Carolyn Cunnison Con't

History: After earning a degree in education from Dalhousie University in Halifax and a B.A. from the University of New Brunswick in Fredericton, Carolyn taught at a secondary school in Saskatchewan. Shortly thereafter, she launched her IT career as a COBOL programmer in MVS and VM environments. Carolyn now has more than 25 years of IT experience, with an impressive resume that includes Travelers Insurance, Royal Insurance, Pansophic Systems and Computer Associates.

Family: Carolyn is a native of northern New Brunswick. Her family lives in Bathurst (home of the Acadie-Bathurst Titans hockey team) and Moncton (a hop and a skip away from beautiful Parlee Beach).

Sports/Hobbies: Cross-country skiing, bicycling and yoga (the traditional variety rather than Power Yoga).

Ideal weekend would be: A Thanksgiving weekend with my family at their camp in northern New Brunswick. We cook the whole meal on the wood stove (except for the turkey, which we cook ahead of time). Then we walk off the big dinner by climbing Mount Carlton. The mountain puts on a dazzling display of gold and orange leaves at that time of year.

Favourite Foods: Meringues. You have to slow cook them in the oven overnight until they reach just the right degree of crispiness on the outside but softness on the inside. Then you glue two of them together with real whipped cream. Heavenly!

If I could be anything at all (besides a SAS programmer), I would be … a gardener. I have a rose garden, a vegetable garden, an herb garden, numerous flower beds, two rain barrels and a compost pile.

When I'm not programming in SAS, I like to … bicycle. I am a member of the Dorval Bicycle Club, which does short trips in the Montreal region. In addition, I have taken bicycle vacations in France, Wales, New England, Nova Scotia and the Eastern Townships.

One thing every SAS programmer should know …

How to create a cumulative value using PROC REPORT:

PROC REPORT is a flexible and powerful procedure available in Base SAS. It has many advantages over PROC PRINT. One of these advantages is the ability to create a derived variable. For example, the code below will create a derived variable that calculates the difference between Actual and Predicted Sales.

Cut out the code below and run it in your SAS session to see the results. For the input data set, I have used the first 12 records from the PRDSALES data set, which is located in your SASHELP library.

```
proc report data=sashelp.prdsale(obs=12) nowd;
   column month actual predict difference;
   define month / display width=5 center ;
   define actual / format=dollar10.0;
```

```
   define predict / format=dollar10.0;
   define difference/
          'Difference between Actual and Predicted Sales'
          format= dollar10.0
          computed;
   Compute difference;
      difference = actual.sum - predict.sum;
   endcomp;
run;
```

Now let's carry this example one step further. Let's create a derived variable that is ***cumulative***. For example, suppose you want to modify the above code to create the cumulative value of Actual Sales. Your first instinct might be to write the following code:

```
proc report data=sashelp.prdsale(obs=12) nowd;
   column month actual cumactual;
   define month / display width=5 center ;
   define actual / format=dollar10.0;
   define cumactual/
          'Cumulative Actual Sales'
          computed
          format=dollar10.0;

   Compute cumactual;
      cumactual=sum(cumactual,actual.sum);
   endcomp;
run;
```

However, if you examine the results, you'll notice that the Cumulative Actual Sales column will contain identical values to the Actual Sales column. In other words, the value is not accumulated. This is because any variable defined on the COLUMN statement is reinitialized for each record.

If, however, you calculate a variable that is ***not*** listed on the COLUMN statement, that variable is not reinitialized for each observation. This is the case for the variable I have named X below. X will be accumulated, and I can move it to the variable CUMACTUAL in order to print it. Et voilà!

```
proc report data=sashelp.prdsale(obs=12) nowd;
   column month actual cumactual;
   define month / display width=5 center ;
   define actual / format=dollar10.0;
   define cumactual/
         'Cumulative Actual Sales'
          format=dollar10.0
          computed;
   Compute cumactual;
      x=sum(x,actual.sum);
      cumactual = x ;
   endcomp;
run;
```