

PROC PMENU and SCL: Forging the Missing Link for Pull-down Menus in Applications

Kernon Gibes, The NutraSweet® Company

Introduction

At NutraSweet R&D familiarity with pull-down menus is common because of the widespread use of Macintosh personal computers. Thus, the PMENU facility of SAS* 6.06 is a very natural user interface to exploit in designing end-user applications. However, there is no direct link between the PMENU facility and the SAS application language called Screen Control Language (SCL). The PMENU facility submits text strings to the Display Manager (DM) command processor for execution as opposed to a conditional branch to a SCL program label. What is desired is a method whereby a single SCL program can take action depending upon the user's PMENU selections. Of course, it is easy to have a customized PMENU call an AF program by having the PMENU submit a DM AF statement. However, this doesn't provide any mechanism for passing parameters, such as the user's response to a dialog box. This apparent lack of a link would be a severe limitation on the usefulness of the PMENU facility if it were not the case that a link can be forged.

The Missing Link

The opportunity to forge a link between the PMENU facility and SCL is provided within the SAS Application Facility (AF) through the use of the SCL CONTROL statement. Although the Display Manager will process any *valid* DM command issued by a PMENU, unrecognized commands will be passed to the window's SCL program provided CONTROL had been set to *ALWAYS* in the programs INIT section. This aspect of the SCL CONTROL feature provides the essence of the missing link. The four parts of the PMENU-SCL link can be summarized as follows:

1. Code the PMENU so that it issues invalid DM statements for customized PMENU functions.
2. In the INIT section of the SCL program issue a *CONTROL ALWAYS* statement.
3. Specify the PMENU name in the "Command menu" field of the screen's GATTR window.
4. In the MAIN section of the SCL program provide constructs which will trap and process these specific invalid DM strings.

Note that these four parts must *all* be performed. If the SCL program doesn't specify CONTROL as ALWAYS or doesn't handle the invalid DM command, an error message will appear in the message area of the active window (provided the window has a message area).

It is convenient to have a uniform method of issuing invalid DM commands so that customized PMENU functions are not inadvertently recognized as DM commands and processed before control returns to your SCL program. One approach would be to preface commands to be trapped by the SCL program with an underscore character. Not only is it the case that no valid DM command starts with an underscore, but also an underscore is not considered to be a keyword delimiter by the DM. Hence, the SCL program will trap and process DM commands whose first word is of the form *_name* where *name* is the character string constructed by the PMENU for a given customized function.

The specific details of how these steps are performed can best be explained by examining a real application.

Sensory Evaluation Analysis System

The Sensory Evaluation Analysis System (SEAS) is a SAS 6.06 AF end-user application which allows NutraSweet R&D Sensory Department analysts to maintain and analyze sensory data that was collected by use of a PC based network running the CompuSense CSA** automated-booth software. Panelists (employee or local area consumers) answer computer (CSA) generated questions using a light pen. The responses are automatically coded and written to a DOS file on the network's file server. When the panel is finished, the CSA files are copied to a MicroVAX where a SAS batch job reads the coded CSA files into SAS data sets. At this point the SEAS application can then be used to make any data corrections that are necessary or to request that the data be analyzed. The first window presented to the user displays information on a "default" sensory study via screen fields linked to a SAS data set. The default study is always the last study that had been copied to the VAX. Other studies (observations of the SAS data set) are accessible by using customized menu functions or simply using the page up/down keys. The menu part of this screen is reproduced in Figure 1, below, where we have assumed that the user has selected the pull-down menu function *edit*, followed by selecting the *Judge Data* selection.

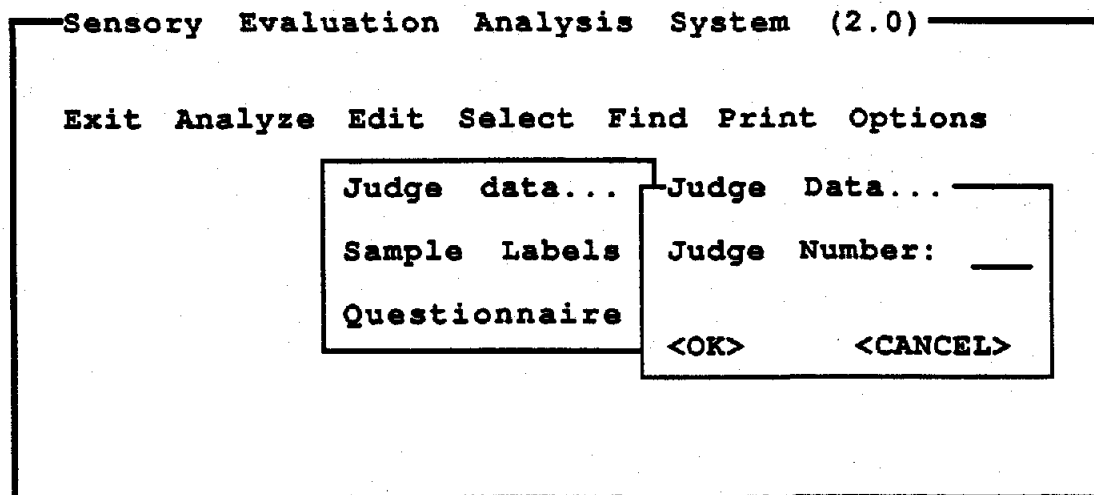


Figure 1: SEAS AF Main Window Screen

This window contains a reference to PMENU "MAIN" in the GATTR field "Command menu". Part of the SAS code which defines this PMENU is,

```
PROC PMENU CATALOG=AF.SEAS DESC "Main Command Menu";
menu main;
item 'Exit'      selection=exit;
item 'Analyze'  selection=analyze;
item 'Edit'     menu=edit;

/* ...more items... */

menu edit;
item 'Judge Data'  dialog=edata;
item 'Sample Labels' selection=elabels;
item 'Questionnaire' selection=equest;

    dialog edata '_EDITDATA @1';
        text #2 @2 'Judge Number';
        text #2 @16 len=3 attr=underlin;

/* ...more items, etc. ... */

quit;
```

where we are focusing on the "Edit" pull-down menu item. Suppose that the user chooses "Edit", then "Judge Data" and enters a "1" in response to the dialog box. This would cause the text string "_EDITDATA 1" to be submitted to the DM. Since "_EDITDATA" is not a valid DM command, it would be passed to the controlling SCL program. The corresponding SCL code fragment is,

```

init:
  control always;
  call execcmdi('PMENU ON');
  log_id = open( 'share.datalog', 'U' );
  call set( log_id );
  rc = fetchobs( log_id, 1 );
  return;

main:
  cmd = upcase( word(1) );
  if cmd ^= ' ' then do;
    select( cmd );
      when( ' _EDITDATA' ) do;
        judgeNo = input( word(2), 3. );
        call display( 'edit', edit_dsid, judgeNo, rc );
        /* ...other processing on this request... */
      end;
    end;
  end;
  /* ...trap and process remaining PMENU functions... */
end;
/* ...other MAIN SCL code... */
return;

term:
  call close( log_id );
  call execcmd('PMENU OFF');
  return;

```

where we have focused on the code to handle the request to edit judge data. Besides issuing the CONTROL ALWAYS SCL command in the INIT section, the PMENU facility is turned on and information from a SAS data set called DATALOG is mapped to screen variables. In the MAIN section of the program, we have illustrated one way to “trap” and handle the edit request. Command line text is retrieved using the SCL WORD command and examined for occurrences of text strings which were designed in the PMENU code given above. Specifically, if “_EDITDATA” is encountered, we know that the user must have chosen the “Edit” pull-down menu option and the “Judge data” selection. At this point, the SCL code would take any action consistent with these choices. Illustrated here is the retrieval of the judge number specified by the user in the dialog box (see Figure 1) and the calling of another SCL program window, *edit*, to handle additional processing on the specified judge number.

Discussion

The ability to link a pull-down menu built using PROC PMENU with a SCL program provides a great deal of flexibility and power all within a user

interface which is very "user friendly" and increasingly popular. One might argue that the link could be made easier by providing additional actions that a PMENU could perform beyond only submitting DM commands. For example, issuing a CALL DISPLAY SCL command or conditional branching to a SCL label. But the link is possible and not too overly tedious. A more important feature that would aid in application development would be to allow PMENU code to be edited and re-compiled from within a PROC BUILD session. Currently, although compiled PMENU code is saved as a catalog entry, the source code cannot be compiled from within a catalog. Therefore, debugging complicated links of PMENU specifications with SCL programs cannot be done solely by using the wonderful TESTAF command of PROC BUILD. Any error or oversight involving the PMENU construct forces one to leave the BUILD environment (at least temporarily) to correct the problem by editing and re-running the PROC PMENU code (contained in an external file or, as some have used, a stand alone SCL program with a submit block).

Trademark Notice

*SAS is a registered trademark of SAS Institute Inc., Cary NC, USA.

**Computerized Sensory Analysis (CSA) is a registered trademark of CompuSense Inc., Guelph, Ontario, Canada.

NutraSweet is a registered trademark of The NutraSweet Company, Deerfield, IL, USA.

Author Information

Kernon M. Gibes
The NutraSweet Company
601 E. Kensington Road
Mt. Prospect, IL 60056
(708) 506-2873

Fax: (708) 506-4345

e-mail (Internet): gibes@swirl.monsanto.com