ABSTRACT:

Although SAS has long been popular with end-users and application development staffs, in the past few years more people in the technical support area have discovered that its ability to read and analyze any type of data make it particularly useful for many technical applications. Instead of installing it and forgetting it, we are using it for performance analysis and monitoring, for accounting, and for data conversion. Here is one technical application we have found very helpful in managing a VM installation.

PURPOSE:

The implementation of High Performance Option for VM/SP and the addition of support for new hardware configurations has made the job of managing VM even more complicated. In addition to the base version of VM we can now have several layers of additional code to keep track of. If a change is sent by IBM to a particular module we now need to insure that it is applied to the correct version of the module. And the CP nucleus map provided by IBM's mapping program can sometimes be little help. For example, if VM is down and the support center wants to know if a particular APAR has been applied, it is necessary to consult a manually kept log or to wade through the nucleus map looking for the affected modules. Or if they ask if we're sure a module has been HPO versioned, we start turning pages in the map while we put them on hold. So keeping VM current now requires people who, in addition to being technically skilled, are good organizers and meticulous record-keepers.

So this simple SAS program is offered as an automated way to provide change management by analyzing IBM's CP nucleus map and producing seven reports. Some of the reports furnish cross references into the nucleus map while others keep track of changes made to the nucleus by comparing it with a 'base' map. This facility can be used to keep track of changes made since a version or put level change, or just those changes made since the last nucleus was built.

The seven reports produced by this SAS application are:

1. A cross reference listing the modules with assembly date, type of module, and the module's entry location. This report is sorted by module name (instead of by entry location as in IBM's map).
2. A listing of the above information by version, so that all base modules are together, all HPO versioned modules are together, etc.
3. A listing of the above information by date so that all recently assembled modules are together (and all those modules that haven't been reassembled in years are quickly identified).
4. A listing of all APARs, with the APAR description, the affected module, the module's assembly date, and the module's entry location.
5. A listing of all modules which have been reassembled since the last nucleus build.
6. A listing of modules which have changed version since the last nucleus.
7. A listing of any APARs which are missing from or added to the new nucleus.

NOTES ON RUNNING THE PROGRAM:

This SAS program runs under SAS 82.3 or Version 5 with no changes required. The program issues a CMS filedef for 'CPNUC MAP *' which is the map of the nucleus just built. It also issues a filedef for 'CPNUC OLDMAP *' which it uses as the base to compare against. Both of these maps are necessary. If you decide to compare the new nucleus with the one previous to it, 'age' the maps through by renaming the CPNUC MAP to CPNUC OLDMAP before you map the new nucleus. You might want to keep as the 'old' map the one created on a version or put level change. Or you can keep several old maps around and rename them and rerun the program several times.

THE 'CPNUC SAS' PROGRAM:

```sas
/******************************************
/* CPNUC SAS */
/******************************************
OPTIONS TMSG=OFF ERRORS=O;
CMS FILEDEF CPOLD DISK CPNUC OLDMAP *;
CMS FILEDEF CPNEW DISK CPNUC MAP *;
MACRO PROCESS
DATA &CP (KEEP=MODULE ENTRY TYPE SASDATE DATE)
SAPAR
(KEEP=MODULE ENTRY TYPE APAR SORT DESC);
RETAIN READFLAG 1;
RETAIN READ MODULE TYPE DATE SASDATE;
DROP READ AT;
INFILE &CP;
IF READFLAG = 1 THEN /* Look for */
DO;
/* modules. */
```
INPUT READ $ 2-5 /* These are */
    MODULE $ 8-15 /* module */
    TYPE $ 17-23 /* records. */
    DATE $ 35-55
%36 SASDATE MMDDYY8.;
IF READ = 'READ' THEN READFLAG = 0;
END;
ELSE
    INPUT AT $ 17-18 /* These are */
        ENTRY $ 20-25 /* APAR */
        APAR $ 26-33 /* records. */
        SORT $ 27-31
        DESC $ 34-96;
    IF SUBSTR(APAR,7,2)='DK'
    THEN OUTPUT &APAR;
    IF AT = 'AT' THEN DO;
        APAR = ''; OUTPUT &APAR;
        READFLAG = 1;
    END;
END;

%LET CP = CPOLD;
%LET APAR = APAROLD;
PROCESS
%LET CP = CPNEW;
%LET APAR = APARNEW;
PROCESS
PROC SORT DATA = CPNEW;
    BY MODULE;
PROC PRINT UNIFORM;
    ID MODULE;
    VAR TYPE ENTRY DATE;
    TITLE 'VM/SP NUCLEUS';
    TITLE2 'BY VERSION, MODULE';
PROC SORT DATA = CPNEW;
    BY TYPE MODULE;
PROC PRINT UNIFORM;
    ID MODULE;
    VAR TYPE ENTRY DATE;
    TITLE 'VM/SP NUCLEUS';
    TITLE2 'BY VERSION, MODULE';
PROC SORT DATA = CPNEW;
    BY SASDATE MODULE;
PROC PRINT UNIFORM;
    ID MODULE;
    VAR TYPE DATE ENTRY;
    TITLE 'VM/SP NUCLEUS';
    TITLE2 'BY ASSEMBLY DATE';
PROC SORT DATA = APARNEW;
    BY MODULE;
PROC PRINT UNIFORM;
    ID APAR;
    VAR MODULE DESC;
    TITLE 'VM/SP NUCLEUS';
    TITLE2 'BY APAR, MODULE';
PROC SORT DATA = CPOLD;
    BY MODULE;
PROC PRINT DATA = CHGDATE;
    ID MODULE;
    VAR NEWDATE OLDENTRY NEWENTRY;
    TITLE 'CHANGES TO VM/SP';
    TITLE2 'MODULES WITH VERSION CHANGES';
PROC SORT DATA = APARNEW;
    BY SORT MODULE;
PROC SORT DATA = APAROLD;
    BY SORT MODULE;
DATA BOTH;
    MERGE APAROLD(RENAME=(TYPE=OLDTYPE DATE=OLDDATE ENTRY=OLDENTRY))
        APARNEW(RENAME=(TYPE=NEWTYPE DATE=NEWDATE ENTRY=NEWENTRY));
    BY MODULE;
DATA CHGAPAR;
SET BOTH;
    IF NEWDATE = OLDDATE THEN OUTPUT CHGDATE;
    IF NEWTYPE = OLDTYPE THEN OUTPUT CHGTYPE;
PROC PRINT DATA = CHGDATE;
    ID MODULE;
    VAR NEWDATE OLDENTRY NEWENTRY;
    TITLE 'CHANGES TO VM/SP';
    TITLE2 'MODULES REASSEMBLED';
PROC PRINT DATA = CHGTYPE;
    ID MODULE;
    VAR OLDTYPE NEWTYPE OLDENTRY NEWENTRY;
    TITLE 'CHANGES TO VM/SP';
    TITLE2 'MODULES WITH VERSION CHANGES';