

## Paper 91

## Using SAS® Software Window® to Dynamically Manage Your Routine Programs

Linda C. Gau, CORE INC., Irvine, CA

**ABSTRACT**

Have you ever faced the problem that you have to run several related SAS programs monthly or quarterly, but sometimes you just need to run only part of the routine programs. One of the most important aspects of the SAS Window is its ability to create a user-friendly customized window for a novice SAS programmer to run routine programs dynamically. You may get results from a computer without necessarily having any particular programming skill.

**INTRODUCTION**

A menu is a list of choices that you can select from. The easiest way to implement a menu is to associate a code, a character or number, with each option. You enter one of the codes in the selection field. The program then responds according to your selection.

The %WINDOW statement creates customized windows controlled by the macro processor. Macro windows allow you to display text and input fields. For many applications, SAS Macro windows are sufficient. For more complicated applications, such as those requiring to recognize a particular cursor position or to perform automatic field validation, you must use SAS/AF® software.

Before command and message lines, the window name is at the top of the window.

You can use the %WINDOW statement either inside or outside a macro definition. You must define a window before you display it. Once defined, a macro window exists until the end of the SAS session, and you can display a window or redefine it at any time.

Defining a macro window within a macro definition causes the macro processor to redefine the window each time the macro executes. If you repeatedly display a window whose definition does not change, it is more efficient to define the window outside a macro or in a macro that you need to execute once only.

**CODES**

```
%window ADJUST irow=4 rows=22
icolumn=10 columns=60
#5 @12 "AGE/GENDER ADJUSTMENTS
      MENU" attr = highlight
#7 @08 "Select to Execute: "
#9 @08 "1 AVERAGE PAYMENTS"
#10 @08 "2 ADMISSION RATES by
      SETTING"
#11 @08 "3 ADMISSION RATES by
      SERVICE"
#12 @08 "4 AVERAGE PAYMENTS by
      SETTING"
#13 @08 "5 AVERAGE PAYMENTS BY
      SETTING and
      SERVICE"
#14 @08 "Q QUIT"
#16 @29 getdata 1 attr=underline
#18 @29 "Press" @35 "ENTER"
      attr=underline
      @41 "to continue.";
```

```

%macro analyzew;
  %display ADJUST;

  %if &getdata = 1 %then %do;
    %include "~/pmtadj1.sas";
    %include "~/pmtadj2.sas";
  %end; %else
  %if &getdata = 2 %then %do;
    %include "~/admadj1.sas";
    %include "~/admadj2.sas";
  %end; %else
  %if &getdata = 3 %then %do;
    %include "~/seradj1.sas";
    %include "~/seradj2.sas";
  %end; %else
  %if &getdata = 4 %then %do;
    %include "~/pseadj1.sas";
    %include "~/pseadj2.sas";
  %end; %else
  %if &getdata = 5 %then %do;
    %include "~/spyadj1.sas";
    %include "~/spyadj2.sas";
  %end; %else

  %if %upcase(&getdata) = 'Q' %then
  %do;
    stop;
  %end; %else %then %do;
    %let sysmsg=
    %upcase(&sysmsg) Is not valid,
    enter option on line above.;
    %display ADJUST;
  %end;
  %let getdata = ;
%mend analyzew;

%analyzew

run;

```

## EXPLANATION

The %WINDOW statement defines a window. The pointer controls used in the WINDOW statement have the same function as the PUT statement. The # pointer control can be used either with a numeric constant or a numeric variable to move the line pointer to a particular line. The @ pointer control also can be used with a numeric constant or a numeric variable to move the column pointer to a particular column on the current line. Video features can be specified with the *ATTR = option*, including *ATTR = UNDERLINE* to underline a field, and *ATTR = HIGHLIGHT* to make a field brighter. The %DISPLAY statement displays macro windows. When you create a field for input, it is very important to assign the length of the input field to the macro variable name. Assigning a length is important because some common macro program statements (e.g. %GLOBAL, %LOCAL) create macro variables with null values. If you define a macro variable in those statements and do not give a length for input, the macro processor does not allow you to input any characters into the field.

A menu is followed by %IF-%THEN / %ELSE blocks to implement the option you choose from, and call the sub-programs. You can write a error message line by assigning a value to the automatic macro variable SYSMSG for invalid input. Because we are not sure whether users will input lowercase or uppercase letters to a character option, it is suggested to translate the user's response to uppercase by the UPCASE function.



