ABSTRACT

Applications with large SAS® data sets may benefit by using SAS Institute's SYSTEM 2000® Data Management Software. Indexes are available to speed queries, and selection criteria can be applied to subset the data. Redundant values can be eliminated. Sorting and visual display of data relationships are provided. This paper examines new options available in Version 5 of SAS/FSP® software's FSEDIT procedure on IBM® computers for SYSTEM 2000 data bases.

INTRODUCTION

The SAS System is used in many data base applications. These vary in complexity. The important variations from a data management perspective include:

- number and sizes of the files
- number of users and their access patterns
- system requirements for performance, security, and availability.

Often, the choice of an application tool dictates the choice of the data management tool it uses. The data base interfaces in the SAS System allow these two choices to be separate.

One of the data management tools that works with the SAS System is SAS Institute's own SYSTEM 2000 Data Management Software. Various interfaces to SYSTEM 2000 data bases are built directly into SAS software. This paper describes one of them: an enhancement to SAS/FSP software's FSEDIT procedure on IBM mainframes. All the capabilities of PROC FSEDIT are available for accessing SYSTEM 2000 data bases. Several new features have also been added.

DECIDING IF THIS INTERFACE IS FOR YOU

Using PROC FSEDIT with SYSTEM 2000 software will be helpful to you if you have a current FSEDIT application with performance problems, particularly in the areas of access speed or data storage. A SYSTEM 2000 data base should be considered for new FSEDIT applications if:

- several related files will commonly be used together
- one or more of the files is very large or contains redundant information
- the queries vary widely and often apply only to a subset of the data.

If one or more of these factors apply, then you may also want to consider the following:

- Is concurrent access, including update, routinely required?
- Is it important to be able to recover files quickly following system or hardware errors?
- Is data security an issue?

CREATING A SYSTEM 2000 DATA BASE WITH PROC FSEDIT

It is easy to try out an application by using PROC FSEDIT to create a SYSTEM 2000 data base. If you have ever created a new SAS data set using PROC FSEDIT, then you are already familiar with this process.

Example: The SALES data base has four files or record types:

- The ENTRY record (C0 for short) has general information for each sales region.
- The SALES PERSONNEL record (C100) has general information about each salesman in each region.
- The MONTHLY SALES record (C110) contains sales figures by month for each salesman in each region.
- The SEMINAR PLANNING record (C200) describes plans for sales seminars in each region.

One way to implement this application is to create four separate SAS data sets, replicating enough information in each data set to tie them together. In a SYSTEM 2000 data base, you group all the files into one data base and define their relationships. The SYSTEM 2000 software ties the files together without replicating information.

To define and create the data base, invoke PROC FSEDIT by entering:

```
proc fsedit newdb=s2k;
run;
```

A panel for defining records and variables is displayed as Screen 1. The panel, shown below, is very similar to the PROC FSEDIT panel for defining a SAS data set, but you will notice one difference immediately. An additional type, 'R', is used to identify different record types. PROC FSEDIT can process more than one type of record in the same session, according to the relationships you define with the Data Base Definition Panel.

Screen 1 Data Base Definition Panel
Each record contains unique information. Relationships can be defined by filling in a short-form record identifier under the "Owner" column on the panel.

You probably recognize these as hierarchical relationships. This means that you can draw the information in a multilevel tree, such as those used for organizational charts. Many applications, though not all, have this characteristic. PROC FSEDIT draws your tree for you, as shown in Screen 2. Enter the TREE command on the command line.

Screen 2 TREE Display for Sales Data Base

PROC FSEDIT can draw different relationships as you experiment with the definition panel, before actually mapping the definition into permanent files. When you are satisfied, enter the END command. Your data base is created. The FSEDIT procedure continues, and the new data base is available to you in this or future sessions. You can use PROC FSEDIT to manually enter data into the records. You can also use other built-in SAS procedures, such as the S2KLOAD procedure, to populate the data base.

ADVANTAGE OF HIERARCHICAL RELATIONSHIPS

Hierarchical relationships have been a common approach to data management for many years, mainly because so many applications lend themselves to this model. Hierarchies allow a one-to-many relationship, as well as the one-to-one relationship of relational systems. Using a one-to-many approach means that only one copy of shared information appears in the data base. The advantage is in performance. Such structures can be updated and retrieved very efficiently.

TAILORING YOUR FSEDIT SESSION

It is not often that you need to access an entire data base. It is more likely that you only want to examine a few fields and then only part of the total number of records. There are two ways to tailor your FSEDIT session

• by using a familiar tool, saving your FSEDIT panel
• by using a new tool, the FSEDIT selection panel.

SAVING SCREENS FOR SYSTEM 2000 DATA

You can use the panel configuration tools of the FSEDIT MODIFY command to tailor and save your application interface, just like a SAS data set. All SYSTEM 2000 information about that session is saved with the screen

```plaintext
proc fsedit database=s2k screen=lib.screen;
run;
```

One handy feature is the ability to use the same screen with both SAS data sets and SYSTEM 2000 data bases. In fact, if you decide to test a current SAS application with SYSTEM 2000 software, you can use all your existing screens as is.

SELECTING VARIABLES FOR AN FSEDIT SESSION

The DATABASE=S2K option in PROC FSEDIT, as shown above, connects you with the SYSTEM 2000 software. If you do not enter additional options, then the data base access panel and the selection panel are presented before the FSEDIT observation display screen. The data base access panel requests the data base name, your password, and whether you desire single-user or Multi-User™ (shared) access to the data base. The selection panel, shown in Screen 3, asks you to pick your variables for the session.

Screen 3 Data Base Selection Panel

The selection panel shown above reflects how the data base information appears after an "S" in the "S/D" column is placed by you. The SYSTEM 2000 indexing feature quickly identifies your subset. And sorting criteria to subset and order the records for display.

At the bottom part of the screen, you can also enter qualification and sorting criteria to subset and order the records for display. The SYSTEM 2000 indexing feature quickly identifies your subset. If you prefer, you can specify non-key processing for particular variables.
VIEWING OBSERVATIONS

Although multiple record types were selected in the example above, PROC FSEDIT shows the data on one screen. It interprets the record relationships into one observation per branch of the tree. The data shown reside in multiple records but are presented in Screen 4.

Screen 4 Observation Edit Screen

Remember that the TREE command can be entered on the command line as a reminder of the relationships. Another command, DATATREE, or DTREE for short, also draws a tree. This tree (shown in Screen 5) shows how the information in the current observation appears in the data base. Since it is a peek into the database, relationships, as well as neighboring records, are shown.

Screen 5 DTREE Display for Observation 1

All the standard FSEDIT commands, such as FIND and STRING/SEARCH, are supported. In addition, several new commands (WHERE, SORT, OUTPUT, and CONTENTS) are described later in the paper.

With SYSTEM 2000 software, the FORWARD and BACKWARD commands are used to skip observations with common data. By placing the cursor on a field instead of on the command line, you can jump across observations to skip details you are not interested in. For example, in the DTREE shown above, if you are on observation 1 with the cursor on the BARKER, JOHN field, the FORWARD command will skip to observation 3. Observation 3 is a different salesman in the same region. This can be a time-saver in data bases with a lot of low-level information.

CHANGING YOUR MIND INSIDE PROC FSEDIT

Suppose that, after viewing a particular subset of records, you decide that you are really interested in another area of the data base. You can easily switch to this new request without ever leaving the procedure. The WHERE command makes this possible, as shown in Screen 6.

Screen 6 Using the WHERE Command

When the WHERE clause is entered, the SYSTEM 2000 software uses indexes to arrive at a new set of observations quickly. You are repositioned at observation 1.

Another convenient command is the SORT command. Enter SORT on the FSEDIT command line, along with your variable names and the sort sequence. For example,

```
   command >> sort desc sales
```

sorts the records in descending order by sales amount.

Neither the WHERE command nor the SORT command require you to use variables appearing on the screen; you can use any variable in the data base.

UPDATING THE SYSTEM 2000 DATA BASE

Although it looks like you are viewing a SAS data set, in fact you are directly connected to the SYSTEM 2000 data base. This means that you can make changes to the actual data base by keystroking them in on the panel. SYSTEM 2000 software handles concurrent update by multiple users, recovery of the data base in case the system fails during your update, and password-monitoring for access to secure data.

If you want to add new records to the data base, use the standard FSEDIT commands ADD or DUP. The procedure knows when to issue multiple update commands because your screen shows multiple records. With ADD, you insert whole new trees into the data base. With DUP, you either create a new copy of a whole tree or you append new branches to the tree you were looking at. New records do not always go at the end of the data base.
SYSTEM 2000 software takes care of finding an available location, thus conserving disk space.

Deleting data works the same way. The FSEDIT DEL command sets all the fields on the panel to missing. In addition, SYSTEM 2000 software examines the database to see if actual record locations are freed up as a result. This makes new locations available for additions and reduces the need to recover space by other means. The algorithm removes records when there are no other related records depending on it.

At any time, you can use the TREE and DTREE commands to observe changes to the database as a result of your updates. For example, a DTREE command before a DUP command shows a different picture than the one immediately after.

If you are accessing the database with SYSTEM 2000 Multi-User software, your updates are visible to other users, and theirs to you. This applies to value modifications and to structural changes, that is, record additions and deletions. You are also warned when someone else's updates have caused your current observation to no longer match your WHERE clause.

With the Multi-User software, the procedure obtains a database lock on the records you are looking at. If another user positions to the same record(s), he or she is informed that you have a lock on them. He or she can view them but cannot change them until you move to another observation. This coordination extends not only to FSEDIT users, but to all users in the SYSTEM 2000 database.

TAKING A SNAPSHOT OF THE FSEDIT SESSION

The new FSEDIT command, OUTPUT, is particularly useful when you are changing WHERE clauses frequently within the same session. This command writes to a SAS data set all the observations that meet your current qualification and sorting criteria. Having saved your results for later use in another SAS procedure, you can now move to a new WHERE clause. You can issue the OUTPUT command as many times as you like during the same session. If you need a reminder of which variables you have selected or what your current criteria are, issue the CONTENTS command on the FSEDIT command line. You will see a protected version of the selection panel described earlier in this paper.

CONCLUSION

PROC FSEDIT with SYSTEM 2000 software represents a new level of integration between the SAS System and SYSTEM 2000 software. It offers a more powerful data management engine to the existing SAS application and a full-screen query and update environment to the existing SYSTEM 2000 application. Many of the subtleties of dealing with hierarchies are handled internally by the software. Some applications are handled better by SAS data sets, others by SYSTEM 2000 data bases. With PROC FSEDIT as a common interface, it is easier to concentrate on the application instead of the underlying data management system.