COPYING SAS PROC DEFINITIONS USING THE ISPF MODEL COMMAND
Rolf Steenge, KLM Royal Dutch Airlines

Summary

The MODEL command of the ISPF editor gives the possibility to copy predefined source code into a program. This paper presents a way to automatically implement a new class of models consisting of SAS PROC definitions as contained in the SAS help library. After the model has been copied into the program, it is modified by the SAS programmer for the specific application. These models are fully accepted by endusers as useful tools for writing SAS programs.

The MODEL command

A ISPF model is a predefined set of statements which can be copied into a program with the MODEL command. ISPF offers models for developing TSO dialog applications. These models are organized in different model classes. Each class defines a set of models for a particular language, such as COBOL, FORTRAN and the TSO command language. This class concept is implemented in a general fashion, which allowed us to create a new class of SAS models for our endusers.

Using models

SAS models are used when an enduser wants assistance in writing his SAS program. After opening the dataset for edit, the MODEL command is entered in the command area of the display e.g.:

```
EDIT --- XB0113.SEUGI85.SASPGM ---------------------------- COLUMNS 001 072
COMMAND === model SCROLL === HALF
****** ******************************************************** TOP OF DATA ********************************************************

```

If an existing dataset is edited the line commands A (after) or B (before) are necessary to determine where the model has to be placed.
After pressing the enter key a selection menu is presented showing all the available SAS models. Each model can be selected by typing in the full name or the model code e.g. R5:

```
OPTION === R5_ SAS MODELS

REPORTING
  R1 CALENDAR
  R2 CHART
  R3 FORMS
  R4 PLOT
  R5 PRINT
  R6 TIMEPLOT

SAS/FSP
  F1 FBROWSE
  F2 FSCALC
  F3 FSEDIT
  F4 FSLETTER
  F5 FSPRINT

SAS/GRAPH
  G1 GCHART
  G2 GMAP
  G3 GPLOT
  G4 GPRINT
  G5 GREPORT
  G6 GLIST

DESCRIPTION
  D1 FREQ
  D2 MEANS
  D3 SUMMARY
  D4 TABULATE
  D5 UNIVARIA
  D6 UNIVARIA

UTILITY
  U1 APPEND
  U2 CONTENTS
  U3 COPY
  U4 DATASETS
  U5 DELETE
  U6 FORMAT
  U7 PRINTTO
  U8 RELEASE
  U9 SORT
  U10 TRANSPOSE

MISC
  M1 HEADER
  M2 COMMENT

Enter END command to cancel MODEL command.
```

The desired model is copied into the program. The notes in the model provide a short description of the procedure and are identified by the characters "=NOTE=" in the line command area. These notes disappear automatically after saving the program. The SAS MODELS selection menu can be skipped by specifying the model name in the MODEL command e.g. MODEL PRINT.

```
EDIT --- XB0113.SEUGI185.SASPGM ------------------------------- COLUMNS 001 072
COMMAND === SCROLL === HALF
***** ********************** TOP OF DATA ***************************
=NOTE= The PRINT procedure prints the observations in a SAS data set,
=NOTE= using all or some of the variables. Totals and subtotals for
=NOTE= numeric variables can also be printed.
000001
000002 PROC PRINT <DATA=SASdataset N UNIFORM DOUBLE ROUND LABEL
000003   SPLIT=splitchar >;
000004 <VAR variables;>
000005 <ID variables;>
000006 <BY variables;>
000007 <PAGEBY variable;>
000008 <SUM variables;>
000009 <SUMBY byvariable;>
```

The enduser can now fill in the desired SAS datasetname and other options or statements. Note that the enduser did not have to specify the model class. This is because the model class defaults to the dataset type. SAS programs at KLM have the dataset type SASPGM, which is also chosen as the model class name.

Besides the SAS proc definition models two models are available (a program header and a comment box) to facilitate the documentation of a SAS program.
How to implement the SAS models?

To implement the SAS models the following 3 steps have to be performed:

1. Convert the SAS help members to TSO panels in the ISPF skeleton library

Run the following SAS program:

```
X ALLOC F(HELP) DA('DO.SASHELP') SHR ; /* SAS HELP LIBRARY */
X ALLOC F(OUT) DA(SPF.SLIB) OLD ; /* SPF SKELETON TEST LIBRARY */

%MACRO PANEL (INMEM,OUTMEM) ;
  DATA _NULL ;
  INFILE HELP(&INMEM) ;
  FILE OUT(&OUTMEM) NOTITLES ;
  INPUT LINE $CHAR72.;
  RETAIN NOTE 'Y';
  IF LINE EO : 'OPROC'
    OR LINE EQ : 'PROC'
    THEN DO;
    NOTE = 'N';
    SUBSTR(LINE,1,1) = ' ';
    PUT ' ' ;
    END;
  IF LINE EO : 'Owhere'
    OR LINE EQ : 'Ooptionlist'
    THEN DO;
    NOTE = 'Y';
    SUBSTR(LINE,1,1) = ' ';
    PUT ' ' ;
    END;
  IF NOTE EO 'y' THEN LINE = ')N ' || LINE ;
  PUT LINE $CHAR80. ;
%MEND PANEL;

REPORTING

%PANEL(CALENDAR,XBMODR01) ; %PANEL(PLOT,XBMODR04) ;
%PANEL(CHART,XBMODR02) ; %PANEL(PRINT,XBMODR05) ;
%PANEL(FORMS,XBMODR03) ; %PANEL(TIMEPLOT,XBMODR06) ;

DESCRIPTIVE

%PANEL(FREQ,XBMODD01) ; %PANEL(TABULATE,XBMODD04) ;
%PANEL(MEANS,XBMODD02) ; %PANEL(UNIVARIA,XBMODD05) ;
%PANEL(SUMMARY,XBMODD03) ;

UTILITIES

%PANEL(APPEND,XBMODU01) ; %PANEL(FORMAT,XBMODU06) ;
%PANEL(CONTENTS,XBMODU02) ; %PANEL(PRINTTO,XBMODU07) ;
%PANEL(COPY,XBMODU03) ; %PANEL(RELEASE,XBMODU08) ;
%PANEL(DATASETS,XBMODU04) ; %PANEL(TRANSPOS,XBMODU09) ;
%PANEL(DELETE,XBMODU05) ; %PANEL(FORMAT,XBMODU06) ;

FSP

%PANEL(FSBROWSE,XBMODF01) ; %PANEL(FSLETTER,XBMODF04) ;
%PANEL(FSCALC,XBMODF02) ; %PANEL(FSPRINT,XBMODF05) ;
%PANEL(FSEDIT,XBMODF03) ; %PANEL(FSLETTER,XBMODF04) ;

GRAPH

%PANEL(GCHART,XBMODG01) ; %PANEL(GPRINT,XBMODG04) ;
%PANEL(GMAP,XBMODG02) ; %PANEL(GREPLAY,XBMODG05) ;
%PANEL(GPLOT,XBMODG03) ; %PANEL(GSLIDE,XBMODG06) ;
```
2. Create the SAS MODELS panel in the ISPF panel library with the ISPF editor

```sas
%---------------------------
SAS MODELS
--------------------------------------
%OPTION ==9> ZCMD
% Reporting %Descriptive %Utilities
% R1 +CALENDAR %D1 +FREQ %U1 +APPEND
% R2 +CHART %D2 +MEANS %U2 +CONTENTS
% R3 +FORMS %D3 +SUMMARY %U3 +COPY
% R4 +PLOT %D4 +TABULATE %U4 +DATASETS
% R5 +PRINT %D5 +UNIVARIA %U5 +DELETE
% R6 +TIMEPLOT
% SAS/FSP %SAS/GRAPH %U8 +RELEASE
% F1 +FSBROWSE %G1 +GCHART %U9 +SORT
% F2 +FSCALC %G2 +GMAP %U10 +TRANSPOS
% F3 +FSEDIT %G3 +G PLOT
% F4 +FSL ETT E R %G4 +GPRINT
% F5 +FSPRINT %G5 +GREPLAY %M1 +HEADER
% F6 +GSL IDE %G6 +GSLIDE %M2 +COMMENT
+
+
+
+Enter%END+command to cancel MODEL command.+}
)INIT
.CURSOR = ZCMD
IF (&ISRMDSPL = 'RETURN ')
.RESP = END
)REINIT
REFRESH(ZCMD)
)PROC
 &ZSEL = TRANS(TRUNC(&ZCMD,').')
 R1,'PGM(ISRECMBR) PARM(XBMODR01)'
 CALENDAR,'PGM(ISRECMBR) PARM(XBMODR01)'
 R2,'PGM(ISRECMBR) PARM(XBMODR02)'
 CHART,'PGM(ISRECMBR) PARM(XBMODR02)'
 ...... other translate specifications
 ...... G6,'PGM(ISRECMBR) PARM(XBMODG06)'
 GSLIDE,'PGM(ISRECMBR) PARM(XBMODG06)'
 *,'?' )
 IF (&ZSEL = '?')
 &ZEMTEMP = &ZCMD
 .MSG = ISRYM013
 &ZCMD = '
 &ISRMEND = 'N'
 /* SET THE END INDICATOR TO NO */
 IF (.RESP = END)
 /* IF ENDING, WHY ... WHO CAUSED */
 IF (&ISRMONC1 = 'Y')
 /* MAKE SURE ITS NOT A CLASS OP. */
 IF (&ISRMDSPL = 'RETURN ')
 /* MAKE SURE ITS NOT END ON MBR. */
 &ISRMEND = 'Y'
 /* NO - ITS BECAUSE USER HIT END */
)END
```

343
3. Adapt the ISPF panel ISREMCLS

Add a new model class in the BODY section of the panel definition:

```plaintext
)ATTR
)BODY
$--------------------------- MODEL CLASSES ---------------------------$
$OPTION --> ZCMD %
%
% 1 +CLIST - ISPF services in CLIST commands
% 2 +COBOL - ISPF services in COBOL programs
% 3 +EXEC  - ISPF services in EXEC commands
% 4 +FORTRAN - ISPF services in FORTRAN programs
% 5 +MSGS  - Message format
% 6 +PANELS - Panel formats and statements
% 7 +PLI   - ISPF services in PLI programs
% 8 +SKELS - File tailoring control statements
% 9 +PASCAL - ISPF services in PASCAL programs
%10 +SASPGM - SAS models (KLM)
%
%
......
......

Change the PROC section of the panel definition:

```
9, 'PANEL(ISREMCL)',
PASCAL,'PANEL(ISREMCL)', /* PASCAL PROGRAM SERVICE MODELS */
10, 'PANEL(XBMOD01)', /* KLM : SAS models */
SASPGM,'PANEL(XBMOD01)', /* KLM : SAS models */

......
```

Refer to the SAS PROC MODELS panel in the ISPF panel library

Final remarks

The implementation of the SAS models only took us a few hours, because all the text of the models could be retrieved from the SAS help members. The approach also guarantees easy update of these models for future SAS releases unless the structure of the SAS help library is changed. The impact on the standard ISPF software is minimal. Only a small panel modification is required.

From the reactions of our endusers it can be concluded that this SAS model facility is a useful tool in writing SAS programs. Typing errors in the procedure options are less likely to occur. Also less time have to be spent in consulting the SAS documentation for finding the correct syntax. Finally this model facility can provide the enduser with site dependent information such as Job Control, plotting instructions etc. We think that it would be a good suggestion to incorporate this kind of facility in the Display Manager.

If further information is required please contact: Rolf Steenge
KLM Royal Dutch Airlines
P.O. Box 7700
1117 ZL Schiphol
the Netherlands
tel: (31) 20 491314