

Using %WINDOW to Obtain User Criteria for Reports

Michael A. Mace, SPS Software Services Inc., Canton, Ohio

ABSTRACT

Using %WINDOW provides an elegant means by which to gather user criteria for customized or ad hoc reports. Several examples will be used to demonstrate gathering single or multiple choices per criterion. In addition, the use of %WINDOW will be compared to using %PUT and %INPUT.

INTRODUCTION

The SAS® system provides a variety of information reporting capabilities which can be used for canned, hard-coded reports that are produced on a regular basis and also reports with basic layouts whose specific contents change based on some criteria input by the user. This paper examines several techniques to gather the criteria for customized or ad hoc reports.

You could create a panel using SAS/AF®, if your site is licensed for such, and you do not mind adding another level of complexity to the application. This technique does provide a great deal of flexibility and macro programming options. However I prefer to keep things as simple as possible, for the client and me as the developer.

My first choice is usually %PUT, %INPUT, along with &SYSBUFFR, tools of the macro facility which comes with the base SAS system. These provide a means to step the user through the criteria-gathering process, expecting typed responses. That can be a concern; after all, computers are persnickety things and will produce output commensurate with the input given to them (GIGO).

%WINDOW provides a very elegant interface to the user along with all of the flexibility and programming power your application should need. In fact, my use of %WINDOW came about because a client, with a very simple application, required the ability to enter multiple values for a given criterion and did not want the bother of typing precisely and I did not want to create the SAS/AF® panels that would have been required.

Let us examine the use of %PUT - %INPUT and %WINDOW. Please refer to *SAS® Guide to Macro Processing, Version 6, Second Edition* for a thorough discussion about the macro facility, macro variables, %PUT, %INPUT, &SYSBUFFR, and %WINDOW.

%PUT and %INPUT

This technique steps the user through the criteria-gathering process. %PUT is used to display the prompt; %INPUT is used to capture the response in a macro variable. An accessory to %INPUT is the automatic macro variable, &SYSBUFFR, which accepts any response to a %INPUT prompt that is not matched with a specific variable. Here is a short example:

```
%PUT Please enter the variable for which you want a frequency
table ;
%INPUT freqvar ;
```

```
PROC FREQ DATA= fileref.dsname ;
  TABLES &freqvar / options ;
RUN ;
```

When working with dates, I make the prompt very specific, remembering that macro variable values are text. Therefore I request the date values in DATE9. format so that I do not need to make any data conversions. Also notice that I have requested a four-character year to make sure everyone remembers that the turn of the century is upon us.

```
%PUT Please enter the BEGINNING date as DDMONYYYYY ;
%PUT i.e. 06APR2007 ;
%INPUT begndate ;
```

The response could be used thusly:

```
SET fileref.dsn(WHERE=("&begndate"D<=&datevar<="&enddate"D));
```

Multiple values can be accepted with %INPUT in one macro variable; however the format of the response is critical. Here is an example within a macro.

```
%MACRO macronam ;
```

```
CLEAR ;
%PUT Please enter the first three characters of desired
homebase(s) ;
%PUT in single quotes, - separate multiple homebases with
commas ;
%PUT ex.: '041','042','037' ;
%INPUT hb ;
```

```
DATA dsname ;
SET fileref.dsname ;
```

```
%IF %LENGTH(&hb) GT 0
%THEN %DO ;
      IF SUBSTR(homebase,1,3) IN(&hb) ;
%END ;
```

```
RUN ;
```

```
%MEND macronam ;
%macronam ;
```

In some circumstances, &SYSBUFFR, which accepts whatever the user enters can be used to gather multiple values. This allows a great deal of flexibility, independence, and yes, uncertainty; however it is crucial for our purposes. Whether you use %INPUT with multiple macro variables or &SYSBUFFR, in a particular situation, will be determined by how many different pieces of information are you expecting from the user and how is the response going to be used.

Several examples of how I have used &SYSBUFFR are:

```
%PUT Please list ALL of the variables you want on the report ;
%PUT in the desired order ;
%PUT THEN press <ENTER> ;
%INPUT ;
%LET var = &SYSBUFFR ;
```

```
%PUT %STR( );
%PUT If you want to specify an additional subsetting criterion, ;
%PUT please enter it as: variable comparison_operator value ;
%PUT OR: FUNCTION(variable,parameters) comparison_operator
value ;
%INPUT ;
%LET criteria = &SYSBUFR ;
TITLE4 "%UPCASE(&criteria)" ;
```

```
/* This code is necessary to prepend the word 'IF' and
append the ':' to the macro variable &criteria */
```

```
%MACRO criteria ;
%IF %LENGTH(&criteria) = 0
%THEN ;
%ELSE %STR(IF &criteria, ) ;
%MEND criteria ;
```

For a complete ad hoc program using %PUT, %INPUT, and &SYSBUFR, please see my paper, "Ad Hoc Reports for Users", also included in these NESUG '97 proceedings.

One assumption I have made in my use of %PUT and %INPUT is that the client will enter exactly what is requested as requested; if the criteria are not input correctly, an empty report will be produced. I do not do any error checking when it comes to the client's inputs; that could be incorporated if deemed necessary.

I should mention that when I work with my clients in creating an application, I constantly emphasize that they are the owners and users of the application and need to know the basics of what things are, how things work, and what can be reasonably expected. Just as they educate me about their needs, I try to educate them about the tools of the SAS® system. This mutual understanding helps not only in the initial development but also during any trouble-shooting or future modifications.

To conclude our discussion of %PUT and %INPUT, please notice that the only control over the user interface is the use of CLEAR; and %STR(). While functional for this 'one question at a time' method, it is not the most appealing. Also, let me re-emphasize that when using %PUT and %INPUT, generally we are asking for one criterion or value at a time and sometimes we expect the response to be in a specific format. The following piece of code, while useful, will serve as a means of comparison as we examine the use of %WINDOW to obtain user input.

```
%PUT %STR( );
%PUT Please enter a COMPANY code ('041'|'051'), if desired: ;
%PUT <<< Use single quotes around each COMPANY >>> ;
%PUT <<< Separate multiple COMPANYS with commas >>> ;
%PUT Simply press <ENTER> for all companies ;
%INPUT company ;
```

```
%PUT %STR( );
%PUT Please enter the REGION for which you want labels: ;
%PUT <<< Use single quotes around each REGION >>> ;
%PUT <<< Separate multiple REGIONs with commas >>> ;
%PUT Simply press <ENTER> for all regions ;
%PUT %STR( 'Allegheny' ) ;
%PUT %STR( 'Blue_Mountain-Reading' ) ;
%PUT %STR( 'Blue_Mountain-Lebanon' ) ;
%PUT %STR( 'Colonial' ) ;
%PUT %STR( 'Delaware' ) ;
%PUT %STR( 'Juniata' ) ;
%PUT %STR( 'Lake_Erie' ) ;
%PUT %STR( 'Laurel' ) ;
%PUT %STR( 'Susquehanna' ) ;
%INPUT region ;
%IF &region NE &null
%THEN %LET region = %UPCASE(&region) ;
```

```
CLEAR ;
%PUT '00'- <National Account>
'99'- Industrial ;
%PUT '01'- Geothermal Heat Pump Dealers
'23'- Energy Management Committees ;
%PUT '02'- Meter & Service Handbook
'24'- Realtors ;
%PUT '03'- Geothermal Manufacturers/Dist
'26'- Electricians ;
%PUT '04'- Elec Heat Distributors
'27'- Builder/Developer ;
%PUT '05'- Heat Pump Distributors
'28'- Solar Dealer ;
%PUT '06'- Heat Pump Dealers
'29'- Residential Heat Dealers ;
%PUT '07'- R T Max '30'- PEA - Pa ;
%PUT '08'- ComfortHome Builders
'31'- PEA - Out of state ;
%PUT '09'- Good Cents Builders
'40'- Geothermal Drillers ;
%PUT '10'- Appliance Dealers
'41'- Geothermal Trenchers ;
%PUT '12'- Insulation Contractors
'70'- Medical Organizations ;
%PUT '14'- Plumbers
'90'- 1990 Good Cents Certified Homes;
%PUT '15'- General Contractors
'91'- 1991 Good Cents Certified Homes;
%PUT '16'- Communications Managers
'92'- 1992 Good Cents Certified Homes;
%PUT '17'- Architects
'93'- 1993 GC/CH Certified Homes ;
%PUT '18'- Engineers
'94'- 1994 GC/CH Certified Homes ;
%PUT '19'- Government/Municipal
'95'- 1995 GC/CH Certified Homes ;
%PUT '22'- School Districts
'96'- 1996 GC/CH Certified Homes ;
%PUT <<< REMEMBER NUMBERS FOR INPUT ON NEXT
PAGE >>> ;
CLEAR PAUSE ;
%PUT Please enter the number(s) of the TYPE(s) for which you
want labels: ;
%PUT <<< Use single quotes around each TYPE >>> ;
%PUT <<< Separate multiple TYPEs with commas >>> ;
%PUT Simply press <ENTER> for all types ;
%INPUT types ;
```

Notice in this last section that the entire screen has been filled with options and the user must remember their choices, in order to type them on the next screen, along with all the single quotes and commas; remember, no spaces. It works, but UGH ! Let's look at what %WINDOW can do.

%WINDOW

The %WINDOW statement defines customized windows, which are displayed with the %DISPLAY statement, both of which are part of the macro language included in the base SAS system. The %WINDOW statement combines the functionality of the %PUT statement, displaying text, and the %INPUT statement, accepting input into macro variables. What it adds is the ability to format the display and open the entire user interface at one time. Again, please refer to *SAS Guide to Macro Processing, Version 6, Second Edition* for a thorough explanation of these two statements and all of their options. My focus here is how to use these statements to gather user criteria for reports.

The %WINDOW statement uses the following pointer control characters:

- # - specifies the row within the window
- / - moves the pointer to column 1 of the next row
- @ - specifies the column within the window
- + - moves the pointer the specified number of positions

Fields can contain text, enclosed in quotes, or a macro-variable.

The macro-variables can already exist or be created within the field, by specifying a name and field-length. Several options, such as rows, columns, color, attributes, protection and required, are available to further enhance and control the display and functionality of the window. Please notice that the %WINDOW statement completely defines the display, with all text, fields, and formatting in one statement, i.e. the semi-colon is all the way at the end.

One caveat when using %WINDOW is that the displayed window disappears when the [ENTER] key is pressed. Therefore the user must be informed to use the [TAB] key or cursor keys to maneuver between fields. Fortunately, the %WINDOW statement creates an automatic macro variable, &SYSMSG, whose value can be set with a %LET statement. Here is an example, which I have enclosed in a macro in order that I can perform validation of the user's input.

```
@MACRO macronam ;
%LET null = ;
```

```
%WINDOW wndoname
#2 @20 "Company Name: Title of screen"
#5 @20 "Please enter the requested data, THEN press
<ENTER>:"
#8 @15 "Company (21,41,51): "
    @40 company 2 ATTR= UNDERLINE
#10 @15 "Customer Premise ID: "
    @40 premise 7 ATTR= UNDERLINE
#12 @15 "Billing date (YYYYMM): "
    @40 yyyyymm 6 ATTR= UNDERLINE
;
%lblstr: ;
```

```
%LET %SYSMSG= Press [ENTER] only after filling in ALL fields;
%DISPLAY wndoname ;
CLEAR;
%IF (&company NE 21 AND &company NE 41 AND &company NE
51)
OR &company EQ &>null
OR premise EQ &>null
OR &yyyyymm EQ &>null OR %LENGTH(&yyyyymm) LT 6
%THEN %GOTO lblstr ;
```

```
%MEND macronam ;
%macronam ;
```

Great!; but that is still just asking for one value per criterion and the user is still typing the responses. What about when the user wants to select several values and doesn't want to spell out anything? I am glad you asked.

In this example, I do not use pointer controls before the first text field to demonstrate that the default positioning for the first field is row 1, column 1. Also, while the %WINDOW and %DISPLAY statements do not need to be within a macro, the %IF and %THEN and macro label statements do.

```
%WINDOW critria1
"<TAB> to choices, use any NON-blank character to choose,
press [ENTER] when ready"
#3 @15 "Please choose which wine(s) you like:"
#5 @20 vino1 1 ATTR= UNDERLINE @22 "Burgundy"
#6 @20 vino2 1 ATTR= UNDERLINE @22 "Rose"
#7 @20 vino3 1 ATTR= UNDERLINE @22 "Zinfandel"
#8 @20 vino4 1 ATTR= UNDERLINE @22 "Chardonnay"
;
```

```
%MACRO macronam ;
```

```
%lblstr: ;
%DISPLAY critria1 BLANK ;
```

```
%LET null = ;
```

```
%IF &vino1 NE &>null %THEN %LET vino1 = BURGUNDY ;
%IF &vino2 NE &>null %THEN %LET vino2 = ROSE ;
%IF &vino3 NE &>null %THEN %LET vino3 = ZINFANDEL ;
%IF &vino4 NE &>null %THEN %LET vino4 = CHARDONNAY ;
%LET wines = &vino1 &vino2 &vino3 &vino4 ;
```

```
%PUT "You have chosen the following wines: &wines" ;
%PUT "Is this correct (Y/N) ?" ;
%INPUT ansr ;
%IF %UPCASE(&ansr) EQ N
%THEN %GOTO %lblstr ;
```

```
/* &wines can now be used as the specified criteria */
```

```
%MEND macronam ;
%macronam ;
```

Now the user simply tabs to the underline in front of his/her choice(s) and types any character. The %IF - %THEN statements determine what value(s) were chosen and there is no concern about misspelling or forgetting a quote or comma. I have provided feedback to the user regarding their selections and a means to revise them. The field macro variable names have not been reset to null, therefore the previous selection(s) will be displayed; this of course, could be modified if appropriate.

Since we are working with macro variables, with some creativity and a bit more work, a dynamic window could be created. The displayed text and underlying values could come from a data set; the positioning of the fields in the window could be determined based on the number of values. SAS/AF® software provides several means to accomplish this dynamic interface. Please refer to *SAS/AF® Software: Usage and Reference, Version 6, First Edition*. Remember though, that the present purpose is simply to gather criteria to produce reports, and everything presented here is available in the base SAS system.

Finally, the following code uses the %WINDOW and %DISPLAY statements to supersede what was presented earlier using %PUT and %INPUT. Notice that two window definitions are used.

```
%WINDOW critria1
#1 @15 "Marketing: Mailing Labels Criteria Page 1"
#5 @20 "Company:"
#5 @30 co041 1 ATTR= UNDERLINE @32 "041-Penelec"
#5 @50 co051 1 ATTR= UNDERLINE @52 "051-Met-Ed"
#10 @20 "Region:"
#10 @30 regn1 1 ATTR= UNDERLINE @32 "Allegheny"
#11 @30 regn2 1 ATTR= UNDERLINE
    @32 "Blue_Mountain-Reading"
#12 @30 regn3 1 ATTR= UNDERLINE
    @32 "Blue_Mountain-Lebanon"
#13 @30 regn4 1 ATTR= UNDERLINE @32 "Colonial"
#14 @30 regn5 1 ATTR= UNDERLINE @32 "Delaware"
#15 @30 regn6 1 ATTR= UNDERLINE @32 "Juniata"
#16 @30 regn7 1 ATTR= UNDERLINE @32 "Lake_Erie"
#17 @30 regn8 1 ATTR= UNDERLINE @32 "Laurel"
#18 @30 regn9 1 ATTR= UNDERLINE @32 "Susquehanna"
;
```

```
%WINDOW critria2
#1 @15 "Marketing: Mailing Labels Criteria - Types    Page 2"
#2 @1 typ00 1 ATTR= UNDERLINE @3 "00-National Account"
#3 @1 typ01 1 ATTR= UNDERLINE
  @3 "01-Geothermal Heat Pump Dealers"
#4 @1 typ02 1 ATTR= UNDERLINE
  @3 "02-Meter & Service Handbook"
#5 @1 typ03 1 ATTR= UNDERLINE
  @3 "03-Geothermal Manufacturers/Dist"
#6 @1 typ04 1 ATTR= UNDERLINE
  @3 "04-Elec Heat Distributors"
#7 @1 typ05 1 ATTR= UNDERLINE
  @3 "05-Heat Pump Distributors"
#8 @1 typ06 1 ATTR= UNDERLINE
  @3 "06-Heat Pump Dealers"
#9 @1 typ07 1 ATTR= UNDERLINE @3 "07-R T Max"
#10 @1 typ08 1 ATTR= UNDERLINE
  @3 "08-ComfortHome Builders"
#11 @1 typ09 1 ATTR= UNDERLINE
  @3 "09-Good Cents Builders"
#12 @1 typ10 1 ATTR= UNDERLINE
  @3 "10-Appliance Dealers"
#13 @1 typ12 1 ATTR= UNDERLINE
  @3 "12-Insulation Contractors"
#14 @1 typ14 1 ATTR= UNDERLINE @3 "14-Plumbers"
#15 @1 typ15 1 ATTR= UNDERLINE
  @3 "15-General Contractors"
#16 @1 typ16 1 ATTR= UNDERLINE
  @3 "16-Communications Managers"
#17 @1 typ17 1 ATTR= UNDERLINE @3 "17-Architects"
#18 @1 typ18 1 ATTR= UNDERLINE @3 "18-Engineers"
#19 @1 typ19 1 ATTR= UNDERLINE
  @3 "19-Government/Municipal"
#20 @1 typ22 1 ATTR= UNDERLINE
  @3 "22-School Districts"
#2 @41 typ99 1 ATTR= UNDERLINE @43 "99-Industrial"
#3 @41 typ23 1 ATTR= UNDERLINE
  @43 "23-Energy Management Committees"
#4 @41 typ24 1 ATTR= UNDERLINE @43 "24-Realtors"
#5 @41 typ26 1 ATTR= UNDERLINE @43 "26-Electricians"
#6 @41 typ27 1 ATTR= UNDERLINE
  @43 "27-Builder/Developer"
#7 @41 typ28 1 ATTR= UNDERLINE @43 "28-Solar Dealer"
#8 @41 typ29 1 ATTR= UNDERLINE
  @43 "29-Residential Heat Dealers"
#9 @41 typ30 1 ATTR= UNDERLINE @43 "30-PEA - Pa"
#10 @41 typ31 1 ATTR= UNDERLINE
  @43 "31-PEA - Out of state"
#11 @41 typ40 1 ATTR= UNDERLINE
  @43 "40-Geothermal Drillers"
#12 @41 typ41 1 ATTR= UNDERLINE
  @43 "41-Geothermal Trenchers"
#13 @41 typ70 1 ATTR= UNDERLINE
  @43 "70-Medical Organizations"
#14 @41 typ90 1 ATTR= UNDERLINE
  @43 "90-1990 Good Cents Certified Homes"
#15 @41 typ91 1 ATTR= UNDERLINE
  @43 "91-1991 Good Cents Certified Homes"
#16 @41 typ92 1 ATTR= UNDERLINE
  @43 "92-1992 Good Cents Certified Homes"
#17 @41 typ93 1 ATTR= UNDERLINE
  @43 "93-1993 GC/CH Certified Homes"
#18 @41 typ94 1 ATTR= UNDERLINE
  @43 "94-1994 GC/CH Certified Homes"
#19 @41 typ95 1 ATTR= UNDERLINE
  @43 "95-1995 GC/CH Certified Homes"
#20 @41 typ96 1 ATTR= UNDERLINE
  @43 "96-1996 GC/CH Certified Homes"
;

%LET SYMSG = <TAB> to choices, use any NON-blank character
to choose;
%DISPLAY critria1 ;
```

```
%LET SYMSG = <TAB> to choices, use any NON-blank character
to choose;
%DISPLAY critria2 BLANK ;

CLEAR ;
%GLOBAL company region types ;
%MACRO queswndo ;
%LET null = ;
%IF &co041 NE &>null %THEN %LET co041 = 041 ;
%IF &co051 NE &>null %THEN %LET co051 = 051 ;
%LET company = &co041 &co051 ;

%IF &regn1 NE &>null %THEN %LET regn1 = ALLEGHENY ;
%IF &regn2 NE &>null
%THEN %LET regn2 = BLUE_MOUNTAIN-READING ;
%IF &regn3 NE &>null
%THEN %LET regn3 = BLUE_MOUNTAIN-LEBANON ;
%IF &regn4 NE &>null %THEN %LET regn4 = COLONIAL ;
%IF &regn5 NE &>null %THEN %LET regn5 = DELAWARE ;
%IF &regn6 NE &>null %THEN %LET regn6 = JUNIATA ;
%IF &regn7 NE &>null %THEN %LET regn7 = LAKE_ERIE ;
%IF &regn8 NE &>null %THEN %LET regn8 = LAUREL ;
%IF &regn9 NE &>null %THEN %LET regn9 = SUSQUEHANNA ;
%LET region = &regn1 &regn2 &regn3 &regn4 &regn5 &regn6
&regn7 &regn8 &regn9 ;

%IF &typ00 NE &>null %THEN %LET typ00 = 00 ;
%IF &typ99 NE &>null %THEN %LET typ99 = 99 ;
%IF &typ01 NE &>null %THEN %LET typ01 = 01 ;
%IF &typ02 NE &>null %THEN %LET typ02 = 02 ;
%IF &typ03 NE &>null %THEN %LET typ03 = 03 ;
%IF &typ04 NE &>null %THEN %LET typ04 = 04 ;
%IF &typ05 NE &>null %THEN %LET typ05 = 05 ;
%IF &typ06 NE &>null %THEN %LET typ06 = 06 ;
%IF &typ07 NE &>null %THEN %LET typ07 = 07 ;
%IF &typ08 NE &>null %THEN %LET typ08 = 08 ;
%IF &typ09 NE &>null %THEN %LET typ09 = 09 ;
%IF &typ10 NE &>null %THEN %LET typ10 = 10 ;
%IF &typ12 NE &>null %THEN %LET typ12 = 12 ;
%IF &typ14 NE &>null %THEN %LET typ14 = 14 ;
%IF &typ15 NE &>null %THEN %LET typ15 = 15 ;
%IF &typ16 NE &>null %THEN %LET typ16 = 16 ;
%IF &typ17 NE &>null %THEN %LET typ17 = 17 ;
%IF &typ18 NE &>null %THEN %LET typ18 = 18 ;
%IF &typ19 NE &>null %THEN %LET typ19 = 19 ;
%IF &typ22 NE &>null %THEN %LET typ22 = 22 ;
%IF &typ24 NE &>null %THEN %LET typ24 = 24 ;
%IF &typ26 NE &>null %THEN %LET typ26 = 26 ;
%IF &typ27 NE &>null %THEN %LET typ27 = 27 ;
%IF &typ28 NE &>null %THEN %LET typ28 = 28 ;
%IF &typ29 NE &>null %THEN %LET typ29 = 29 ;
%IF &typ30 NE &>null %THEN %LET typ30 = 30 ;
%IF &typ31 NE &>null %THEN %LET typ31 = 31 ;
%IF &typ40 NE &>null %THEN %LET typ40 = 40 ;
%IF &typ41 NE &>null %THEN %LET typ41 = 41 ;
%IF &typ70 NE &>null %THEN %LET typ70 = 70 ;
%IF &typ90 NE &>null %THEN %LET typ90 = 90 ;
%IF &typ91 NE &>null %THEN %LET typ91 = 91 ;
%IF &typ92 NE &>null %THEN %LET typ92 = 92 ;
%IF &typ93 NE &>null %THEN %LET typ93 = 93 ;
%IF &typ94 NE &>null %THEN %LET typ94 = 94 ;
%IF &typ95 NE &>null %THEN %LET typ95 = 95 ;
%IF &typ96 NE &>null %THEN %LET typ96 = 96 ;
%LET types = &typ00 &typ99 &typ01 &typ02 &typ03 &typ04 &typ05
&typ06 ;
%LET types = &types &typ07 &typ08 &typ09 &typ10 &typ12 &typ14
&typ15 &typ16 ;
%LET types = &types &typ17 &typ18 &typ19 &typ22 &typ23 &typ24
&typ26 &typ27 ;
%LET types = &types &typ28 &typ29 &typ30 &typ31 &typ40 &typ41
&typ70 &typ90 ;
%LET types = &types &typ91 &typ92 &typ93 &typ94 &typ95 &typ96
;
```

%PUT &company;
%PUT ®ion ;
%PUT &types ;

%MEND queswndo ;
%queswndo ;

CONCLUSION

I have presented two techniques for obtaining user criteria, the first using %PUT and %INPUT along with &SYSBUFFR, and the second using the %WINDOW and %DISPLAY statements. If the responses sought are single values, a date, one variable name, or one value, then using %PUT and %INPUT is my first choice. It is simple and quick to code and virtually maintenance-free, depending on the prompts used. However, if there are multiple, static choices or you want to present your clients with all the questions at once, the %WINDOW statement, while requiring more up front programming, is much more elegant, flexible, user-friendly, and comes closer to being error-free. Both of these methods are extremely useful and I trust they will prove to be valuable tools in meeting the reporting needs of your clients.

SAS and SAS/AF are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® Indicates USA registration.

Other brand name and product names are registered trademarks or trademarks of their respective companies.

REFERENCES

SAS Institute Inc. (1990), *SAS ® Guide to Macro Processing, Version 6, Second Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1989), *SAS/AF ®Software: Usage and Reference, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

Michael A. Mace
1152 E 344 ST
Eastlake, OH 44095-2942
440-953-8924 (H)
330-796-3981 (W)
e-mail: michael.a.mace@worldnet.att.com