ABSTRACT

When you install the SAS® System under OS, you must set many system options and choose several installation options. This paper discusses the implications of these choices and suggests some tuning techniques that could be used at your site. The paper also addresses some common problems encountered by installation representatives over the past year.

INTRODUCTION

In the Technical Support Department at SAS Institute, we consistently hear comments on the complexity of installing the SAS system and we receive requests for guidance in selecting installation parameters and SAS system options. The objectives of this presentation are to clarify the general process of installing the SAS system and to provide some suggested option settings that should improve the overall performance of your SAS programs.

The SAS System installation consists of two phases. The purpose and timing of these two pieces has caused some confusion. The first phase (PART I) completes the majority of the installation for all of the products that you have ordered. Phase two consists of a set of steps (PART II) that are unique and separate for each product. For example in Part I you run a job called INSTALL. This job installs from tape the load modules for all of the products on the installation tape you received. Because the load library is common to all products, it is completely loaded in one job. Specifically, if you ordered the base SAS software, SAS/FSP® and SAS/GRAPH® the INSTALL job downloads all three of these products at once. You will not see another install step in the PART II description. In PART II for one product you may only find a test job, yet for another product you might find a library that must be installed that is unique to that product.

The next aspect of the SAS System installation that has confused many installers is the SASEDIT step, during which you edit two members of SAS.CNTL and then submit the job called SASEDIT. The purpose of this step is to reduce the number of edits required before you submit each of the various installation jobs. After describing this process and attempting to clarify other common bottlenecks in the OS installation process, I will explain how to modify these steps when you install an "Add-on" product to a previously installed SAS system.

During the installation there are three major points where you will make decisions that affect performance. First, in the SASEDIT or JCL tailoring phase, you must decide whether you want separate load libraries for each product or just one. You can also choose block size, primary space quantities, and other attributes for any library that will be installed. These decisions can have performance impact. Second, in the INIT step, you must select the default SAS system options. Finally, you must build the SASLP module, which may be the single most important step toward improving SAS system performance at your site.

SUMMARIZED INSTALLATION PROCESS

PART I: STEPS THAT AFFECT ALL PRODUCTS

Part I completes all steps that will be necessary for all products and generates the JCL for use in all parts of the installation. After completing Part I, you should have allocated all desired libraries, loaded the SAS load library, set the SAS system default options, loaded the UTILITY library, loaded the AUTOCALL MACRO library (optional), loaded the HELP library (optional), and loaded the SAMPLE library (optional). In Step 5 you are directed to install the SAS SVC Routine 0. Do this before completing Part I.

Creating the SAS.CNTL Data Set and Tailoring the JCL

IEBGENER: Enter the IEBGENER job which is listed on cover letter. Submit this job to download the SASUPDTE job from the tape to any data set of your choice.

SASUPDTE: Submit the SASUPDTE job to allocate the SAS CNTL data set, load the first set of members from the tape, and build the member SASEDITP. At this point the PDS will contain several members including SASEDITO, BEDITP, SASEDITP, and SASEDIT.

The member SASEDITO contains the object form of a very rudimentary macro-type language. Once link-edited the program will assign values to symbolic variables defined in the input data set:

```
PREFIX=userid.SAS
LOAD=LOADDSN+PREFIX..LIBRARY
LOADBLK=6144
```

search for symbolic references of the form %variable in another input data set:
```
//SLOAD: DD DSN=LOADDSN,
//SLOAD: DCB=(DSORG=PO,RECFM=U,BLKSIZE=LOADBLK),
```

and substitute defined parameters in place of any symbolic reference:
```
//DD DSN=userid.SAS.LIBRARY,
//DCB=(DSORG=PO,RECFM=U,BLKSIZE=6144)
```

Note that if LOAD=* then the DD statements would become comments.

The member BEDITP contains a list of installation parameters that are used while you are installing all products as well as those specific to base SAS software. For example, in BEDITP you can specify the JOB statement to be used for all the subsequent jobs or the prefix to be used on all of the newly created data sets and libraries (%PREFIX above). This member also allows you to specify the names of certain optionally allocated libraries such as the SAS HELP library or the AUTOCALL MACRO library.

The member SASEDITP contains the installation parameters for each product. For example, you specify the name of your SAS load library, its blocksize, and location only in this member and these specifications are automatically used throughout the installation. You will also notice several members in SAS.CNTL with the suffix EDTP, such as EEDITP and FEDITP. Each of these is associated with one of the products that you are installing. You will never have to inspect or edit these members. The SASUPDTE job downloaded each member from separate files on the tape and automatically concatenated them into the member SASEDITP.

Finally you will find a set of members in SAS.CNTL with names in the form: xCNTL. These contain the skeleton JCL for each of the installation jobs. You DO NOT have to edit or even inspect these members.
SASEEDIT: Once you have selected all of the options provided in the BEDITP and SASEDITP members, you will run the job in the member SASEEDIT. This job will:
1. link edit the code in the member SASEDITO, 
2. read the parameters specified in BEDITP and SASEDITP, 
3. read in the xCNTL skeletons, 
4. perform symbolic substitutions, and 
5. output the JCL required for all remaining installation steps.

After running the SASEEDIT job, the SAS.CNTL PDS will contain several new members, including xINIT, xHELP, xALLOC, xINST TALL, xUTIL, xMAUTOS, xSAMPLE, etc., where x represents a SAS product initial. While completing PART I of the install, you should never need to submit or inspect any of these members except the one where x is blank. For example, BALLOC, FALLOC and GALLOC were all tailored and concatenated into the member ALLOC. The individual product members were only needed by the SASEEDIT job.

Installing the SAS Load Library

Once you have completed the SASEEDIT step, you can submit the job INSTALL, and all load libraries will be loaded for all products according to the parameters you specified in BEDITP and SASEDITP.

Picking the SAS System Options

Deviating slightly from the methods used so far, to pick the SAS system options, you must edit a set of members specific to each product. These members are of the form:

- xINITP - if the options do not vary between TSO and batch
- xINITPT - for options to be used in the TSO environment
- xINITPB - for options to be used in the batch environment

where x is the product initial, as defined below:

B = Base SAS Software
E = SAS/ETS®
G = SAS/GRAPH®
F = SAS/FSP®
A = SAS/AF®
O = SAS/OR®
L = SAS/IML®
I = SAS/IMS- DLI®
R = SAS/REPLAY-CICS®
Q = SAS/QC®
D = SAS/DMP®
U = SAS Usage Notes
S = SAS SVC Routine 0
H = SAS/SHEAR®
Z = SAS/DF2®
C01, C02, C03, C04, C05, and C06 = SAS CBT Courseware.

These initials are subject to change in future releases and represent the values used for the OS SAS Release 5.16.

INIT: After editing these members, submit the INIT job to set all options. To speed up your installation, initially you can simply submit INIT accepting the defaults as sent by SAS Institute. Or, you can edit each using the guidelines discussed below. You can change the option defaults at any time in the future by simply rerunning the INIT job as many times as you like.

Load the Supporting Libraries

The remainder of PART I involves simply submitting a series of required and optional jobs to load various supporting libraries such as the SAMPLE library or UTILITY library. Very few people have experienced difficulty with this section.

PART II: FINAL ADJUSTMENTS TO EACH PRODUCT

While the steps in Part I affected all products simultaneously, the steps in Part II, are unique and only affect that product. In fact each product will have its own Part II manual. Simply follow the instructions and submit the required programs. These jobs were also created by the SASEEDIT step in Part I and should require only that you submit the appropriate member, unless otherwise noted.

Base SAS Software is considered a product also. You must complete the Part II steps for the base product before you begin the Part II steps for other products. These steps will create the module SASLPA and install the SAS catalogued procedures and CLIST.

ADD-ON PRODUCTS

When you order an additional product from SAS Institute, you will receive essentially the same installation instructions as you did for your original tape. You must again complete Part I, because you will again have to tailor your JCL for the new product and again run the INSTALL job to download the new modules. The new tape contains only the modules required for your new product. Therefore, you are not reinstalling the entire system. The most common installation mistake with an Add-On product is to start with the Part II manuals. This does not work.

You have the option of reloading your previous SAS.CNTL library with the new installation members. This works, but the common members, such as INSTALL, ALLOC, and INIT will be overwritten, which may not be desirable if you ever need to rerun these original jobs. We recommend that you create a second SAS.CNTL library just for the additional product. You can then copy over the BEDITP member from the first SAS.CNTL library. BEDITP will automatically pass in the required parameters such as the name of your SAS System load library, the HELP library and the UTILITY library. Note that only one SAS HELP library and one UTILITY library are permitted. Concatenations will not work. SASEDITP for an "Add-On" tape includes only the parameters needed for the new products.

PERFORMANCE CONSIDERATIONS AT INSTALLATION TIME

You must make decisions that can affect performance at three points during the install. You have options during the SASEEDIT step, the INIT step, and the BLPALIB step.

SASEEDIT OPTIONS

MULTIPLE LOAD LIBRARIES: In the member SASEDITP, you can choose to load each product into separate libraries or all into one. To create separate load libraries, you must set the xLOAD = option to null and be sure that the xLOADDSN = parameter specifies a name other than %LOADDSN. The default is to load all modules into the same library, which is specified by the parameter %LOADDSN.
Depending upon the nature of the applications at your site, you can reduce contention by placing different products in different load libraries on different packs. If, for example, most of your users use only the base SAS software, placing the base SAS modules in one library and including only that library in your STEPLIB concatenation certainly decreases the library search time.

The benefits of these manipulations depend heavily on your time.

were installed in separate load libraries on the same IBM 3380. The second used five load libraries all on the same IBM 3380. The third used five load libraries, but four of the products were installed in separate load libraries on the same IBM 3380, and the remainder were loaded into one library on a different IBM 3380. I ran the same job five times on each system. The job executed almost all procedures available in the test system. This intentionally induces a high program FETCH activity. The results of the test runs are listed in Table 1. Using Tukey’s test on the mean comparison, all EXCP means were significantly different. For CPU time, only the single library showed significant improvement. Execution times were more variable and only the five libraries on one disk pack were significantly different and slower.

Table 1 Mean comparisons for multiple versus single load libraries.

<table>
<thead>
<tr>
<th>Library</th>
<th>CPU seconds</th>
<th>Execution time</th>
<th>Total EXCP count</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIB 1 PACK 1</td>
<td>51.89</td>
<td>214.78</td>
<td>14490.6</td>
</tr>
<tr>
<td>LIB 5 PACK 1</td>
<td>53.13</td>
<td>281.18</td>
<td>16292.0</td>
</tr>
<tr>
<td>LIB 5 PACK 2</td>
<td>53.21</td>
<td>213.12</td>
<td>16287.2</td>
</tr>
</tbody>
</table>

Although this was certainly not an exhaustive test, the results suggest that placing all products in the same library is recommended. If you need to separate your load libraries, put them on separate disk packs. These changes can be made in one step while editing the SASEDITP member of SAS.CNTL.

LOAD LIBRARY BLOCK SIZES: In MVS/XA you can now block your load libraries larger than 18K. Unfortunately, we use a bootstrap version of the PDSCOPY procedure in our INSTALL job that does not accept the MAXBLOCK option we now offer in Version 5 PROC PDSCOPY. In other words, even if you allocate your load library with blocksize 19069 or 32760, the INSTALL job does not produce a module blocked larger than 18K. Using the larger block sizes will also reduce the number of tracks required to store the SAS load library. The maximum block size for IBM 3350 disks is 19069 and the maximum for IBM 3380 disks is 32760. To optimize space utilization against EXCP count, use a block size of 23476 for IBM 3380 disks where the maximum record length is 47K.

OTHER SASEDIT OPTIONS: Along with block size and library name, you can also specify the volume on which your library should reside and the primary and secondary allocation of space. All data libraries are allocated by the ALLOC job, which specifies space in terms of blocks instead of cylinders or tracks. This was done to ensure device independence. The load libraries are allocated with space requests in terms of 6144-byte blocks. The option of allocating in CYLS or TRKS is not offered within the SASEDIT step. If you want to allocate in cylinders or tracks, you must manually edit the ALLOC job after calculating the number of cylinders or tracks required. You do this, you must also alter the primary and secondary space quantities. In SASEDITP the parameters for the SAS load library are LOADQTY and LOADQTY. With Version 5.16 of the SAS System, we provided an extra document titled, "Space Requirements for the OS SAS System Release 5.16." This should help you determine how much space to allocate.

SAS SYSTEM OPTIONS

Many of the SAS system options have little or no performance impact. Others may have an impact but can only be set according to the situation or will allow the use of facilities that may ultimately affect your systems performance. Still others can directly and significantly affect the CPU time, memory requirements, EXCP count or execution time of your SAS program. Below, I categorize each of the SAS system options and discuss those that have an impact. The only options mentioned are those that appear in the installation member BINITPB. Refer to this section when editing this member.

Low impact options:
The following list contains those system options which will have no significant performance impact or that must be set by the user and not set as a default for the SAS System. For these options we recommend that you use option settings as shipped by SAS Institute.
TIO, PRINT1, RELEASE1, SASBLDL, SASBOSSL, SASBRO, WORK~
TRANTAB~

Algorithm to keep only the most frequently accessed modules. A resource usage and will not be discussed in the presentation. not be covered in this discussion.

Several of the options below were discussed at SUGI 11 (Squillace 1986). An * in front of the option name below indicates that this option is also mentioned in this paper. Macro options will be discussed separately, while sort and graph-related options will not be covered in this discussion.

**BLDLTABLE**: This option specifies whether to activate the dynamiC BLDL table facility. It should reduce search and load times for SAS modules by maintaining a list of module names and addresses for those modules that have been loaded during a SAS step. The table size is 4K, so it provides enough space for about 63 entries. The BLDL table is initialized with the modules listed below. During the job, the table is updated using an LRU algorithm to keep only the most frequently accessed modules. A maintenance zap currently exists for the BLDLTABLE option in Version 5.16. The number is Z5162893, and it is a recommended zap.

Base SAS software modules:

APPEND, BROWSE1, CALENDAR, CHART, CONTENTS, CONTENT2, COPY1, COPY2, CORR, DATASETS, DELETE1, EDITOR, FORMAT1, FREQ1, GLM, MATRIX, MEANS, MOVE1, MX2, OPTIONS, PDSCOPY, PDSCOPY2, PCS1, PLT1, PRINTTO, PRINT1, RELEASE1, SASBLDL, SASBOSSL, SASBRO, SASDFLT, SASEDATE, SASEDATI, SASETIME, SASETIMI, SASDATE, SASFEXP, SASMOD, SASFN, SASFNM, SASGLOBL, SASLIBR, SASHELP, SASINT, SASMACC, SASMACS, SASSORT2, SASSORT, SORT, SUMMARY1, TABULATE, TRANSP, UNIVARIA

SAS/FSP software modules:

FSBROWSE, FSEDIT, FSEDIT2, FSEDIT3, FSLetter, FSLETT2, FSLIST, FSPRINT, FSPRINT2

A small amount of initialization time is dedicated to loading the initial BLDL TABLE list. This table is initialized whether you specify the option or not. The BLDLTABLE option only determines if the table is updated throughout your job or session. Interactively this option is extremely valuable.

**BLKSIZE**: This option specifies the default SAS data set block size. The recommended value and default as sent by SAS Institute is 23476 for IBM 3380 disk drives and 19069 for IBM 3350 disk drives. Because a SAS library uses a DSORG = DA, the block size for the library is always set to the maximum for that device. The block size of the SAS data sets within the library can only be determined by using PROC CONTENTS on each data set. Also be sure to read the description of the BLKSIZE option in the "SAS User's Guide: Basics, Version 5 Edition." The method SAS Software uses to determine block size for each data set is explained in Chapter 17, "SAS Files", in the "SAS DATA SET OPTIONS:"

Supporting Squillace (1986), my tests show that a larger block size will generally improve performance of I/O on SAS data sets just as it would on any other OS data set. To illustrate this, I duplicated the tests that Merrill (1984) describes in his book, "Merrill's Expanded Guide to Computer Performance Evaluation Using the SAS System." In Chapter 42, pages 840-854, Merrill illustrates how I/O block size and buffering reduce CPU consumption. In his experiment he read flat 80-byte records with an assembler routine and recorded CPU time while changing block size and the DD statement parameter BUFNO. I completed the same experiment using SAS data sets.

I created several data sets, each with 69,000 observations and 80-byte records. The "read" program was:

```
data _null_; set fileref.x; run;
```

The results are shown in Table 3. BUFNO has no appreciable impact, so the data below only display the results for BUFNO=1. SAS I/O routines are written using the EXCP access method with our own channel programs and therefore, do not use the BUFNO parameter.

**Table 3 Performance effect of SAS Data Set Block Size.** Numbers below represent a mean of 5 observations per category.

<table>
<thead>
<tr>
<th>BLOCKSIZE</th>
<th>CPU seconds</th>
<th>EXCP count</th>
<th>EXEC seconds</th>
<th>AVERAGE working set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024</td>
<td>6.17</td>
<td>5924</td>
<td>103.02</td>
<td>393.58</td>
</tr>
<tr>
<td>1684</td>
<td>4.46</td>
<td>3460</td>
<td>63.46</td>
<td>400.96</td>
</tr>
<tr>
<td>2484</td>
<td>3.84</td>
<td>2400</td>
<td>43.88</td>
<td>406.96</td>
</tr>
<tr>
<td>4804</td>
<td>3.19</td>
<td>1324</td>
<td>25.54</td>
<td>416.66</td>
</tr>
<tr>
<td>6324</td>
<td>2.93</td>
<td>1048</td>
<td>20.41</td>
<td>415.42</td>
</tr>
<tr>
<td>11444</td>
<td>2.70</td>
<td>657</td>
<td>13.64</td>
<td>428.06</td>
</tr>
<tr>
<td>23476</td>
<td>2.51</td>
<td>410</td>
<td>9.55</td>
<td>441.98</td>
</tr>
<tr>
<td>32724</td>
<td>2.47</td>
<td>343</td>
<td>9.10</td>
<td>444.03</td>
</tr>
</tbody>
</table>

**CHKPT**: Useful only in debugging, this option checkpoints the status of certain SAS options at the end of every SAS DATA or PROC step and will degrade performance. Set NOCHKPT as the default option.
**CMDMAC**: This is a new option that is documented in SAS Technical Report P-148, "Changes and Enhancements to the Version 5 SAS System." CMDMAC indicates that command-style macros can be invoked from DMS command line without a prefixing percent sign. When in effect, the SAS System must search the autocall libraries for a possible command-style macro each time a user issues a command. We recommend that you set NOCMDMAC as the system default and turn this on only at the specific point in your SAS session where you need it. Then turn it off again. The comments below on the option MEXTMOD also apply to CMDMAC.

**DMS**: The DMS option is ignored in batch. In BINITPB therefore, it is not an important option and is set to NOMDS. The SAS System is faster and requires less memory with NODMS, but your interactive productivity is better with DMS. Hence we recommend DMS for interactive system default.

**DMSXPGS**: Also a new option in Version 5.16, DMSXPGS is documented in the online HELP. Enter HELP DMSXPGS on SAS Display Manager command line. DMSXPGS specifies the number of extra 4K pages of memory wanted and enables you to run larger programs under DMS (default=2). A large DMSXPGS value may reduce memory fragmentation, if you are developing large programs under DMS; otherwise the value should remain small. DMSXPGS is obviously unimportant in batch and therefore in BINITPB.

**FILEBLKSIZE**: Affecting external non-SAS data sets, the FILEBLKSIZE option will follow the same rules as the BLKSIZE option. It is usually unimportant, however, because the block size of external files are usually predetermined or specified in the JCL. In general if you use a larger FILEBLKSIZE when running PROC SOURCE and you do not specify BLKSIZE in the JCL, the procedure tends to take slightly more memory, but less CPU time.

**GEN**: The number specified by this option determines the number of generations of history data (that is, the SAS source code used to create the data set) that a SAS data library should save. This data are stored in the SAS directory and therefore increase the amount of space required to store the data as well as increase I/O when you are reading and writing SAS data sets. The preferred setting from a performance standpoint is GEN=0, but your site may want to use the default of GEN=2 to serve the needs of most people.

**LEAVE**: This option is used to specify the amount of memory to "leave" unallocated by the SAS system. LEAVE = 0 is the recommended setting. Under normal circumstances a nonzero LEAVE is not necessary. Specifying LEAVE greater than 0 can impose some overhead on your job.

**MEMFILL**: DO NOT turn on MEMFILL unless specified to do so by Technical Support at SAS Institute. This option is used for debugging and initializes memory with the value specified in the FILMEM= option. The overhead induced by this process is significant. In an example job that normally took 53 CPU seconds, the CPU time jumped to 5 minutes and 12 seconds when NOMEMFILL was changed to MEMFILL. In the same job the execution time went from 2:53 to 8:27, and the page seconds from 7115 to 89,380.

**PROC SIZE**: This option can have a performance impact under certain situations, but should normally be set to PROC SIZE=MAX. This sets the maximum number of bytes a procedure can allocate in a single request to the SAS procedure interface routine and has no effect in TSO.

**SPOOL**: The SAS System saves SAS statements in the WORK data set for later use with the %INCLUDE statement, if the SPOOL option is on. You should set NOSPPOOL as the default unless you intend to %INCLUDE lines that you entered earlier in the job. NOSPPOOL is the default as shipped by SAS Institute.

**MACRO Language Options**

**MACRO**: This option determines whether the macro facility is available or not. Turning the option on (MACRO) or off (NOMACRO) costs very little in overhead to the system. If you use the facility, the impact will depend on the nature of the programs you write.

**MAUTOSOURCE, IMPLMAC and MEXTMOD**: The new autocall macro facility is enabled with the option MAUTOSOURCE. This option by itself costs very little, but in conjunction with the IMPLMAC and MEXTMOD options, can be deadly. MAUTOSOURCE tells the macro processor to search for macros in a library specified by the SASAUTOS option. The IMPLMAC option allows you to call macros without precedeing the macro name with a percent sign. Macros called this way are known as statement-style macros. If you use IMPLMAC, the macro processor examines the first word of every statement to see if that word is a statement-style macro call. If MAUTOSOURCE is on, then the processor will also search the autocall library at the beginning of each statement. This can obviously slow down your program. If you have applications that use statement-style macros, you may want to turn IMPLMAC on just before that section of your application, and then turn it off afterward. Always set NOMIMPLMAC as the SAS System default in the member BINITPB.

In addition, the undocumented option MEXTMOD causes the macro processor to search not only the autocall library for macros, but also all load libraries in the STEPLIB concatenation as well as in the link list. Using MEXTMOD can effectively double your execution or response times and will roughly increase your EXCP count by 20% and your CPU time by about 5%.

**MCOMPILE**: If you are sure you will be using the macro language, you should turn this option on. The option only dictates whether the macro compiler will be loaded by default or on demand. You should set your system default to NOMCOMPILE. If you have installed SASLPA in the Link Pack Area as recommended, and included SASMACL in SASLPA, the macro compiler is already in memory and the option has no performance implications.

**MLEAVE**: This option specifies the amount of memory to leave for the macro language processor from the amount specified in the MSIZE= option. The default as shipped by SAS Institute is MLEVEx=6K, however, Squillace (1986) recommended MLEVEx=64K with MSIZE=128K.

**MRECALL**: The option instructs the macro processor to search the autocall macro library for a name that was not found in an earlier call. We recommend NOMRECALL to reduce the number of searches. You can specify this option at invocation or during the job, so there is no advantage to setting the default to anything but NOMRECALL.

**MSIZE**: You can specify the amount of memory available to the macro processor with the MSIZE option. The default as shipped by SAS Institute is MSIZE=12K, however, Squillace (1986) recommends MSIZE=128K.

**MWORK and MSYSIZE**: These options specifies the size of the work area available to the macro processor and the initial size of the macro symbol table. See the section entitled Usage Notes for Memory Management in Chapter 19, "SAS Macro Language" of the "SAS User's Guide: Basics, Version 5 Edition" for recommendations on the use of these options. The values as shipped by SAS Institute are MWORK=2K and MSYSIZE=1K.

**SASAUTOS** and **MERROR** impose no significant performance impact.
Other Options

The options that affect the behavior of the SORT procedure and those relating to SAS/GRAPH will not be discussed in this paper. The options for the SORT procedure that appear in BINITPB are the following:

DYNAALLOC, FILESZ, SORT, SORTDEV, SORTLIB, SORTLIST, SORMTMP, SORTPGM, SORTSIZE, SORTWKDD, SORTWKNO, SYNCSORT®

The options for SAS/GRAPH that appear in BINITPB are the following:

GRAPH, GACCESS, GCLASS, GCOPIES, GDEST, GFORMS, GPRTCL, GWAIT, GWRITER

USING A NON-OVERLAID VERSION OF THE SAS SUPERVISOR

The advantages of using the SASLPA module from the Link Pack Area have increased with each release of the SAS System. The size of the module increased, thereby directly affecting your memory usage. The importance of the CPU time savings has also increased with Version 5.

SASLPA Size: Release 82.4 versus 5.08 versus 5.16

If your site supports a large SAS user population, the memory savings of placing SASLPA in the Link Pack Area is obvious by just noting the size of the module. In SAS Release 5.16 each user will require 575K less if they do not have to load the supervisor for every SAS session or job. You should think twice about leaving SASLPA in the STEPLIB because each job or session will immediately need 575K before it can do anything. The memory differences are apparent in the SAS log if you use the option MEMRPT.

<table>
<thead>
<tr>
<th>Module</th>
<th>Release</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>SASLPA</td>
<td>82.4</td>
<td>249K</td>
</tr>
<tr>
<td>SASLPA</td>
<td>5.08</td>
<td>429K</td>
</tr>
<tr>
<td>SASLPA</td>
<td>5.16</td>
<td>575K</td>
</tr>
<tr>
<td>SAS</td>
<td>82.4</td>
<td>44K</td>
</tr>
<tr>
<td>SAS</td>
<td>5.08</td>
<td>50K</td>
</tr>
<tr>
<td>SAS</td>
<td>5.16</td>
<td>88K</td>
</tr>
</tbody>
</table>

Other performance measures on SASLPA

To measure the actual effect of using the SASLPA module, I ran three sets of test programs. Program A only initialized the SAS System, then ended. This should establish the base line cost or savings incurred by just executing the SAS System. Program B executed each procedure in the test system that I installed. This test should induce high load activity involving CPU time, and EXCP count. Program C created a very large data set, but only executed one DATA STEP. The EXCP count and CPU times should be high, but should not show extreme variation when SASLPA comparisons are made. The results are shown in Table 4.

<table>
<thead>
<tr>
<th>TEST</th>
<th>LOCATION</th>
<th>ENTRY POINT</th>
<th>TOTAL</th>
<th>EXCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST</td>
<td>LOCATION</td>
<td>ENTRY POINT</td>
<td>TOTAL</td>
<td>WORKING</td>
</tr>
<tr>
<td>TEST</td>
<td>LOCATION</td>
<td>ENTRY POINT</td>
<td>TOTAL</td>
<td>CPU</td>
</tr>
</tbody>
</table>

SUMMARY

During the installation of the OS SAS System, there are three places where you can make decisions that can either improve or degrade SAS System performance. Many decisions can be made during the SASEDIT phase that minimize the number of steps required to fine tune your system. Correctly specifying the SAS System options during the INIT job is extremely important. Finally, install the SASLPA module in the Link Pack Area if at all possible if you have a large SAS user community.

REFERENCES


Table 4 Mean comparisons using different ENTRY points in the SAS System

<table>
<thead>
<tr>
<th>TEST</th>
<th>LOCATION</th>
<th>ENTRY POINT</th>
<th>TOTAL</th>
<th>EXCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>STEPLIB</td>
<td>SAS</td>
<td>280</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>STEPLIB</td>
<td>SASLPA</td>
<td>278</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>STEPLIB</td>
<td>SAS</td>
<td>146</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>STEPLIB</td>
<td>SASLPA</td>
<td>13956</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>STEPLIB</td>
<td>SAS</td>
<td>6354</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>STEPLIB</td>
<td>SASLPA</td>
<td>6216</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>STEPLIB</td>
<td>SASLPA</td>
<td>6085</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TEST</th>
<th>LOCATION</th>
<th>ENTRY POINT</th>
<th>TOTAL</th>
<th>WORKING</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>STEPLIB</td>
<td>SAS</td>
<td>443</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>STEPLIB</td>
<td>SASLPA</td>
<td>875</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>STEPLIB</td>
<td>SAS</td>
<td>521</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>STEPLIB</td>
<td>SASLPA</td>
<td>1010</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>STEPLIB</td>
<td>SAS</td>
<td>633</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>STEPLIB</td>
<td>SASLPA</td>
<td>517</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TEST</th>
<th>LOCATION</th>
<th>ENTRY POINT</th>
<th>TOTAL</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>STEPLIB</td>
<td>SAS</td>
<td>00:00:43</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>STEPLIB</td>
<td>SASLPA</td>
<td>00:00:42</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>STEPLIB</td>
<td>SAS</td>
<td>00:59:23</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>STEPLIB</td>
<td>SASLPA</td>
<td>00:50:02</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>STEPLIB</td>
<td>SAS</td>
<td>1:03:56</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>STEPLIB</td>
<td>SASLPA</td>
<td>1:03:15</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>STEPLIB</td>
<td>SASLPA</td>
<td>1:03:44</td>
<td></td>
</tr>
</tbody>
</table>