ZAP To ZAPS

Jason Moore, SAS Institute Inc., Cary, NC

INTRODUCTION

This paper discusses the maintenance process adopted by SAS® Institute Inc. for Release 6.08 of the SAS System and later releases. This process is used to correct known problems, add new features, and introduce new products in the field.

To help illustrate this maintenance process, I will attempt to take you through the life of a "bug" found by users in the field - from the beginning when the customer telephoned Technical Support reporting this problem, to the time the same customer completed testing the maintenance release that corrected the problem.

Along the way I will provide insight on what goes into creating a maintenance release at SAS Institute: The coding, porting, managing, testing, distributing, and finally the installing of maintenance. I'll point out the pros and the cons of this process and why we feel this is the best way to deliver maintenance for the SAS System.

In theory, this process sounds simple: Find the bug, fix the bug, and ship the fix for the bug. But in practice, we have found this to be a complex project involving hundreds of employees working thousands of hours on the goal of delivering quality maintenance in a timely fashion.

BACKGROUND

Throughout this paper I make reference to the *maintenance cycle*. This cycle is the amount of time it takes to begin and end a specific maintenance release. I call it a cycle because once the current maintenance release is shipped, the process is repeated for the next maintenance release. There are several milestones that occur during a maintenance cycle, and I will touch on a number of these and how they effect the progress of our "bug" as it moves through this cycle.

Each maintenance release is distinguished by a unique Technical Support level, or TSLEVEL. This TSLEVEL appears at the top of each SAS LOG. It gives Technical Support the ability to confirm the latest maintenance installed at your site. This information is crucial to the Technical Support Division when contacting them with a question or problem.

RELEASE	MAINTENANCE-SHIPPED	TSLEVEL
6.08	Production/No Maintenance	TS404
6.08	1st Maintenance	TS405
6.08	2nd Maintenance	TS407
6.08	3rd Maintenance	TS410
6.08	4th Maintenance	TS415
6.09	Production/No Maintenance	TS027
6.09	1st Maintenance	TS037
6.09	2nd Maintenance	TS039

Figure 1 *TSLEVEL for Maintenance Shipped*

FIND A BUG TO TRACK

For this paper, I chose a bug that was reported by someone who contacted Technical Support with complex questions and problems. I also wanted the bug to be a "portable" problem. A portable problem is defined as one that occurs in the SAS System across multiple operating systems. Of course, the bug should also be able to be fixed with a maintenance release. I was successful in finding one that met these criteria.

This problem was originally reported as a memory problem by one of our mainframe users. This person described the problem by saying, "All we do is a DATA step, SET a permanent SAS data set on disk, and the job fails with an out-of-memory message." Technical Support asked the customer a number of questions before requesting the CONTENTS procedure be run on the data set they were attempting to read. We were surprised when the PROC CONTENTS created the following output;

```
-----Engine/Host Dependent Information-----

Data Set Page Size:          823296
Number of Data Set Pages:    1
File Format:                 607
First Data Page:             1
Max Obs per Page:            22789
Obs in First Data Page:      520
Physical Name:               SASSWM.V6COPY.PAGESIZE
Release Created:             6.08
Release Last Modified:       6.08
Created by:                  $SASJOB
Last Modified by:            $SASJOB
Subextents:                  13
Total Blocks Used:           134
```

Output 1 *Partial Proc Contents Output*

As you can see from the slide, the PAGESIZE for the SAS data set is unusually large. This was obvious to the Technical Support consultant who was working with the customer. The PAGESIZE of a SAS data set is the buffer size of the input/output buffer the SAS System uses when transferring data during processing. This page, or number of bytes, is moved from external storage to memory in one logical I/O operation. The PAGESIZE is a permanent attribute of the data set and can only be modified at creation time.

To compound the problem, the customer had the SAS System option BUFNO=10 specified. This option specifies the number of page buffers to allocate for an open SAS data set. Thus, the number of bytes the SAS System was allocating for buffer space when attempting to read this data set was 10 times the PAGESIZE or 8232960 bytes or 7.8 MEG. It is not suprising the job stopped processing due to memory constraints!

We were able to circumvent the customer's problem by having them lower the value for the BUFNO= option. This satisfied the customer, but it didn't solve the problem of the unusually large PAGESIZE.

Technical Support's first question was, "How did the PAGESIZE become so large?" Without going into details, there are SAS System options and DATA set options that allow you to specify the PAGESIZE of a SAS data set. Both of these possibilities came up negative as the user had not modified the defaults for these options.

After more investigation, the customer reported that they had used the COPY procedure to move the SAS data set from disk to tape and back to disk again, repeatedly.

In our own testing, Technical Support was able to exhibit the following behavior: Using PROC COPY to move a SAS data set to unlike media (tape to disk for example) caused the PAGESIZE to increase to the next multiple of the physical blocksize of the SAS data library. Thus, with repeated copying to unlike media, the PAGESIZE of a SAS data set could increase without bound. Note, this customer had used PROC COPY to move this data set from disk to tape, and back to disk again over twenty-five times.

Copying a data set once or twice will not create a data set with a PAGESIZE of the above magnitude.

Collecting the information to reproduce this problem was simplified in this case because the customer had done a great deal of work before contacting Technical Support. They were able to reproduce the problem after narrowing the test program down to 2 lines of source code, and they had tracked the history of the SAS data set they were attempting to read.

Needless to say, this was a problem that needed to be investigated by our development staff. Technical Support is responsible for initiating a problem report or defect when they deem it appropriate. It was certainly appropriate in this case. The defect was reported along with a test program exhibiting the faulty behavior to our development staff.

Technical Support also enters these problem reports into a data base, called SAS Notes, that are used as a tool by other Technical Support consultants when researching problems and handling your calls. These same SAS Notes are available to users for problem determination and resolution. I hope some some of you have had a chance to look at these.

The following report was entered into the SAS Notes system by Technical Support documenting the problem.

```
V6-COPY-7021

Error: Out of memory when copying libraries between unlike devices

OPERATING SYSTEM: ALL                    RELEASE REPORTED:  6.08
        PRODUCT: BASE                      RELEASE FIXED:  6.08  TS410

   By default, PROC COPY, and the COPY statement in PROC DATASETS, attempts
   to exactly duplicate SAS data sets when copying them from one location
   to another.  One of the attributes COPY attempts to maintain is the page
   buffer size, or Data Set Page Size.  This attribute is established
   during data set creation, and it is an even multiple of the blocksize
   for the SAS library.  Problems occur when data sets are copied to a
   library that has a blocksize that is either not the same, or is not a
   multiple of, the input library blocksize.  In these instances, Data Set
   Page Size is increased to the next multiple of the output library's
   blocksize.  If this process is repeated, eventually the Data Set Page
   Size can become so large, it will have a substantial impact on memory
   and disk/tape space requirements.  This problem is most frequently
   encountered when libraries are moved between disk and tape.

   To problem can be circumvented by specifing the NOCLONE option on the
   COPY statement, beginning with Release 6.08.  Setting this option forces
   COPY to recalculate the Data Set Page Size, using the current setting of
   the system option BUFSIZE.  Note, if a specific Data Set Page Size is
   desired, specify the BUFSIZE= system option in conjunction with the
   NOCLONE option.
```

Output 2 *SAS Note V6-COPY-7021*

This problem will be referred to as V6-COPY-7021 hereafter.

Let's pause and review where we are in the maintenance process. Technical Support was able to collect enough information to reproduce the problem, the problem was reported to the development staff, a SAS Note was entered documenting the problem, and paramount to this user, Technical Support was able to provide a circumvention to the problem. But with this new maintenance process we can do better than circumventions. We can provide real solutions to complex problems you uncover in the SAS System.

Because the problem was found and reported does not necessarily mean it will be fixed in a maintenance release. Many questions must be answered before a fix is approved for maintenance, but we now have a vehicle for delivery of solutions to complex problems that could never have been fixed in the past.

Let's stop and think about what this would have meant in the past. Five years ago, when Version 6 was still in development status, and either Release 5.16 or 5.18 was the production release, the SAS System ran on six operating systems: MVS, CMS, VSE, VMS, AOS, and PRIMOS. To correct a problem on these six operating systems, six different developers, each writing code for one of the above platforms, would write a zap or patch for their respective systems. That's six different zaps to PROC COPY. This is a considerable use of resources, but one that was required for serious problems.

There were enough serious problems found in Release 5.18 that SAS Institute provided over 680 production zaps to support all products in the SAS System for these operating systems.

380 production zaps written for MVS
175 production zaps written for CMS
83 production zaps written for VMS
42 production zaps written for VSE

Figure 2 *Production ZAPS/PATCHES for Version 5*

If you consider the amount of time that Release 5.18 was supported by SAS Institute, this is not an extraordinary number of zaps. But it was difficult to tell a new customer who just finished installing a shiny new 5.18 release that they needed to apply 100 or more zaps to the new system to prevent them from running into known problems. The same can be said for Release 5.16, except the numbers were actually higher.

Of course, zaps could not be written to correct every problem. In certain situations they were impossible. In our bug, V6-COPY-7021, a zap to correct this would have been extremely difficult due to the nature of the changes that needed to be made.

With SAS Institute's commitment to support more operating systems in Version 6, it became apparent that we needed to devise a better plan to provide maintenance for our software.

Let us return to our PROC COPY problem, V6-COPY-7021. The first question that needed to be answered by the participants in the maintenance process was, "Should this problem be fixed with maintenance?" It was not clear to all parties involved that this was a problem that should be fixed via maintenance. Maybe it should be fixed with the next production release? That would mean Version 6.10 for some hosts and Version 7 for others. Translated into real time, over a year before all the users received the fix.

That may seem like an unacceptable amount of time to some, but there are a number of issues to consider, both external and internal, when determining the severity of a problem. This, in turn helps us decide if this problem should be fixed in maintenance. Externally, we must consider how many users are affected? Does the problem make the software hard to use? Does the problem compromise the integrity of the data? Are incorrect results being produced, and if so are the incorrect results obvious to the user? From an internal viewpoint there are two main issues: coding impact and testing impact. We must study both of these impacts to guarantee we do not risk achieving our goals in providing maintenance to you.

One of SAS Institute's goals in delivering maintenance for our software is timeliness. If too much time passes between maintenance releases, we run the risk of not satisfying or even losing our customers due to outstanding problems that aren't being fixed in a timely fashion. If too little time passes, we jeopardize our main goal in providing maintenance - quality.

Research is done before the decision is made to fix a problem on what components of the SAS System are impacted by making this fix. This research helps us answer the question of whether the problem should be fixed in the current maintenance track, the next

maintenance track, or the next production release. Because testing by host and product testers as well as our Quality Assurance department takes up a significant amount of the maintenance cycle, and what is tested is directly related to the components that will change, this research is paramount in maintaining the quality of maintenance and ensuring that it is delivered in a timely fashion.

The changes to the SAS System that will be made by correcting a problem may have widespread effects that are not always obvious. Let me provide you with an example. It was decided for our bug, V6-COPY-7021, that a new NOTE should be written to the SAS LOG when PROC COPY is executed and the new code added to correct V6-COPY-7021 is exercised. With TSLEVEL 410 of the SAS System, the following NOTE will be written to the SAS LOG when executing PROC COPY under certain situations:

```
NOTE: BUFSIZE is not cloned when copying across
dissimilar engines. System Option for BUFSIZE was used.
```

Figure 3 *Partial SAS LOG Displaying New NOTE with PROC COPY*

This appears to be a harmless change, but the testing impact is significant.

One of the ways our software is tested is by running groups of tests called suites and comparing the output from these suites to benchmarks, previous output we know to be correct. Much of this process is automated, and running a number of suites that do not produce differences can be done quickly. When differences appear though, the process is slowed dramatically as each difference must be investigated and resolved. Because PROC COPY is a very functional procedure, it is used in many of these suites. Writing the new NOTE to the SAS LOG would cause differences to show up in a number of suites. The research I mentioned earlier determined that approximately 200 new benchmarks would have to be made so that tests could be run without differences in Release 6.08. Those same 200 benchmarks would have to be made for Release 6.09 and Release 6.10. Thus, the appearance of a new NOTE to the SAS LOG impacted the testing of maintenance in ways that were not obvious to everyone involved. But, discovering this before the change was made allowed us to allocate additional resources so that the maintenance cycle will not be delayed.

After we determine the severity of the problem and the impact fixing this bug would have on the maintenance cycle, we decide whether or not to approve it. This decision making is shared among Technical Support, Quality Assurance, Development staff, and Host and Product Testers. For our bug, V6-COPY-7021, the severity of the problem was judged to be high enough to warrant inclusion in maintenance. The coding and testing impact was determined to be within an acceptable range, thus the decision was made to fix this problem in the next maintenance release. So, let's continue tracking the movement of the fix in the maintenance cycle.

To fix our problem, the algorithm PROC COPY used to determine BUFSIZE needed to be modified. This is not a trivial change, and as I mentioned earlier, it is a change that would have been virtually impossible to provide with a zap or patch. But, in our new maintenance process where changes are made to the source code, this type of fix is possible. In addition, our Multi-Vendor Architecture® (MVA®) where 90% of the source code is shared among operating systems, allows a portable developer to make the modification, and this modification is delivered to all operating systems later in the maintenance cycle.

Because our PROC COPY bug was portable, we were able to reap one of the benefits of MVA. But what happens if our bug doesn't fall in the portable code, but falls in the 10% of host-specific code? Those problems are handled differently, and I will discuss those a little later.

During the maintenance cycle, portable developers have a limited amount of time to make changes. The changes accepted into the maintenance release are for approved fixes only. The source code is regulated so that no unapproved changes are entered into a maintenance release. This allows us to keep track of what is fixed in each maintenance release and reduces the possibility of introducing new problems with maintenance.

Here are the number of portable fixes made for each maintenance release to date:

```
178 portable fixes to TS405
97 portable fixes to TS407
92 portable fixes to TS410
```

Figure 4 *Portable Fixes by TSLEVEL for Release 6.08*

TS405 contained a larger number of fixes than TS407 or TS410 because TS405 was the first maintenance for Release 6.08, and its cycle was twice as long as the cycle for TS407 or TS410.

Portable developers also make changes to our SCL-based products such as SAS/ASSIST®, SAS/EIS®, SAS/PH-Clinical®, and SAS/CPE®. These products, because they are SCL-based, are impossible to zap or patch with operating system utilities. Thus, New installation process typically required New installation documentation must be printed without our new process, it would be difficult to provide maintenance for these products on a regular basis. As the number of these SCL-based products grow, this process will establish itself as the best way to provide maintenance to all products in the SAS System.

DEVELOPMENT ENVIRONMENT

As the portable developers make changes to source code for maintenance, where are these changes stored? Let me attempt to explain. At SAS Institute, portable development is done on over 1000 Hewlett Packard (HP) workstations. Development is done on future releases of the SAS System, as well as maintenance for current releases. There exists a staff of programmers whose job is to develop and maintain our own in-house source management system, called SDS. SDS contains tools that are used to maintain the numerous levels or tracks of source code for past, future, and present releases of the SAS System. An entire paper could be written describing SDS at SAS Institute, so I will limit our discussion of SDS to how it relates to the maintenance process.

At any given time there are three levels of the SAS System maintained on the HP workstations specifically dealing with maintenance. Those levels are:

RELEASE	the current stable version for porting to other host systems
MAINT	the current stable development release that will be moved to the RELEASE level
ADD-ON	the level containing add-on products that will be shipped in experimental or beta status with the next maintenance release. This level gets ported to other hosts

Note that both the MAINT and RELEASE levels contain a subset

of the SAS System consisting of only the pieces of the software that have changed due to maintenance. This is due to the fact that only the changed pieces of the SAS System are shipped with maintenance. I will discuss this point in detail a little later.

All three of these levels of the SAS System can be executed or tested due to a look-through process maintained by SDS. This look-through process establishes a concatenation of data sets that allows each level to be executed as if it contained a complete SAS System.

Similar organizations of levels exist on all host systems. This organization allows for a consistent viewing of the source across all platforms and with the proper system command existing on each host, allows for a uniform way to execute each level of the SAS System on all hosts.

Portable developers make their changes and deliver them to the MAINT level. At the MAINT level, the portable developer will verify that their change corrects the reported problem and determine if the change adversely effects the entire SAS System. If problems arise when testing the change, the developer can easily back it out of the MAINT level. When the developer feels confident with the change, the code gets delivered to the RELEASE level. It is this level that will be ported to all other hosts. When the deadline for making portable changes is met, the RELEASE level is built and then subjected to a series of build verification tests that verify the basic functionality of the system. Any component that fails these tests must be fixed before the source is ported to other hosts.

After the build verification tests are run successfully, the RELEASE level is simultaneously ported to all other hosts. Each host receives the port from the HP workstations and builds this port into an executable image on each of their respective hosts. Each host runs build verification tests against its executable image. This ensures that the porting process completed normally and the same basic functionality exists on the hosts as it did on the HP workstations.

Now that each host has an executable image that contains all portable fixes, the issue of host-specific bugs can be addressed. As I mentioned earlier, not all bugs occur in the 90% of portable code that makes up the SAS System. Some bugs will be uncovered in the host-specific code. It's at this time, after an executable image is built for each operating system, that host developers will make changes to the source code that corrects problems specific to each operating system.

DIFFERENCING

A primary aspect in quality control of a maintenance release is something we call differencing. *Differencing* is the process of determining which pieces of the SAS System have changed since the previous maintenance release. As I mentioned earlier, since only the changed pieces of the software are shipped, we had to develop a process for validating which pieces have changed. If a piece of the software doesn't change, whether it is a load module, a SAS message file, or a SAS catalog entry, we do not ship it.

For this process each host has developed tools to determine what pieces of the software have changed. All file types are used as input to these tools: images, catalogs, message files, samples, autocalls, and SAS data sets. The tools generate output that is compared among the host groups to make sure a change to the software occurs across all operating systems. If an executable image has changed on one host, it should have changed on all others. All discrepancies must be resolved before testing of the maintenance release can officially begin. This is just another way to make sure the maintenance release we ship is of the highest quality.

TESTING BEGINS

After host fixes are made and any differencing issues have been resolved, the most time-consuming piece of the maintenance cycle begins: testing.

Two basic kinds of testing are performed for the maintenance process: fix verification tests and regression tests.

Fix Verification Tests

For fix verification testing, tests are performed to confirm that all approved fixes have been delivered into maintenance. To accomplish this, a program is written and executed for every approved fix, and for portable bugs, executed again on each operating system. New test verification jobs are written during every maintenance cycle to validate each fix. After these test programs are written and executed, they are cataloged and reused in testing of subsequent maintenance releases as well as testing of future production releases.

Regression Tests

The second type of test, regression testing, is testing that ensures no regressions have occurred in the maintenance release. These tests confirm that changes to correct problems do not introduce new problems. By design, the SAS System is dependent on itself. There are components that depend on other components. The regression testing ensures making changes in one component does not adversely affect another component's behavior.

The largest part of the maintenance cycle is allocated to testing because, if done properly, it will guarantee the quality of the maintenance release. There is a constant battle early in the maintenance cycle weighing the number and types of fixes approved against the impact each fix will have on testing. We must take into account the number of fixes because, as more fixes are approved, more pieces of the SAS System will change, which in turn increases the number of test suites that must be run. Running more test suites takes time but it is not only the numbers we must be aware of. For example, in our bug V6-COPY-7021, the source code for PROC COPY had to be modified, which led to one fix verification test and over 200 regression tests being run on each operating system. If another fix was approved for PROC COPY there is a very good chance that only a single fix verification test would have to be written and executed. No additional regression tests were likely to be run. The same thing is true if 10 more fixes were approved for PROC COPY. But if 10 more fixes were approved in other areas, thousands of additional regression tests may need to be executed. Thus, we must also be aware of the areas of the software that are affected when approving each fix for a maintenance release.

We would like to be able to fix all reported problems during a maintenance cycle but this is not a practical goal. Occasionally, problems are reported that could impact testing to an unacceptable degree. They may cause too many suites to be run if approved, or too many tests would have to be written to fully test the fix. In these cases, we either may either delay the approval of the fix until the next maintenance cycle, or if the problem is serious enough, allocate additional resources for testing to accommodate the increased workload.

Testing New features

Two other factors that significantly affect testing are new features

and add-on products.

New features can be defined as new functionality to an existing product in the SAS System. Before this new maintenance process was established it would have been extremely difficult and costly to send out new features. Generally, new features to existing products would have to wait until the next production release before being shipped, but with our new maintenance process this is not a problem. Some new features that have been shipped with TS405 and TS407 of Release 6.08 are:

PRODUCT	FEATURE
Base SAS®	SQL Query Window
SAS/CONNECT®	Remote Library Services
Base SAS	Console Window for VSE
Base SAS	Printing using JECL for VSE
Base SAS	Printing enhancements for OS/2® and Windows
SAS/IML®	Performance enhancements
SAS/GRAPH®	New Full Screen Drivers
SAS/SHARE®	Security enhancements for CMS
Business Object Library for SAS/EIS®	All 6.08 TS415 and 6.09 TS039 platforms
Image additions to SAS/GRAPH	All 6.08 TS415 and 6.09 TS039 platforms

Figure 5 *New Features Shipped with Maintenance*

These new features increase the impact on testing because of the additional tests that must be written and run, but the feedback received from users by Technical Support has been positive. Because of this feedback, we feel the advantages of sending these new features far outweigh the cost.

Testing Add-on Products

At SAS Institute, we are striving to stay abreast of the latest technology and, as the market demands, develop new products on various operating systems. When testing and shipping these new products, we have to ensure the production release is not adversely affected in any way. Needless delays or worse, bugs introduced to the SAS System are possible when delivering a new add-on product into the SAS System. With our new maintenance process, we believe we have a system that minimizes these possibilities.

In the past, if a new product was not stable at the time a production release was shipped, we had a few choices:

- Postpone the shipment of the SAS System until the new product was stable. This caused needless delays to all customers, some of whom didn't care about the new product.
- Wait until the next production release to ship the product. This could be much too long to expect the users to wait for new technology.
- After the product was stable, ship it separately as an add-on product. This was extremely costly to SAS Institute as special tapes had to be cut, new installation procedures needed to be developed, and install documentation had to be written which caused confusion among the customers installing the product and required Technical Support to support multiple install procedures. Also, testing became more complicated after the production release was shipped as new levels had to be created separate from the production release.

In the new maintenance process,-we have alleviated many of the problems that made add-on products so difficult by having the following:

- Separate add-on track maintained within SDS that allows for ease of testing of the add-on product.
- No special tapes must be cut as the add-on product is included in the next maintenance release.
- Installation procedure does not change as each host has developed its installation process to handle add-on products. This means Technical Support only supports one install process per host. This makes it much easier on the person performing the installation.
- In the new maintenance process, the installation documentation may change with each maintenance release, thus the add-on product is easy to integrate into the modified version of the installation instructions.

Nothing comes without costs. Delivery of add-on products with maintenance has impacted testing by adding a few days to the testing phase of the maintenance cycle. Also, altering, verifying, and reprinting the installation documentation for each host in every maintenance release has proven to be a time-consuming task. But, once again, SAS Institute has found the new maintenance process to be the optimum manner in which to deliver add-on products to the users.

As an example of the success we have had with delivery of add-on products with maintenance for various operating systems, refer to the following tables:

PRODUCT	PLATFORM
SAS/ACCESS® to ODBC™:	Windows
SAS/ACCESS to ORACLE®:	CMS, MVS
SAS/ACCESS PC File Format:	OS/2® and Windows
SAS/ACCESS to SYBASE:	Windows, OS/2 (beta)
SAS/CPE:	MVS
SAS/ENGLISH®:	CMS, MVS, OpenVMS VAX, Windows, OS/2

Figure 6 *Add-on Products for Release 6.08, TS405 & TS407*

PRODUCT	PLATFORM
SAS/ACCESS to INFORMIX:	HP-UX, Solaris 1.0 & 2.0, AIX for RS/6000
SAS/ACCESS to INGRES:	Solaris 2.0, AIX for RS/6000
SAS/ACCESS to IDMS:	Experimental on MVS
SAS/ACCESS to ORACLE:	Solaris 2.0
SAS/ACCESS to SYBASE:	Solaris 2.0
SAS/CPE:	re-shipping on HP-UX, AIX for RS/6000, Solaris 1.0 & 2.0
SAS/GIS:	HP-UX(Beta), Solaris 1.0 & 2.0, AIX for RS/6000
SAS/ENGLISH:	VSE, CMS, MVS, HP-UX, Solaris 1.0 & 2.0, ULTRIX, OpenVMS VAX, AIX for RS/6000
SAS/SHARE®:	OS/2, HP-UX, Solaris 1.0 & 2.0, AIX for RS/6000, ULTRIX
SAS/SPECTRAVIEW™:	HP-UX

Figure 7 *Add-on Products for Release 6.08 TS410 & Release 6.09 TS037*

PRODUCT	PLATFORM
SAS/TOOLKIT®:	OpenVMS AXP
SAS/SHARE®:	VSE, OpenVMS VAX, OpenVMS AXP

Figure 8 *Add-on Products for Release 6.08 TS415 & Release 6.09 TS039*

As a side note, all of our foreign language versions of the SAS System have shared similar problems, thus there is work ongoing to incorporate these foreign language versions into the maintenance process.

Testing of Installation Process

I will discuss the installation process and the various options available during installation a little later, but I wanted to point out that the installation procedure, as well as the installation documentation, is tested at this time.

We consider the installation procedure to be an important piece of our software, and occasionally we find problems with this procedure. These problems are treated like any other bug in our software, and as such, the fix for these problems must be approved and tested. Also, the installation of any add-on products that were delivered with maintenance must be tested. Each host group can test the basic functionality of their installation procedure at this time to ensure that a change has not adversely affected this process. These tests are typically performed using a disk version of the SAS System, and thus are not the final testing performed on the installation process. But detecting these kinds of problems early in the cycle will prevent delays later in the cycle when exhaustive testing of the installation process is performed.

INSTALLATION VERIFICATION

When host testers, product testers, and Quality Assurance have completed testing on a given operating system, that host can begin the process of moving the tested copy of the SAS System to media appropriate for shipping. The testing groups will not complete testing on all hosts simultaneously, thus a divergence in the maintenance cycle for each host is likely to occur at this point.

The first step in moving the tested copy of the SAS System to media where it becomes the next maintenance release involves our Software Production Systems, or SPS department. Each host delivers *submasters*, subsets of the SAS System broken down by product, to SPS. SPS receives the submasters and creates a SAS System on various media that is used for installation verification by each host group.

Installation verification media is used for the following reasons:

- Each host group can fully test its installation process using each type of media that will be shipped. This should uncover any problems in the SPS media-generating procedure as well as any problems in the installation process that may be dependent on media type.
- After each medium is installed to disk, a compare is performed on the newly installed version to the old tested version used to generate the submasters. This ensures that the newly installed version is a carbon copy of the tested version.
- Each newly installed version is subjected to a high-level set of regression tests to ensure its basic functionality.

TRIAL PACKAGE

Once the installation verification is completed, the last step of our maintenance cycle gets underway, trial packages. A *trial package* can be defined as the output from the maintenance process. A trial package includes all enclosures, installation instructions, and install media, received by a customer. A trial package is delivered to each host group, the Quality Assurance division, and the Technical Support division. The host group and Quality Assurance check the package for correct contents while Technical Support takes the trial package testing a step further.

Technical Support performs a complete installation using the media in the trial package. They typically receive more than one trial package so they can test different types of media, each containing a different product mix. Technical Support will use this trial package to diligently test the installation process. Since Technical Support will eventually talk with the customers installing the maintenance, they want to guarantee there are no late developing problems that will generate a rash of calls. After Technical Support completes the trial package testing, the maintenance can begin shipping.

INSTALLATION OF MAINTENANCE RELEASE

Let's suppose you have experienced the V6-COPY-7021, PROC COPY bug, and receive a maintenance package. What should you do with it? What are your options? The first steps are to determine what type of maintenance package you have received and what type of installation you should perform.

Three types of maintenance packages:

Maintenance only	package contains maintenance for products currently licensed
Maintenance with add-on product	package contains maintenance for products currently licensed, along with at least one new (add-on) product
New customer	new customer, never installed the SAS System

NOTE: In the context of this discussion, an add-on product will be defined as a product in the SAS System you are not currently licensed for. This differs from my earlier discussion of an add-on product being defined as a new product developed for the SAS System.

Two types of installation

Direct installation	maintenance overlays your current SAS System
Staged installation	maintenance installed in "staging" area, thereby not affecting your current SAS System. This option should be exercised if no interruption in your production environment can be tolerated

Maintenance packages are customized based on the products each customer licenses. Thus, the installation media you receive contains only the maintenance for the products you currently license. This differs from media for earlier Version 6 releases where maintenance for all products was included and it was the installer's responsibility to determine what maintenance to select from the installation media.

With customized media, SAS Institute was able to accomplish the following:

- Simplify the installation process. Eliminate the need for installer to determine what products they currently license.
- Reduce the amount of installation media we were required to ship. Including maintenance for all products on the installation media takes more space. As more operating systems are included in the maintenance process and the cost of media continues to rise, this seemed like the logical choice.
- Reduce the risk of running products at different maintenance levels. Customized media in conjunction with the new installation process reduces the risk of sites executing the SAS System with this unsupported configuration.

If you decide to license a new product, the package you receive will contain maintenance for all the products you previously

licensed, along with the add-on product at the same maintenance level. Your add-on product will automatically install at the same maintenance level as the other products you currently have licensed. The installation process installs the add-on product in this manner because running products in the SAS System at different maintenance levels is not supported. As a further note, sites licensing SAS/CONNECT software should verify that both the local and the remote sessions are executing the same maintenance release.

The installation type you choose is up to you, but once again, the Staged installation is suggested over the Direct installation if no interruption in your production environment can be tolerated.

New customers receiving the SAS System for the first time will find, after installation, a SAS System running at the latest maintenance level.

IS MAINTENANCE WORKING?

If the number of maintenance packages our Distribution department has shipped is any indication of the success of this process, then we have been extremely successful. The following statistics were gathered from the distribution log at SAS Institute at the beginning of March, 1994.

6.08 TSLEVEL 405 MAINTENANCE PACKAGES SHIPPED

OPERATING SYSTEM	MVS	CMS	OS/2	VMS	VSE	Windows	TOTAL
TS405	580	156	314	230	42	851	2173

6.08 TSLEVEL 407 MAINTENANCE PACKAGES SHIPPED

OPERATING SYSTEM	MVS	OS/2	VMS	VSE	Windows	TOTAL
TS407	849	835	269	48	2619	4620

6.08 TSLEVEL 410 MAINTENANCE PACKAGES SHIPPED

OPERATING SYSTEM	MVS	CMS	VMS	VSE	TOTAL
TS410	849	183	236	57	1325

6.09 TSLEVEL 037 MAINTENANCE PACKAGES SHIPPED

OPERATING SYSTEM	HPUX	SUN4	OLVH	AIXR	TOTAL
TS410	93	236	8	74	411

Display 1 *Maintenance Packages Shipped*

Since the maintenance packages are sent by customer request only, the numbers above indicate the resources devoted to the maintenance process and the time spent producing a maintenance release by SAS Institute are well worth it.

CONCLUSION

If you are currently running, or considering installing Release 6.08 or Release 6.09 of the SAS System, SAS Institute strongly recommends you request and install the latest maintenance release.

The installation process will not take long and is designed so that the maintenance release can be installed and tested in a way that will not interfere with any production work you have ongoing.

Each maintenance release is cumulative - so even if you missed one of them, there is no excuse not to install the latest.

Why discover problems other customers have already discovered? A lot of blood, sweat, and tears go into each maintenance release to guarantee its quality. You should take advantage of this service provided by SAS Institute to ensure your SAS System is of the highest quality available.

SAS, SAS/ACCESS, SAS/ASSIST, SAS/CONNECT, SAS/CPE, SAS/EIS, SAS/ENGLISH, SAS/GIS, SAS/GRAPH, SAS/IML, SAS/ODBC, SAS/PH-Clinical, SAS/SPECTRAVIEW, and Multi-Vendor Architecture are trademarks or registered trademarks of SAS Institute Inc. in the USA and other countries.

OS/2 is a trademark or registered trademark of International Business Machines Corporation.