Using a SAS/AF® Front End to Simplify Drill Downs in SAS/EIS®

Beth Bednarek, CNA Insurance, Chicago, IL.

Abstract

When a variety of drill paths are required for a SAS/EIS® application, consider building and registering a dataset which uses one standardized array for the drill variables. A SAS/AF® front end can then be used to select your actual drill variables and populate the array.

Coding Steps

1) Build a SAS/AF® Frame or Program entry to allow the user to select the desired drill variables and the order in which they want to drill.

2) In a Data step define 2 arrays, one for the drill variables and one for the labels associated with the drill variables. The format of the drill array should be able to hold any of the drill variables that may be selected.

   ```
   array drill(8)$ 14;
   array label(8)$ 40;
   ```

3) Using subscripts and the information from the SAS/AF® screen, populate the drill variable and label arrays, making (1) the lowest level in the drill. Be sure to save the highest subscript used.

   ```
   COLLOC:
   sub = 1;
   drill(sub) = claimno; label(sub) = "Claim Number";
   drill(sub) = &coffield; link labelcol;
   drillmax = SUB;
   ```

4) Loop through the label array and put each one out as a macro variable. These variables will be used in LABEL statements in a later Data step. Put the highest subscript out as a macro variable

   ```
   call symput('lbI1', label1);  
   call symput('lbI8', label8);  
   call symput('maxdrill', drillmax);
   ```

5) The final dataset should contain all the analysis variables and the entire drill variable array, even if some fields are blank. When this dataset is registered set the drill path to go from the highest element in the drill array down to the lowest. i.e. DRILL8 to DRILL1. Register the dataset with generic labels for all the drill fields since most of the drill objects use the labels at registration time for display purposes. Only the PIE chart uses the labels from the dataset at execution time.

   ```
   proc datasets library = eislib;
   modify drillfil;
   
   proc data datasets library = eislib;
   modify drillfil;
   ```

6) As mentioned above, there may be blank drill variables at the top of the drill array. To avoid blank drill screens at the start of the drill, create multiple iterations of each drill object with different starting points. Use the macro variable with the highest subscript to determine which object to call.

   ```
   maxdrill = symgetn('maxdrill');
   
   if maxdrill = 5 then do;
   call execcmd('run EisLib appl = eislib.abc.graph5.chart'); end;
   ```

Summary

As they say, 'The proof is in the pudding'. In this case the "pudding" is an application that has 134 distinct drill down paths. All the drills are accomplished with one registered dataset and seven iterations of each drill object. By pairing the power of SAS/EIS® drill objects with the flexibility of a SAS/AF® screen, you can build an easy to maintain drill down application that any user will love.