ABSTRACT

ZLMS is a system developed at SAS Institute Inc. to collect, verify, maintain, and re-distribute "zaps" from a multitude of sources. The purpose of ZLMS is to assure the quality of zaps, to enhance the information regarding zaps, and to improve the process of SAS® software maintenance.

INTRODUCTION

The life-cycle of any production software can be broken into five basic phases: (1) design, (2) write (code/compile/link), (3) test, (4) distribute, and (5) maintain. ZLMS was developed to enhance the last phase, which may also be the longest.

Software maintenance comes in many "flavors". Probably the most widely used form of maintenance is the "zap" (or "patch"), which is a set of commands used by a program editing utility to alter a target program's executable form. Zap utilities and their commands allow one to verify the target program's current instructions and data, to alter the program's instructions and data, and to record status information to maintain an audit trail. The use of zaps extends the life of the software during the last phase of a complete cycle. Zaps are fairly easy to produce, distribute, and apply, and their use allows the production software to remain relatively stable.

However, unless a few simple rules are followed, zaps may not apply correctly, or may cause more problems than were (supposedly) fixed. One of the main goals of ZLMS is to minimize the occurrences of such errors.

ZAP DISTRIBUTION SYSTEM

At SAS Institute, the first step in the current process of getting a zap "out the door" is to document the problem by writing a usage note; this usage note is normally marked for "internal use only" until the zap has been approved. After the zap has been written and tested, the usage note is then updated to indicate the zap's category and whether (and how) the zap should be distributed.

Once a month, a tape is created which contains usage notes and zaps. The DIAL-A-ZAP files are also updated during this process. The usage notes are selected for inclusion on the tape and are scanned for zap type and zap availability. Zaps are then collected and written to the appropriate files. When monthly processing is complete, the zaps are also made available to be included on new product installation tapes.

Zaps can be distributed in two other ways—verbally or in printed form. These forms work best if zaps are kept short. Short zaps have a better chance for clear verbal communication and for correct transcription into machine-readable form for application. Punctuation in the data areas (when allowed by the zap utility) can also make a zap easier to read and transcribe.

System Limitations

Although the current zap system has worked well, it does have its limitations. First, there is no automatic enforcement of zap format and coding standards. Second, the zap processing programs require frequent maintenance to allow access to all required zap libraries. These libraries are not protected from further updating, so a zap could be updated and re-distributed—an event which will very likely cause confusion. Since zaps are processed monthly, new zaps may not appear on DIAL-A-ZAP for up to a month. Finally, although precautions are taken, it is possible for a zap to be shipped before being tested.

AUTOMATING ENFORCEMENT

What can be done to plug the "holes" in the system? We can solve most of the problems by automatically enforcing rules, by immediately alerting the zap developer of violations, and by protecting zaps from alteration after being distributed. If the zap developer has an automatic means of keeping track of code and "patch space" affected by his or her zaps, all the better. These goals and others have guided the design for the Zap Library Management System (ZLMS) to help assure the quality and integrity of zaps shipped by SAS Institute.

Automated enforcement by ZLMS of most of the process is accomplished by using a strict (but flexible) set of zap coding standards, SAS data libraries for record-keeping, a protected distribution zap library, and utility routines for managing zap collection, verification, printing, distribution, and conflict avoidance.
Zap format enforcement includes validating all zap commands, requiring data verification of all data areas that are replaced, and prohibiting the use of zap commands that may be invalidated by installation differences (such as the PDZAP "REF=address" command under DOS/VSE). The process also enhances zaps by automatically verifying and/or adding CHECKSUM and IDRDATA statements on OS zaps and LOG statements on CMS zaps, and by supplying standard comment headers with information from the usage note. Even minor details, such as checking for proper punctuation of hex data and limiting hex data length, are attended to so a zap developer will be made aware of their need for zaps that are transcribed from paper or over the phone.

**BENEFITS OF ENFORCEMENT**

The steps necessary in getting a zap into the system and "out the door" help assure that the problem is properly documented, that the zap was tested, and that the zap cannot be altered once it has been distributed. Some unnecessary zap text, such as unimportant comment statements or remarks on a zap command, are removed. In addition, the data areas modified by each zap are recorded, allowing developers to keep up with patch space and to check zaps for possible conflict.

With ZLMS in place, the process of distributing zaps to you is greatly improved. The JCL for collecting zaps from the many zap libraries is eliminated, and the authority to access these libraries is no problem, since zap developers themselves use ZLMS to add zaps to a test-level zap library. In addition, incremental daily updates to DIAL-A-ZAP is feasible, which is more desirable than a once-per-month batch update.

**SASLPA Considerations**

For OS sites, maintenance of base SAS software is improved, since zap segments referring to the SASLPA module are removed. Removing the SASLPA zap segments, a process many sites perform manually, can prevent several problems. SASLPA may be built differently from one site to another, and may not exist in the same library as the other modules specified in the zap (in fact, it may not be installed at all), so a zap that references SASLPA may make AMASPZAP "complain." Repeatedly applying zaps to SASLPA may also exhaust the initial supply of CSECT IDenification Records (IDRs), a situation less likely with zaps solely applied to unbundled modules. The best approach to SASLPA maintenance is to re-run the installation job "BLPALIB" after applying all zaps.

**IMPLEMENTATION**

ZLMS is implemented as a SAS/AF® application under MVS TSO, using SAS/FSP® software, SAS/SHARE® software, and lots and lots of SAS macros. Zaps are stored in two libraries, a test-level library and a limited-access production library. Other OS libraries hold user profile data, system interface routines and CLISTS, and SAS formats and informats. The system interface routines are used for services such as verifying the library access level of the user, verifying the existence of libraries and their members, manipulating OS partitioned data sets, and notifying users. SAS informats and formats are used for input validation and space conservation. (Even though the system menus and other screens require over 150 tracks on a 3350 drive, conservation of DASD space is considered quite important.)

Final testing of ZLMS for support of OS, CMS and VSE zaps began in March, with full production status expected by early April. Support for PC/DOS, VMS®, and AOS/VS zaps (better known as "patches" on these systems) may be added as early as the third quarter of this year.

**ACKNOWLEDGEMENTS**

The author wishes to thank Brian R. Perkinson for his boundless energy, ideas and work in co-authoring the Zap Library Management System. A special thanks also goes to Annette Harris for cheerfully answering endless questions on SAS/AF® software. The author also thanks Susan Marshall, Phil Herold, Martin Mincey, and Dale Ingold for assistance in testing and for their many valuable suggestions.

**CONCLUSION**

The design and goal of ZLMS is to help assure the quality of zaps you receive from SAS Institute. Its implementation will allow exploration of other areas of support, such as automating the process of applying zaps at your site.

Your comments and ideas on ZLMS are welcomed, and may be addressed to the author at SAS Institute Inc., Cary, NC, USA.

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

VMS is a trademark of Digital Equipment Corporation, Maynard, MA, USA.