The SAS® System for SMP/E conversions

K.L Kanthan

Chemical Bank

Abstract:

System Modification Program/Extended is an IBM product designed to install and modify Software products. But not all the software come in SMP/E installable format. SAS makes the conversion of non SMP/E to SMP/E installation easier. Many utility functions that are needed for the conversion are achieved through SAS code and PROCs.

1. Introduction:

In any large installation there can be multiple IBM mainframes and several software products at different versions and service levels. Manually managing and tracking the software at several levels become more difficult as the sites become larger. System Modification Program / Extended (or SMP/E) makes this task easier. But not all the software come in SMP/E installable format. For those products that are in non SMP/E format, conversion to SMP/E format can made easier through SAS. Some utility functions can be built with SAS to achieve this goal.

Some of the most common functions that are involved in the conversion of the software installation with SMP/E are as follows:

- Set up SYSMODS based on the following details
- Create SMP/E MCS control statements
- Create SYSTIN statements for IEBCOPY
- Find duplicate member names in the target library for MAC, MOD, LMOD, SRC
- Check to see if module NAME and CSEC names have overlapping addresses.
- Create SYSTIN statements for IEBUPDTE etc.

Many of the above functions can be done through manual editing only if the software product is not very complex. For large software packages involving several different source, macro, load, object and panel libraries, manual editing can be cumbersome and may give rise to errors. Some simple SAS programs may be developed to assist this conversion. The following section describes how SAS was used to attain these goals.

2. Methodology:

SMP/E assists the installation of products by creating distribution, target library members from the SYSMODS. Functions are groups of related program elements such as modules, source, and macros. SMP/E can install preventive, corrective and user modification to the target and distribution libraries. SYSMODS are made up of MCS (Modification Control Statements). These control statements can be ++FUNCTION, ++PTF, ++USERMOD, ++VER, ++ICLIN, ++MOD, ++ZAP, ++SRC, ++MAC and so on. These may be headers to identify the type of modifications, specify how the SYSMOD relates to other SYSMODs, module replacements, source and macro replacements etc.

In this context SAS procedure SOURCE can be very handy since it can print the contents of a library, or process the directory of a library to be input for utility or other programs, route the members of a library to other programs, print the alphabetical order of member list, select members in a particular range, give the number of records in each member and so on.

Some examples of module, macro, source replacements will be shown below. In this connection it is also necessary sometimes to check for duplicate elements, since the SMP/E will not allow the installation of the duplicate named elements.

One of the most common requirement is to set up an IEBCOPY and SYSTIN control statements based on the member names of a given PDS. The format of these statements follow:

S m=xxxxxxxx
S m=yyyyyyyy
...            
S m=(zzzzzzzzz,R)

where xxxxxxx, yyyyyyyyy, ... are the member names in the PDS.

This can be easily done using PROC SOURCE for the given PDS. Listing of this job with the SAS code is shown in Job_1 Appendix. Here the DDname, INDDI, points to the PDS for which the member names are to be selected. Note that SAS has flexibility to select or exclude the members with a given prefix. Formats of these statements are marked (1). SAS substitutes the
member names in position marked xxxxxxx on statement marked (2), and generated the lines based on selection criteria. The first few lines of the SYSIN statements are shown in (3). Generated SYSIN statements can be passed on to the next step or written to a permanent dataset by changing the DDname, OUTDD1, in the first step.

To set up a user SYSMOD one needs to generate SMP/E statements defining the SYSMOD, and some SMP/E MCS control statements of the following forms:

```
++MOD (xxxxxxxx) LKLIB(loudlib) DIST(anklib).
++MAC (yyyyyyyy) TXTLIB(loudlib) DISTLIB(anklib).
++SRC (zzzzzzzz) TXTLIB(loudlib) DISTLIB(anklib).
++FUNCTION
++VER
```

Here again xxxxxxx, yyyyyyyyy, zzzzzzzzz indicate the member names from a given PDS.

JOB_2, shown in Appendix, reads the member names for the dataset with DDname INDD1 and generates the +++MAC(yyyyyyyy) card images, and appends it to the images coded in the 'FIRST' statements. These generated card images, by altering the OUTDD1 statement can be used either to write to a permanent dataset or passed on to the next step. The statements marked (1) describe the userMOD, and (2) the SMP/E control cards for macro, SRC etc. libraries.

When installing elements, SMP/E will not allow duplicate names. Let us take the example of SAS installation with SMP/E. There are some member names common to load library, Macauto, and sashelp. Construction of the function SYSMOD and USERMODS are dictated by these duplicate names. Members LEFT and VERIFY are in three libraries; namely load library, MACAUTOS, and SASHELP. Similarly, members TRIM is in MACAUTO, and SASHELP. These factors have to be considered in the building of the FUNCTION SYSMOD and USERMODS. For this example, LEFT and VERIFY will be installed as SRC elements at the same time the load modules are installed. After that, Macauto and sashelp can be installed as MAC elements. (See example A in the Appendix)

It is essential to check to see if the same name is used by another product. An example of a program to list the common names between two PDSs is shown in JOB_3 in Appendix. Primarily, this job reads the two PDSs referred with DDnames IN1 and IN2 and prints the member names that are common to both. This PRINT statement is marked (1). This program also has a few more lines of SAS code that is commented out, which if needed, would print the member names that are unique to IN1 and IN2 only. This is marked as (2).

JOB_4 is used for generating IEBUPDTE jobstream. This reads the member names from the PDS pointed by DDname INPUT and creates the '/ ADD ' cards and also adds the contents of the members after this. By changing the dataset definitions in DDname OUTPUT, the generated card images may be written to a permanent dataset or submitted through the internal reader (INTRDR), for job execution. See JOB_4 in Appendix. This can be very useful in handling the PDSs of object modules.

3. Conclusion:

SAS system is very powerful, flexible and easy to implement different types of applications. Use of SAS to generate the JOB streams and control statements came in very handy in covering SMP/E installations. Use of SAS for the conversion considerably reduces the errors that can occur when manual processing these types of functions.

Acknowledgements:

The author wishes to thank Joan Payne who reviewed the manuscript and made valuable comments.

SAS® is the registered trademark of SAS Institute, Inc., NC, USA