRQUAL). This first data step is where the analyst or applications programmer can select the cases she is interested in. All the tools of the SAS system could be used to subset WATRQUAL, and to merge it with other datasets in the database. The variables saved in this dataset will be passed to the next dataset STEP2.

In this example, the selection of cases from the WATRQUAL dataset is very simple. The PRINT procedure that follows the first data step produces the following:

STA_NAME	VST_DATE	CTDFILE
VA91-123	910706	DISK$DATA:[VA91.CTD]8473989.DAT
VA91-123	910810	DISK$DATA:[VA91.CTD]8848595.DAT
VA91-456	910708	DISK$DATA:[VA91.CTD]8885857.DAT
VA91-456	910910	DISK$DATA:[VA91.CTD]8848395.DAT

Note that the file names stored in the variable CTDFILE are system-dependent VMS file specifications. The syntax of these file names would of course be different on other platforms.

The next data step reads the four observations in STEP1. For each of these observations, the file name in the variable CTDFILE is passed to the INFILE statement through the FILEVAR option. The INPUT statement that actually reads the files is inside a DO loop that iterates until the end of the current file is reached. This loop is controlled by the variable named in the END option of the INFILE statement. When the loop stops executing, the next observation from STEP1 is read. At this point, a new value for CTDFILE is passed to the INFILE statement, so it will then point to the new file. For each of the four observations in STEP1, multiple observations are created by reading the raw data file named there. The resulting dataset STEP2 will have 400 observations if each of the four raw data files has 100 records.

The dataset STEP2 contains all the variables read from the raw data with the INPUT statement, plus the variables STA_NAME and VST_DATE from the dataset STEP1. (The variable CTDFILE is not included in the final dataset. The variable named in the FILEVAR option is never saved in the resulting dataset.) This final dataset can now be processed with any SAS procedure.

Some Limitations and Possibilities

This technique can be used to read raw data files with much more complex structures than those in the above examples. Most of the features of the INPUT statement, such as pointer controls and line-hold specifiers made up of @ symbols, are compatible with the FILEVAR option of the INFILE statement. However, the technique should NOT be used with INPUT statements that use specific line pointer controls, made up of # signs. The FILEVAR option should be used in conjunction with an END option in the INFILE statement, and not with the EOF option.

Using the type of logic described above, a piece of SAS code can be written and saved that will read whatever raw files are passed to it the FILEVAR variable. The second data step in the above example could be stored as a production routine and called with %include statements whenever needed. This code can be incorporated into SAS/AF® applications by placing it within a SUBMIT block, allowing an easy, online access to the raw data.

The FILEVAR option has many other useful applications. It can be used in combination with the X command (or your platform's command to invoke the operating system) to obtain a list of raw data files currently residing in a directory. The list of raw data files can then be read into a temporary SAS dataset, and passed to an INFILE statement with FILEVAR option. This is a quick and easy way to read all of the raw files in a directory with one SAS program (provided that all the raw files have the same record format). The FILEVAR option can also be used on a FILE statement, allowing a datastep to write to multiple output files.

CONCLUSION

The FILEVAR option of the INFILE statement allows very flexible data steps to be written to process raw data files. A data step can be coded that will read any file passed to it, or multiple files in sequence. Used in conjunction with a permanent SAS dataset containing the names of raw data files, this technique is quite powerful. All the data management tools of the SAS system-indexing, sorting, merging, SQL, etc., can be called upon to manipulate the SAS dataset. The raw data files named in it can then be read into SAS format, automatically combining their data elements with the data elements in the SAS database.

Contact Author:

Henry W. Buffum
27 Tarzwell Drive
Narragansett, RI 02882
(401) 782-3183