Automated Maintenance of SAS® Software
James O. Ward, SAS Institute Inc., Cary, NC

ABSTRACT

The growth in size and complexity of the SAS® System has made the task of maintenance more difficult for the System Administrator. This paper describes an automated system for managing the maintenance of SAS® Software, and discusses the installation and use of such a system.

INTRODUCTION

Software maintenance is a "necessary evil" to keep a system running smoothly. Until "no-fault software" is invented, someone in your shop will be responsible for maintaining one or more software systems, including the SAS® System. The maintainer can also expect an increase in the time and effort required to maintain software products that are growing in complexity, coupled with the need for those products to interface with an increasing variety of hardware and software. Meanwhile, perhaps we can find ways to smooth out the process for our favorite line of software products!

The scope of this discussion will be the maintenance to SAS Software load module libraries on OS and CMS in the form of zaps distributed by SAS Institute. This comprises the bulk of distributed maintenance, the remainder usually in the form of source file updates for CLISTs or EXECs.

Maintenance Basics

The basic steps in our maintenance process include: (1) installing the maintenance and tools such as zaps, source updates and usage notes, (2) reviewing the usage notes to determine whether or not zaps are applicable or necessary, (3) selecting zaps based on the review step, (4) preparing modules or zap text for those zaps which require special handling, (5) applying selected zaps, (6) testing the resulting system, (7) accepting the altered modules back into "production" libraries, and (8) recording the results.

Note that some of these steps may be performed at the same time as others, and some (such as testing) may be considered as optional, often with disastrous results!

Maintenance "Styles"

A major factor in software maintenance is the underlying approach or management style employed. There are two main styles to consider.

The first approach is the "reactive" style, which is practiced by those who believe that "if it ain't broke, don't fix it." This style has its advantages and its devotees, but often results in spotty maintenance and recurring bouts of "fire-fighting." Good record keeping is essential to avoid back-tracking when the next problem arises.

The second approach is the "active" style, which is characterized by the term "preventive maintenance." This approach relies on the timely application of whole groups of fixes, usually at regular intervals. Preventive maintenance is preferred by many, although it may take more planning and discipline to implement and involves a good amount of "faith" in the validity of the fixes themselves.

Naturally, due to the higher volume of maintenance, the "active" style stands to gain more from improvements in the maintenance process, and it is this style we want to employ. Nevertheless, even the "reactive" style can benefit from improvements such as an automatic system of record keeping.

Introducing Automation

Improved speed, attention to detail and accurate performance of repetitive tasks are some of the goals of increased automation, and are high on the list of goals that motivate the purchase of computer systems and software. It should then be obvious that increased automation of the software maintenance process is also a desirable goal.

However, some steps cannot be automated, nor should they be. For example, the review or preparation steps must be done manually or not at all. And while it is possible to automate testing, this is most easily and commonly done by "aging" the maintained software in a staging library. Therefore, we will be aiming for a well-augmented system rather than one that is completely automated.

Existing Automated Maintenance Systems

Before we build a new maintenance system, it may be instructive to examine the availability and applicability of existing systems.

For the MVS operating system and other IBM® products running under MVS, the installation and maintenance tool is a monster known as the System Modification Program (SMP, or the newer "Extended" version, SMP/E). The DOS/VSE system, although out of the scope of this paper, provides Maintenance System History Program (MSHP). Both of these programs can be awkward to use even with IBM code, and both seem too restrictive and unwieldy for much of the non-IBM vendor code. On the VM side, there is a notable lack of a formal maintenance system such as SMP or MSHP. This is probably because an original design goal for VM was for its use as a tool to assist in the conversion from one operating system to another.

In the author's opinion, the volume of maintenance now and in the future may be more than a manual process should handle, but the automated systems that exist are inappropriate for SAS Software. Designing and supporting a customized system for our own products may be much more beneficial and appropriate.

AUTOZAP - an Automated Zap Processing System

To assist in maintaining SAS Software, the Technical Support Division of SAS Institute will be distributing and supporting AUTOZAP, a set of programs and full-screen menus which can assist in the selection and application of zaps and to maintain a reasonable set of maintenance records. This system should be available around the middle of 1989 for both OS and CMS, and will be shipped as part of the Usage Notes tape. By designing AUTOZAP as a SAS application and distributing it on the monthly usage notes tape, any necessary fixes and enhancements to AUTOZAP itself can be made on a more timely basis. What follows is a brief description of the system's design and implementation.
Design Guidelines

Of the basic maintenance steps, AUTOZAP is designed to augment or replace the selection, application, acceptance processing, and (especially) the recording of these activities, while leaving it up to you to perform the installations, review the usage notes, test zaps, review, test, and perform special preparation for load modules or zaps prior to zap application.

Some restrictions must be made on the type of processing that can be done in order to keep the system simple yet usable (following the "K.I.S.S." design principles). As a result, some manual processes will still be required. AUTOZAP is designed primarily to be used along with TSO CLISTs or CMS EXECs, meaning MVS batch execution will be under the TSO Terminal Monitor Program (TMP). Even though it is not entirely out of the question, AUTOZAP does not currently process "zaps" that are really source updates. Finally, some zaps simply require manual processing, modification, or input.

Zap Categories

One of the more useful bits of information that can help determine whether or not a zap can be applied automatically is its ZAP CATEGORY. The five categories for zaps distributed by Technical Support are:

1. REQUIRED,
2. HIGHLY RECOMMENDED,
3. RECOMMENDED,
4. SPECIAL CONSIDERATION, and
5. REMOVE.

Zaps in any of the first three categories require no special action or preparation, making them prime candidates for automatic application, either individually or in groups. The SPECIAL CONSIDERATION category is used when a module must be copied and renamed, reinked, or when special zap data must be supplied by the maintainer. The special instructions for processing one of these zaps will usually be found in the usage note or in the body of the zap itself. REMOVE is used to identify a previously distributed zap that has been withdrawn from production and should be removed (if it has already been applied); usually, a "removal" zap with a name such as "Rrrrmnn" will be supplied to "back off" the offending zap.

AUTOZAP Software and Libraries

AUTOZAP is an interactive, menu-driven application, although most of its functions may also be processed in "batch" mode. Extensive use is made of the SAS Macro Language and its autocall facility, as well as TSO and CMS commands. The need to use TSO and CMS commands is dictated primarily by having to dynamically allocate files and to invoke the system's zap processor and check its return codes. Also, since PROC PDSCOPY does not exist on CMS, the CMS LOADLIB command must be used to locate and copy members of CMS LOADLIB files.

PROC DISPLAY is available in the Base SAS System (release 5.18 and later), so you only need SAS/AF Software if you wish to customize the AUTOZAP menus or program screens. SAS/AF is not required for use of the system. Most of the SAS code making up AUTOZAP is in the form of autocall SAS macros rather than in the program screens, so some customization is still possible even without SAS/AF.

AUTOZAP uses several libraries of its own, and must have access to files such as the usage notes, zap libraries, and the load module libraries. A SAS library must be available to contain the SAS/AF catalogs and the AUTOZAP recording logs. Macros used by AUTOZAP may be installed in their own autocall maclib, or they may be installed in the same maclib used by the "SAS" CLIST or EXEC. Similarly, CLISTs or EXECs that are used by AUTOZAP may be installed in either shared or unique libraries. (The names of all AUTOZAP Macros, CLISTs, and EXECs begin with the letters "AZ" for easy identification.)

AUTOZAP will only process the first copy of a load module in a maintenance "set." Modules that have been copied into system libraries, for example, must be re/copied. One example of this situation is SASWTR, which must be available to the MVS External Writer from an authorized library in the system link list. Another (special) example is the non-overflow version of the SAS supervisor, SASLPA, which is often put in the MVS Link Pack Area (LPA). Because SASLPA may or may not exist on your system, and may or may not have the CSECTs specified by a given zap, distributed zaps do not have segments that may (or may not) apply directly to module SASLPA. SASLPA maintenance is best done by reusing the job that created it, namely the BLPA1IB installation job. (AUTOZAP will make an attempt to determine the need to run the BLPA1IB job.)

Basic Functions of AUTOZAP

Before AUTOZAP can be used to select and apply zaps, you must identify it to it a "set" of libraries, releases and products to maintain (it is possible to maintain multiple sets). Once this is done, zaps may be applied individually or in groups.

The procedure for applying an individual zap is probably the "heart" of AUTOZAP, and can be broken down into the following steps:

1. locate the zap,
2. obtain the module names for the zap,
3. copy the zap to a working file for the system zap processor,
4. as necessary, copy the module(s) to the working zap library,
5. copy the module(s) to a backup library,
6. invoke the zap processor,
7. if the zap could not be successfully applied,
8. note the failure and restore the load module(s), otherwise
9. note the success and return.

(The working zap library may be either the live SAS load module library or a "holding" library; step 4 is only done in the latter case, which is also the recommended mode of operation.)

For processing zaps in groups or batches, a selection program scans the usage notes for zaps that meet the specified criteria, and builds a list of selected zaps to be applied individually by the rules described above.

| 551 |
After zaps have been applied and tested and you are satisfied with the stability of the test system, AUTOZAP will allow you to "accept" the changes and return each altered module to its original library.

Besides zap application and module management, AUTOZAP allows you to produce reports that detail which modules have been modified or which zaps have been applied.

There are also utility functions that may be invoked to do things such as bringing modules "into the fold" for preparation prior to applying SPECIAL CONSIDERATION zaps. Each of the functions described can be selected via menu screens, or may be invoked by the underlying macros during "batch" processing.

AUTOZAP Usage Scenarios

As an example of the usefulness of AUTOZAP, consider the possibility of "delta" maintenance processing, which involves finding and applying each of the zaps that were distributed after the last batch of zaps you applied. By specifying a date value greater than the date of the previous tape, the usage notes "FIRST TAPE" field date values will be examined to find zaps to apply. Naturally, other criteria may also be specified to make the zap selection even more appropriate.

If a SAS Software product is separately installed after the base and other products, AUTOZAP can be used to bring that product's maintenance "up to snuff" by selecting all zaps for the product code used in the usage notes.

If a zap failed to apply because a pre-requisite zap had not been applied, it (or the pre-requisite zap) can be applied later as an individually selected zap.

CONCLUSION

The routine maintenance of SAS Software can be greatly enhanced by using the power of a system we are already familiar with: the SAS System itself.

SAS and SAS/AF are registered trademarks of SAS Institute Inc., Cary, NC, USA.

IBM is a registered trademark of International Business Machines Corp., Armonk, NY, USA.