

Installation Instructions and System Manager's Guide for Release 8.2 (TS2M0) of the SAS[®] System under CMS

Table of Contents

Chapter 1, Introduction and Orientation	1
How This Book is Organized	1
Installation Objectives	2
Tape Contents	3
References	5
Contacting SAS	5
 Chapter 2, Installing the SAS[®] System.....	 7
Part I, Overview and Checklist of the CMS SAS [®] Software Installation Steps	8
Part II, Pre-Installation Considerations	10
Part III, Loading the SAS [®] System from Tape to Disk Using the Installation Aids....	13
Running the GETSAS EXEC	13
The GETSAS User Interface	14
Installing From Multiple Tapes	17
Full-Screen Key Definitions	18
Rerunning GETSAS.....	19
Disk Space Implications.....	19
Return Codes from GETSAS.....	20
Part IV, Customizing and Verifying the SAS [®] System.....	21
Enabling the SAS Autocall Macro Libraries	21
Installation Verification Testing	22
Customizing the SAS EXEC	22
Settings for SAS System Options.....	23
Customizing SASV8SYS CONFIG.....	23
Customizing SASV8 CONFIG	23
SAS Sort Options	24
Windowing Environment PF Key Settings	25
Customizing Globally Used Catalog Members.....	25
Customizing Default Forms	25
Customizing Default Display Manager Keys	26
Using the NEWS and SAS HELPCMS Files.....	26
SASMAIL REXX Exec	27
Sample Libraries.....	27
Defining Printers	27
CMS specifics for variables needed by PROC PRTDEF	28
Considerations for printing disk files	28
Sending output directly to a printer.....	28
Using the PUNCH device for printing	30

Chapter 3, Installing the CMS SAS[®] System in Saved Segments	31
Overview.....	31
Step 1: Planning for Installing the SAS System in Saved Segments.....	32
Step 2: Defining the Saved Segments.....	34
Files generated by CMSSEGS command.....	35
Step 3: Creating the Saved Segments.....	36
Step 4: Storing and Saving the SAS System in the Saved Segments.....	36
Step 5: Setting Option Defaults to use the Saved Segments	37
Procedure Checklist:	38
Sample CMSSEGS file	39
CMSSEGS File for Saving Only User Libraries	40
Reference.....	41
Types of Saved Segments	41
Tailoring What Goes Into Segments	41
Collecting Segment Usage Data Using the SGSTAT option.....	41
CMSSEGS Control Files.....	42
LLIST Files.....	46
Chapter 4, Maintaining the SAS[®] System Under CMS	47
Part I, Updating Product Licenses Using SETINIT	48
Running PROC SETINIT	48
Updating the SAS Registry.....	49
For More Information	49
Part II, Applying Maintenance to SAS Software Products.....	51
Part III, SAS Notes and Zaps.....	52
Using SAS Notes.....	52
Applying ZAPs for Maintenance	52
Saved Segment Implications	53
Part IV, Installing Additional SAS System Products	54
What is an Add-On?.....	54
Instructions for Installing Your ADDON Tape	54
Appendix A, Implementing SAS/ACCESS[®] Interface to ORACLE[®]	57
Oracle Releases.....	57
Assigning Login Defaults	57
Creating and Loading the Sample Tables (optional)	57
Appendix B, Tailoring SAS/ACCESS[®] Interface to DB2 Server for VM.....	59
Appendix C, SAS/ACCESS[®] Interface to SYSTEM 2000[®] Software.....	63
Appendix D, Post-Installation Setup for SAS/ASSIST[®] Software.....	65
Profile Changes and Enhancements.....	65
Appendix E, Post-Installation Setup for SAS/CONNECT[®] Software.....	67
Appendix F, Installing SAS/GIS[®] Census Tract Maps	69
Installation Requirements.....	69
Tape Contents	69
Contacting SAS	70

Part I, Overview and Checklist of the SAS/GIS Census Tract Maps Installation Steps for CMS	70
Part II, Pre-Installation Considerations	71
Part III, Loading SAS/GIS Census Tract Maps from Tape to Disk Using the Installation Aids.....	73
Running GETGIS EXEC.....	73
The GETGIS User Interface	74
Installing from Multiple Tapes.....	75
Full-Screen Key Definitions	76
Rerunning GETGIS.....	77
Disk Space Implications.....	77
Post-Install.....	78
Return Codes from GETGIS.....	78
Appendix G, Post-Installation Setup for SAS/GRAPH[®] Software	79
Client-Side Components.....	79
ContourApplet (ctrapp.jar)	79
GraphApplet (graphapp.jar).....	79
MapApplet (mapapp.jar and related map data jar files).....	80
MetaViewApplet (metafile.zip).....	80
RangeViewApplet (rvapplet.jar).....	80
SAS/GRAPH Control for ActiveX.....	80
Installing the Client-Side Components.....	80
Managing Device Catalogs.....	81
Setting Up and Modifying Device Catalogs	81
How Device Catalogs are Used.....	81
How and When do You Modify Catalog Entries	82
Example	82
Using SAS/GRAPH with ASCII Devices.....	84
Setting up a SAS/GRAPH Translate Table.....	85
Using SAS/GRAPH with ASCII Terminals and ASCII Terminal Emulation Software.....	85
Using SAS/GRAPH with ASCII Printers.....	85
Installing the Linkable Driver	86
Using SAS/GRAPH with IBM 3270-Type Terminals and 3270 Emulators	86
Using SAS/GRAPH with IBM 3287, 3268, and 4224 Printers	87
Using SAS/GRAPH with GDDM	87
Appendix H, Post-Installation Setup for SAS/MDDDB Server[®] Software	89
Appendix I, Post-Installation Setup for the Metabase Facility.....	91
Setting Up the System Repository Manager Files.....	91
Registering the SASHELP Repository in the Repository Manager	92
Appendix J, Post-Installation Setup for SAS[®] OLAP Server Software	93
Setting up Access Control without SAS/EIS Software on Your Server.....	93
Starting the Access Control Setup Dialog Window.....	93
Setting Your Access Control Key and Environment Programmatically	93
Doing Your Access Control Definitions (Users, Groups, ACL) Programmatically	96
Write Access to SASHELP.AC and SASHELP.MB.....	105
Specifying OLAP Classes	105

Appendix K, Implementing SAS/SHARE[®] Software	107
Client-Side Components.....	107
SAS/SHARE Data Provider.....	107
SAS ODBC Driver.....	108
SAS/SHARE Driver for JDBC	108
SAS/SHARE SQL Library for C	108
Client Components.....	108
Appendix L, MACLIBs on the Installation Tape.....	109
Appendix M, CMS Storage Dumps.....	111
Appendix N, User Exits in the SAS[®] System under CMS.....	113
Using User Exits.....	113
Available User Exit Points.....	113
User Exit Source Code Specifications.....	114
How to Create a User Exit Load Module	117
How to Test Your User Exits.....	117
How to Make the User Exit Load Module a Permanent Part of the SAS System	118
SASUSER and Saved Segments	118
Federal Government Rights Notice.....	119
CMS SAS[®] Installation Feedback Form	121

Chapter 1, Introduction and Orientation

How This Book is Organized

This document contains instructions to guide you through your installation of Release 8.2 (TS2M0) of the SAS System under CMS. It is divided into four chapters and a set of appendices. The following describes the contents of the chapters and appendices.

- ❑ Chapter 1, **Introduction and Orientation** explains basic terms and concepts used throughout this book. Please read this section before continuing with any other section.
- ❑ Chapter 2, **Installing the SAS System** provides step-by-step instructions for the installation process. It contains the following sections:
 - “Overview and Checklist of the CMS SAS Software Installation Steps” is a concise list of the installation steps.
 - “Pre-Installation Considerations” helps you prepare for the installation, including instructions for loading the installation aids from tape to disk.
 - “Loading the SAS System from Tape to Disk” explains how to use the installation aids (specifically the `GETSAS EXEC`) to load the SAS System to disk.
 - “Customizing and Verifying the SAS System” explains what you need to do to make your SAS System ready to use at your site.
- ❑ Chapter 3, **Installing the CMS SAS System in Saved Segments** explains putting the CMS SAS System in saved segments to reduce memory requirements and improve performance.
- ❑ Chapter 4, **Maintaining the SAS System** explains the tasks you may need to perform to update and maintain your SAS System under CMS. It contains the following sections:
 - “Updating Product Licenses using `SETINIT`” contains instructions for license renewal and new product licensing.
 - “Applying Maintenance to SAS Software Products” describes how to apply maintenance, including broad base fixes to the SAS System.
 - “SAS Notes and Zaps” explains how to install updated information about known problems.

- “Installing Additional SAS Software Products” explains how to incorporate add-on products into your existing SAS System.
- **Appendices** supply information for specialized concerns, including notes on special implementation processing for individual products. Review the appendices for the products you install.

Note: Be sure to read the cover letter, system requirements sheet, and alert notes that are included in your software package. Keep these items and this document for reference after you complete the installation.

A soft copy of this document is provided on the tape. It is called `INSTALL DOC` and is loaded from the tape with the installation aids. It can be used to search for special instructions, but it is not formatted or meant to be printed.

Installation Objectives

One of the following installation objectives should describe your goal, depending on the type of tape you are installing (which has been custom made for your site) and what you want to accomplish:

- **New SAS Release 8.2 site, including established SAS site running a release prior to Release 8.2 (TS2M0)**

You want to install Release 8.2 (TS2M0) of the SAS System under CMS for the first time. Your tape contains Base SAS Software and all other products you have licensed. The external label on the tape reads *PRODUCT & SAS SUPPORT TOOLS*.

SAS Support Files include a `ZAPLIB` (if necessary) and a **Read Me** file that contains information on SAS Notes. SAS Notes are reports that were previously included with the tape and are now available via the SAS Web site.

Even if you already have a prior release of the SAS System installed on CMS, you need to install Release 8.2 in a new location. This is a new release, not a maintenance tape. Do not install it as maintenance over your existing SAS System.

If this is your objective you should finish reading this chapter and then follow the instructions in Chapter 2, “Installing the SAS System under CMS,” and Chapter 3, “Installing the CMS SAS System in Saved Segments.”

- **Established SAS site wanting current SAS Notes or Maintenance**

You currently have Release 8.2 of the SAS System under CMS installed, and you have requested the current Maintenance. Your tape contains the Maintenance. The external label on the tape includes the words *MAINTENANCE FILES*.

If this is your objective, please finish reading this chapter and then follow the instructions in Chapter 4, “Maintaining the SAS System.”

❑ Established SAS site requesting additional products

You currently have Release 8.2 of the SAS System under CMS installed, and you want to install an additional SAS System product. Your tape contains the new product. The label on the tape reads *PRODUCT & SAS SUPPORT TOOLS*.

If this is your objective, please finish reading this chapter and then follow the instructions in Chapter 4, "Maintaining the SAS System."

Tape Contents

The tape that accompanies this document contains Release 8.2 (TS2M0) of the SAS System under CMS. The installation tape is a non-labeled tape, and the files are in IBM's TAPE DUMP format. Except for the installation aids, the files are blocked VB and must be read using `BLOCKTAP`[†] which is included with the installation aids. The tape's external label indicates the density of your tape.

Note: This document refers to the installation tape rather than tapes. It is one logical tape, although it can extend over two or more physical reels or cartridges. If you received more than one physical tape, the external labels indicate the number of reels or cartridges. For example, the labels for a two-reel tape read *1 of 2* and *2 of 2*.

The products contained on your installation tape are listed on the transmittal letter contained in your installation package. You can also list the products that are on the tape and their space requirements by loading the SAS installation aids from the tape, and then with the tape still attached as virtual device 181, issuing:

```
GETSAS TAPEMAP
```

Detailed instructions for doing this are included later in this document. If you do not have the correct tape, contact the Distribution Center at SAS for a replacement by sending email to `distrib@vm.sas.com` or by calling (919) 677-8000, extension 7850, Monday-Friday between 9 a.m. and 8 p.m., Eastern time.

The installation tape is a collection of CMS files that are logically divided into the following sections:

❑ Installation Aids

This section contains files used to install the SAS System, including the `GETSAS EXEC`. These files are loaded to disk using the `CMS TAPE LOAD` command. The `GETSAS EXEC` is the *driver* for the rest of the installation process. This is what you run to load the rest of the SAS System from tape to disk.

[†] `BLOCKTAP` was written by Steve Howes at Brigham Young University. It was written to improve the functionality of `TAPE DUMP` by trapping `TAPEIO` calls and blocking at 20K, 32K, or 64K. The source code and related `MACLIB` have also been included on this tape. Any problems related to this tape and `BLOCKTAP` should be directed to SAS Institute Inc. Many thanks to Steve for his good work.

□ **SYSTEM**

This section contains any SAS software licensed by your site. The files in this section are required in order to run the SAS System (i.e., loadlibs, modules, SAS help files, message files, etc.). The `SYSTEM` files for each SAS software product are grouped together, with each product having its own tape file.

Installation verification programs are also in this section. These test programs are described in "Installation Verification Testing" on page 22.

□ **SAMPLE**

This section contains libraries of SAS sample programs that illustrate features of SAS procedures and programming techniques. Most (but not all) SAS products have sample libraries. On the tape the sample files are grouped together by product, with each group having its own tape file.

We encourage you to install and publicize these programs. SAS Institute analysts often refer users to sample programs for answers to programming questions.

If you are an established site receiving only SAS Notes, your tape does not include a `SAMPLE` section.

□ **MAPS**

This section is included only when SAS/GRAPH software is on the tape.

If no one at your installation uses `GMAP`, the map data sets do not have to be loaded to disk. If the map data sets can be used but you are short on disk space, you can compromise and erase any large or unnecessary data sets, keeping only the smaller data sets on disk.

□ **SUPPORT TOOLS (ZAPLIB)**

This section contains a Read Me file with information on the SAS Notes, which are reports that were previously included with the tape. SAS Notes are now available via the SAS Web site. This section can also contain files for short-term emergency software maintenance, most commonly in the form of ZAPs.

References

The following SAS manuals can assist you and your users with Release 8.2 of the SAS System. These manuals can also provide helpful information for completing your system set up and product implementation:

- ❑ *SAS Language Reference: Dictionary, First Edition*
- ❑ *SAS Procedures Guide, First Edition*
- ❑ *Getting Started with the SAS System in the CMS Environment, First Edition*
- ❑ *SAS Companion for the CMS Environment, First Edition*
- ❑ *SAS/GRAPH Software: Reference, First Edition,*
- ❑ *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software, Second Edition*

Contacting SAS

If you have difficulty with this document or any of the procedures described in it, contact the SAS Technical Support Division at (919) 677-8008, between 9 a.m. and 8 p.m., Eastern Time.

Chapter 2, Installing the SAS[®] System

Part I, Overview and Checklist of the CMS SAS Software Installation Steps

Part II, Pre-Installation Considerations

Part III, Loading the SAS System from Tape to Disk

Part IV, Customizing and Verifying the SAS System

Part I, Overview and Checklist of the CMS SAS[®] Software Installation Steps

The following is an overview of the steps to follow when installing SAS software for the CMS environment. A complete explanation for each step is provided later in this document.

1. Read all of the installation instructions before attempting to install SAS software.
2. Review all the items in your product package.
3. Ensure your operating system and hardware meet the specifications indicated on the System Requirements document contained in your package.
4. Obtain write access to a small amount of clean disk space to hold the installation aids.
5. Mount the tape and have the tape drive attached to the userid.
6. Load the SAS installation aids (including the `GETSAS EXEC`) from tape to disk using `TAPE LOAD`.
7. Determine the disk space requirements for CMS SAS software. This can be done by running the `GETSAS EXEC`.
8. Obtain the required resources to hold the SAS System, including clean disk space to which you have write access.
9. Install the SAS System from tape to disk by running the `GETSAS EXEC`. `GETSAS` prompts you for all necessary information, then loads products from the tape to disk.
10. Enable the SAS Autocall Macro Libraries.
11. Run CMS SAS programs for installation verification.
12. Edit and customize the `SAS EXEC`.
13. Review and customize installation defaults for SAS system options.
14. Customize catalog members used globally by users.
15. Review and customize the `NEWS` file.
16. Add Help File to CMS HELP Facility.
17. Review the information on sample libraries.

18. Perform product-specific installation tasks:

- ❑ If you have SAS/ACCESS to ORACLE software, see the appendix "Implementing SAS/ACCESS Interface to Oracle" on page 57.
- ❑ If you have IBM's DB2 VM software and SAS/DB2 VM software, link the IBM DB2 VM resource manager stub routine with the SAS/DB2 VM software. See the appendix "Tailoring the SAS/ACCESS Interface to DB2 Server for VM" on page 59.
- ❑ If you have SYSTEM 2000 software, tailor the `SASS2K EXEC`. See the appendix "SAS/ACCESS Interface to SYSTEM 2000 Software" on page 63.
- ❑ If you have SAS/ASSIST software, see the appendix "Post-Installation Setup for SAS/ASSIST Software" on page 65.
- ❑ If you have SAS/CONNECT Software, refer to *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software*.
- ❑ If you have SAS/GRAPH software, tailor SAS/GRAPH if required. See the appendix "Post-Installation Setup for SAS/GRAPH Software" on page 79.
- ❑ If you have SAS/MDDDB Server Software, see the appendix "Post-Installation Setup for SAS/MDDDB Server Software" on page 89.
- ❑ If you have SAS/SHARE software, prepare the server machine. See the appendix "Implementing SAS/SHARE Software" on page 107.

Note: The remaining SAS software products do not have product-specific installation instructions.

19. Create the CMS SAS saved segments. See Chapter 3 for more information. **We strongly encourage you to install your SAS software products in saved segments.**

20. Rerun CMS SAS programs for installation verification.

21. Read the file `GOVT NOTE` (U.S. Federal Government employees only). This section appears on page 119 at the end of this document.

If you have a prior release of the SAS System installed on CMS, any customizations that were made at your site that you want made available with Release 8.2 will need to be remade.

After you have finished the installation, and the `SETINIT` and Registry files have been updated, you can erase the installation aids. The installation aids are specific to the tape they are on, so the installation aids from one tape should not be used to install another tape.

Part II, Pre-Installation Considerations

Before you begin installing the SAS System, perform the following tasks:

- ❑ Review all the items in your product package, including the Cover Letter, Transmittal Letter, Release Notes, Alert Notes, and Copyright Notice. Unless otherwise noted, your product package from SAS contains all the items listed in the Transmittal Letter. The product package is shipped to the SAS Software Representative at your site. If you think any items are missing from your package, contact your SAS Software Representative who in turn may need to contact SAS.
- ❑ Ensure your operating system and hardware meet the specifications indicated on the System Requirements document contained in your package.
- ❑ Obtain write access to a small amount of empty disk space (600 4K blocks should be enough). The disk space can be either a minidisk or CMS Shared File System directory. This will be used to contain the installation aids. The installation aids should go on their own disk, rather than putting them on the same disk as the rest of the SAS System or other software products. The installation aids are specific to the tape they are on, so the installation aids from one tape (or set of tapes) should not be used to install another tape (set).
- ❑ Have the CMS SAS distribution tape mounted on a tape drive and have the drive attached to your virtual machine at address 181. The tape is nonlabeled, and the density is indicated on the external physical label. Mount the tape in read-only mode.
- ❑ Load the SAS installation aids (including the `GETSAS EXEC`) from tape to disk. Ensure that you are at the beginning of the tape by issuing:

```
TAPE REW
```

Next, load the files from tape to disk by issuing the following command:

```
TAPE LOAD * * filemode
```

where *filemode* indicates the filemode of the minidisk or Shared File System directory that will contain the installation aids.

When this step is completed, the `GETSAS EXEC` and all other files from the Installation Aids section have been loaded to disk. Leave the tape mounted so that you can continue with the next step.

- ❑ Determine the disk space requirements for CMS SAS software. You can have the disk space requirements calculated for you by running the `GETSAS EXEC`. With the tape still mounted and attached as virtual device 181, issue:

```
GETSAS
```

Answer the selection prompts, and exit (F3) from the screen showing disk requirements. This screen also prompts for filemodes, but as long as you exit from this screen none of the pieces of the SAS System get loaded off the tape. A soft copy of the disk space requirements based on your selections can then be found in the file `GETSAS AUDIT` on the installation aids disk. The numbers given are the recommended number of 4K blocks, including a minimal amount of padding. See the online help (F1) on the disk requirements screen if you want more details.

Your answers to the prompts will be saved and can be used as defaults in a subsequent invocation of `GETSAS` (i.e., when you are ready to do the actual loading of the SAS System from tape to disk). Doing this makes running `GETSAS` a two-pass process. On the first pass, you select what you want to install and you are given summarized disk space requirements. Then you set up your disk space. On the second pass, you can have your previous selections used as default. Then you specify where you want things to go and the SAS System gets loaded from tape to disk.

For the most current notes on using `GETSAS`, see the online help associated with each screen (including general information available from the help on the very first screen).

Other methods can be used to determine disk space requirements. First decide what sections and products you will unload from the tape. Then look up the corresponding sizes in either the System Requirements document or the screen output that is displayed when you issue:

```
GETSAS TAPEMAP
```

with the tape still mounted and attached. You can then manually add up each of the sizes to find out the total disk space requirements. Be sure to include some padding if you are manually calculating disk requirements. The numbers in the System Requirements document no longer include any padding. The recommended minimum padding is 1000 4K blocks for the disk that will hold system files, and 150 4K blocks for each of the other sections you select (sample, maps, and `zaplib`).

Notes: When you install Release 8.2 of the SAS System, you can install everything from your tape, or you can select only certain sections of the tape (System, Sample, Maps, and `zaplib`). You can also select specific products. This includes specific sets of maps if you license SAS/GRAPH software.

You can install all of the SAS System to a single location or you can install each section of the tape (System, Sample, Maps, and `zaplib`) to a separate location. Any combination of sections can be grouped into separate locations. You should not however install different products to different locations. For example, you could put the System files on one disk, the Samples, Maps, and SAS Notes on a second disk.

- Obtain the required disk space. You need to have write access to this disk space and the disk space should be empty. The disk space that you set up should be used exclusively for SAS software, rather than installing the SAS System to a location containing other software products. Some of the modules in the SAS System may duplicate names of modules in other packages. Installing the SAS System to its own location ensures that you do not inadvertently overwrite another program. Also, installation of new SAS software releases and products is easier when you start with a newly formatted minidisk or empty Shared File System directory.

We recommend a 4K disk block size for the SAS System disk. However, files can be loaded on disks formatted in different block sizes. The System Requirements sheet included in your installation package contains information to help you calculate the number of cylinders necessary to install the SAS System. For more information, refer to the System Requirements document.

Note: Throughout this document, references to CMS minidisk can be replaced by accessed CMS Shared File System directory.

Part III, Loading the SAS[®] System from Tape to Disk Using the Installation Aids

Running the GETSAS EXEC

Use the GETSAS EXEC to install the SAS System. GETSAS is a full-screen application that prompts you for necessary information, such as what you want to install, and where you want it installed. Based on your answers, the appropriate files are then loaded from tape to disk.

Before running the GETSAS EXEC, perform the tasks listed in Part II, "Pre-Installation Considerations" on page 41. For example, you need to load the SAS installation aids (including the GETSAS EXEC) from tape to disk.

Note: Among the tasks that GETSAS performs is updating the SASHELP library. This is done through a SAS job, so the SAS System must be invoked without segments. A virtual machine size of at least 32 megabytes is required.

You must also have the CMS SAS distribution tape mounted on a tape drive and have the tape drive attached to your virtual machine at address 181. The tape is nonlabeled and the density is indicated on the external physical label. Mount the tape in read-only mode.

To perform the install, run GETSAS from a CMS Ready prompt by issuing:

```
GETSAS
```



Note: Uppercase Installation: If you have licensed uppercase SAS and wish to perform the installation in uppercase, issue the following:

```
GETSAS UPCASE
```

Continue with the remainder of the installation as described in the text below.

Answer the prompts and let GETSAS do the rest.

The GETSAS EXEC includes the following features:

- Automatic calculation and checking of disk space requirements.
- Ability to page backward through the screens (and change previous answers).
- Screens requiring filemodes allow you to go into CMS subset mode.
- Expanded online help. Select F1 from any screen to receive help.
- After everything has been loaded from tape to disk, the SASAUTOS EXEC is automatically run to merge the autocall macro libraries.

- ❑ An audit trail is kept that records your answers to the prompts and information about your run time environment.
- ❑ Answers from a previous invocation can be automatically used as defaults.

Release 8.2 is a new release. If you already have a prior version of the SAS System installed on CMS, install Release 8.2 to a new location; do not install it as maintenance over your existing SAS System. Any customizations you have made to a previous version of the SAS System that you want to be available with Release 8.2 at your site will need to be remade.

Do not install different products to different locations. If you do, subsequent maintenance installs will fail and you may end up with unapproved combinations of maintenance levels. As part of GETSAS processing, some files are created with the naming convention SAS* KEEP_ME. As the filetype implies, **do not** erase these files or move them to a different disk. If you do, subsequent installs (i.e., for reruns, maintenance, or addons) will fail.

For the most current notes on using GETSAS, see the online help associated with each screen (including general information available from the help on the very first screen).

Note: The GETSAS EXEC turns on BLOCKTAP and NUCXLOADs it. When GETSAS terminates it turns off BLOCKTAP and NUCXDROPs it.

The GETSAS User Interface

GETSAS operates in full-screen mode. When you issue the GETSAS command, the following message appears:

NOTE: Positioning tape and loading tape map.

Next, the following screen appears:

```

CMS SAS Release 8.2 TS2M0                                     1
Thank you for licensing SAS Software products for the CMS system.
    Copyright(c) 1999-2001 by SAS Institute Inc., Cary, NC USA.

See HELP for an overview of the installation procedure.

                Press ENTER to continue

Enter=Continue  F1=Help  F3=Quit  F12=Backtrack

```

Press Enter to continue.

GETSAS then gives you the choice of installing the entire tape or of installing only selected parts. If you choose to do a selective installation, you may choose which products to install, and which additional sections (Samples and Maps, if any) to install for those products.

If you choose a selective installation, license Base and receive 3480 cartridge media, Base will appear as two entries on the software Product List display: Base SAS and Base SASHELP Library. If Base SAS is selected, the Base SASHELP Library will automatically be included within the Base SAS tape unload.

After you have either selected a complete install, or completed your choices for a selective install, GETSAS reports the amount of disk space (in 4K blocks) that will be required and prompts you for the filemode (or filemodes) where you want the products to be installed. The help screen at this point supplies detailed information on how the requirements were calculated. If you need to suspend GETSAS at this point to prepare the disk space, you can press PF2 to enter CMS Subset mode.

You can exit from GETSAS at any point by pressing PF3. If you do so at or beyond the point at which disk space requirements have been calculated, the requirements will be saved in file GETSAS AUDIT for your reference as you allocate space. When you rerun GETSAS, your previous responses become the new default responses so you can quickly return to where you left off.

When you continue past filemode selection, GETSAS summarizes disk space allocation by minidisk or filespace for your review, and warns if you have not allocated sufficient space.

The last GETSAS screen, shown below, asks if you want to see the names of files as they are loaded from the tape. No files are actually installed if you exit from this screen or any prior screen.

```
CMS SAS Release 8.2 TS2M0                                18
As files are loaded from the tape, the filename, filetype,
and destination filemode can be displayed.

Do you want fileids displayed as files are loaded (Y|N)? Y

This is the last prompt before files are loaded from the tape.
When you press Enter, loading will begin.

Enter=Continue  F1=Help  F3=Quit  F12=Backtrack
```

After you have responded to all prompts, GETSAS loads or bypasses the appropriate files from the tape to the specified disks. This process takes several minutes.

If you directed GETSAS to display the file information as the files are loaded from tape, information similar to the following is displayed:

NOTE: Positioning tape and loading tape map...

NOTE: Verifying tape...

NOTE: Positioning tape to first file...

Loading...

SETNAME SAS D2

GOVT NOTE D2

End-of-file or end-of-tape

Loading...

OCORE SASHELP D1

End-of-file or end-of-tape

Loading...

SASBASE LOADLIB D2

SASUSER ASSEMBLE D1

-

-

-

SUPPORT EXEC F1

SUPPORT SAS F1

VMLIBNAM SASMACRO F1

End-of-file or end-of-tape

GETSAS: Loading from tape is complete.

The tape is being rewound.

SASAUTOS MACLIB NOT FOUND, WILL CREATE ONE

. . . SASBAUTO MACLIB A2 WILL BE MERGED

. . . ADDING AF

. . . ADDING ANGLE

-

-

-

. . . ADDING DS2CSV

. . . ADDING FILESERV

The SASAUTOS MACLIB is complete.

. . . Compressing SASAUTOS MACLIB

SASAUTOS MACLIB A is completed and compressed.

Use system option

sasautos=('sasautos maclib *')

to enable use.

Press Enter to continue.

GETSAS will now invoke the command

```
EXEC SAS SETNAME (SETINIT NOSSEG ALTLOG=TERMINAL NOOVP
```

to apply the SETINIT information (licensing information).

SAS software will not run without valid SETINIT information.

You have the option of bypassing this step and doing it later. Do you want the SETINIT information to be applied now? (Y|N)

Y

The SETINIT information has been successfully applied.

Checking for any Registry updates, please be patient.

The REGISTRY information has been successfully applied.

GETSAS: Automatic post-processing is complete.

Please perform required post-installation tasks for the following products (see installation documentation):

BASE SAS

Please run installation verification tests.

For best performance, we recommended that you install the SAS System in saved segments (see installation documentation).

GETSAS: installation completed successfully.
Ready;

When all requested sections and products are loaded, the tape is rewound (but not unloaded or detached).

If system files were installed, the SAS autocall MACLIBs are automatically merged. Several messages are issued recommending the post processing steps you should take. Status and error messages issued to the screen are also written to the file GETSAS AUDIT for later reference. Then the exec terminates.

Note: As part of GETSAS processing, some files are created with the naming convention SAS* KEEP_ME. As the filetype implies, these files should not be erased. If you do, subsequent installs (for reruns, maintenance, or addons for example) will fail.

For post-installation tasks, please see Part IV, "Customizing and Verifying the SAS System" on page 21, and Chapter 3, "Installing the CMS SAS System in Saved Segments" on page 31. Also see the appendices for product-specific install tasks.

Installing From Multiple Tapes

If your installation tape extends over two or more physical tape reels or cartridges, the files selected may not exist on the current tape. When this happens, GETSAS prompts you to mount the next tape with the following sequence of messages:

```
End-of-file or end-of-tape
NOTE: The current tape has been completed and is being
rewound.
GETSAS will now put you in CMS Subset mode so that you can
mount the next tape.
The tape that needs to be mounted is externally labeled
nnnnnn.
When the tape is ready, enter the RETURN command to resume
GETSAS.
Waiting for tape rewind to complete before entering CMS
subset...
Tape is rewind.
CMS subset
```

The tape needs to be attached as virtual address 181 so you may need to drop the previous tape by issuing:

```
DETACH 181
```

Once the next tape has been mounted and attached, and you have issued the RETURN command from CMS subset, GETSAS continues the loading process. Information similar to the following is displayed if it was requested:

```
NOTE: Verifying tape...
NOTE:   Positioning tape to first file...
      Loading...
      GSOURCE MACLIB E1
```

Full-Screen Key Definitions

One or more of the following key definitions may be active and appear on a screen:

- Enter=Continue

Press ENTER to continue to the next screen. GETSAS validates the data you have entered. If there is invalid data, it will not continue to the next screen.

If the screen you are using has fewer lines than GETSAS is attempting to display, Enter will scroll down to show the rest of the information. Press ENTER again to continue.

- F1=Help

Press F1 or F13 for screen-sensitive online help. This includes general help on the initial screen.

- F2=CMS Subset

On the disk requirements and filemode selection screen, press F2 or F14 to go into CMS subset mode. Use the RETURN command to return to GETSAS.

- F3=Exit

Press F3 or F15 to terminate GETSAS.

- F6=Describe

On screens that list SAS products, press F6 or F18 to see the product descriptions.

- F7=Scroll up

Within Help screens, press F7 or F19 to scroll up.

- F8=Scroll down

Within Help screens, press F8 or F20 to scroll down.

- F12=Backtrack

Press F12 or F24 to return to the previous screen. Within a help screen this returns you to the screen from which you issued help.

Rerunning GETSAS

After running GETSAS to load files from tape to disk, you may sometimes need to rerun GETSAS. Typical reasons for doing this include wanting a product or sample that you did not previously select, or having to reinstall a product that has been erroneously modified. In this type of situation, simply rerun GETSAS and select the sections/products that you want to load to disk.

Any time you rerun GETSAS you need to have write-access to all disks that contain any part of your SAS System Release 8.2 CMS installation. One exception is that if you want to load the `zaplib` section of the tape and nothing else, you only need write-access to your SAS `zaplib` location.

GETSAS keeps track of what has been previously installed, and sets selection defaults to 'N' for those selections/products.

You may need to (re)do post-installation tasks after rerunning GETSAS. See the section "Customizing and Verifying the SAS System" on page 21 and Chapter 3, "Installing the CMS SAS System in Saved Segments" on page 31.

Every time you run GETSAS, a file named `GETSAS_AUDIT` is written that contains an audit trail of the execution. If you want to keep the version of this file from a previous execution, you must rename it or make a copy of it before rerunning GETSAS.

Disk Space Implications

If you select anything that was not previously installed, use the information from the disk space requirements screen to help you determine how much disk space you need.

If you want to completely re-install everything you previously installed, erase all files that GETSAS loaded to disk before you rerun GETSAS. In this case be sure to erase the `SAS* KEEP_ME` files, but do **not** erase the installation aids.

If you want to reinstall a subset of what you previously installed, you still need to ensure that you have enough disk space to reload the files. When `TAPE LOAD` overwrites an existing file, it temporarily needs space for both the new and the old copies.

There are two ways to ensure that you have enough disk space:

- Erase any files that you plan to reload that are bigger than the available space currently on the disk.
- Get enough additional disk space to hold an extra copy of the largest file that you plan to reload.

If you have set up your original disk space with the recommended 1000 4k extra blocks, these are the only files that need extra space:

Base SAS	SASBASE	LOADLIB	14861 4k blocks
	7HELPOC	SASHELP	11830 4k blocks
	0FSP	SASHELP	2298 4k blocks
SAS/ASSIST	0ASSIST	SASHELP	2207 4k blocks
SAS/ETS	SASETS	LOADLIB	2721 4k blocks
SAS/GIS	0GISDATA	SASHELP	3175 4k blocks
SAS/GRAPH	SASGRAPH	LOADLIB	3219 4k blocks
SAS/STAT	SASSTAT	LOADLIB	4114 4k blocks

Return Codes from GETSAS

Specific descriptive messages are issued to the console for all non-zero return codes.

0	Executed successfully
24	Invalid option specified with GETSAS. <i>or</i> GETSAS was issued from within CMS SUBSET. <i>or</i> Unrecognized tape type XXXX. <i>or</i> Invalid GETSAS TAPEMAP file (possible invalid SAS distribution tape).
28	Required installation aids file not found or not unique. <i>or</i> Tape not attached as 181 or wrong tape. <i>or</i> Could not turn on blocktap.
36	Write access to the SAS installation tools disk is required but not available.
100	TAPE LOAD failed or other TAPE command error
101	Abend

Part IV, Customizing and Verifying the SAS[®] System

Enabling the SAS Autocall Macro Libraries

Many of the products available with the SAS System (including Base SAS) come with useful macros. These macros are shipped by product in macro libraries, and referred to as autocall macros.

When you run GETSAS, after everything you've selected has been loaded from the tape, these product specific macro libraries are automatically merged into a single file called SASAUTOS MACLIB. If a problem occurs while the SASAUTOS MACLIB is being created (for example disk full), then you must manually rerun the SASAUTOS EXEC to create a correct version. If you do not have a correct version of SASAUTOS MACLIB, you will get an error from the SAS System when you try to use the autocall facility to access a macro.

To rerun the SASAUTOS EXEC get write access to your SAS system disk as filemode A, and issue:

```
SASAUTOS A
```

For more information on this exec, issue:

```
SASAUTOS ?
```

The option setting of SASAUTOS='SASAUTOS MACLIB *' specified in the SASV8SYS CONFIG file gives the SAS System access to these macros. If you have macros of your own that you want available through the autocall facility, put them in their own MACLIB or save them in an SFS directory with filetype SAS. Then add the name of the MACLIB or SFS directory to the SASAUTOS= option. For example:

```
SASAUTOS= ('SASAUTOS MACLIB *' 'name2 maclib b')
```

Or

```
SASAUTOS= ('SASAUTOS MACLIB *' 'user.sasautos.directory')
```

If you prefer to store the autocall macros as separate files in an SFS directory, create the directory and re-run SASAUTOS specifying the directory name instead of a filemode. Then change the SASAUTOS statement in SASV8SYS CONFIG to reference the SFS directory instead of the MACLIB.

For more information on the SASAUTOS option and file specification, refer to the *SAS Language Reference: Dictionary*, and the *SAS Companion for the CMS Environment*.

Installation Verification Testing

In this step, you test the SAS System software to verify that the installation was successful. The following files are provided to help you test CMS SAS software:

TESTAF	SAS	test job input for SAS/AF
TESTBA	SAS	test job input for Base SAS procedures
TESTET	SAS	test job for SAS/ETS software
TESTFS	SAS	test job for SAS/FSP software
TESTGI	SAS	test job for SAS/GIS software
TESTGR	SAS	test job for SAS/GRAPH software
TESTML	SAS	test job for SAS/IML software*
TESTLB	SAS	test job for SAS/LB software
TESTMX	SAS	test job for SAS/MDDB software
TESTOR	SAS	test job for SAS/OR software
TESTORL	SAS	test job for SAS/ACCESS Interface to ORACLE*
TESTQC	SAS	test job for SAS/QC software
TESTST	SAS	test job for SAS/STAT software
TESTS2K	SAS	test job for SAS/ACCESS Interface to SYSTEM 2000*

Note: When running these test programs, link the SAS System disk in read-only mode. This will prevent the SASLOG and/or LISTING files from being created on the SAS System disk. If you create them accidentally on the SAS System disk, delete them. An '*' denotes a LISTING file being created.

To run a test job, enter a SAS command and the filename of the test you want to execute. For example:

```
SAS TESTBA
```

The test programs produce output files with filetypes SASLOG and LISTING. When no saved segments are used, the programs must be run in a virtual machine defined to be at least 32M. Note that the programs can take up to 20 minutes to execute. If a test program completes with a return code of '0', the test was successful.

Customizing the SAS EXEC

The installation tape contains a file called SAS EXEC. This is the EXEC used to invoke the SAS System. You can edit the SAS EXEC to tailor it for your installation. Once you have loaded the SAS System, the SAS EXEC file is on your SAS System minidisk and can be edited as long as you have write access to the minidisk. You may want to put the SAS EXEC on a commonly accessed system disk and add code to the EXEC to access the SAS System disk. Please keep track of all customizations in case you need to remake them for a future version.

We do not recommend that users access the SAS System disk as a read only extension of their A-disk. This can cause problems accessing SAS catalogs and data sets.

Settings for SAS System Options

The SAS System uses SAS system options to control many important features. SAS Institute supplies default values for these options, but you can change the supplied settings to settings that are appropriate for your site.

For descriptions of SAS system options, refer to the *SAS Language Reference: Dictionary*, and *SAS Companion for the CMS Environment*. Descriptions of some important SAS system options also appear later in this section.

If you are installing an add-on product, please read the section below on customizing the SASV8SYS CONFIG file.

Please keep track of all customizations in case you need to remake them for a future release.

Customizing SASV8SYS CONFIG

SAS Institute provides two files for changing SAS system options, SASV8SYS CONFIG and SASV8 CONFIG. The SASV8SYS CONFIG file is part of the SAS System disk and should only be changed by the site representative. The options placed in this file affect all SAS users.

The LOADLIBS used by the SAS System are identified in the SASV8SYS CONFIG file by SASLOAD= options. It is recommended that you comment out SASLOAD= options for products not installed at your site. To comment out a line in SASV8SYS CONFIG, put an asterisk (*) in column one.

Customizing SASV8 CONFIG

Users can create a SASV8 CONFIG file on their own minidisks to contain any SAS system option settings (with the exception of the CONFIG= option). If the same option is included in both the SASV8 CONFIG file and the SASV8SYS CONFIG file, SAS uses the value specified in the SASV8 CONFIG file.

In some prior versions of the SAS System, all PROFILE SAS files were concatenated and used. However, this is not true for the SASV8 CONFIG and SASV8SYS CONFIG files. Refer to the *SAS Companion for the CMS Environment* for more information on the CONFIG= option and the SASV8 CONFIG file. To assist you in your own configuration, an example of a SASV8 CONFIG file is on the installation tape.

Note: When using an option in the SASV8SYS or SASV8 CONFIG files that requires a value or argument, an equal sign (=) must be used, as in the following example.

```
SASLOAD='SASBASE LOADLIB *'
```

SAS Sort Options

SAS Institute supplies one sort program, `SASSORT`, as part of the SAS System under CMS. SAS also supplies standard `E15` and `E35` sort interface routines. Products using these routines, such as `SYNCSORT`, `CASORT`, `VMSORT`, and `DFSORT`, can be used easily with the SAS System by setting the `SORTLIB` and `SORTPGM` options appropriately.

You should evaluate the sort programs available to you and decide which performs best in your environment. If your installation licenses `SYNCSORT`, `DFSORT`, `VMSORT`, or `CASORT`, we recommend that you use it with the SAS System for large sorts.

For smaller data sets use `SASSORT`, an in-memory sort program that is efficient for data sets with fewer than 2500 observations.

The `SORTPGM` and `SORTLIB` options control which sort program is invoked for SAS programs. How these options are used and their valid specifications are as follows:

□ `SORTPGM=program`

specifies the sort program, where `program` is a keyword indicating the entry point into the globalized `TXTLIB`. The default is `SORTPGM=BEST`. Valid specifications are:

```
SORTPGM=SAS for SASSORT
SORTPGM=SORT for third-party sort
```

or

```
SORTPGM=BEST
SORTPGM=HOST
```

The values of `SORT`, `BEST`, and `HOST` require that the program name available in the `TXTLIB` specified in the `SORTLIB` option be `SORT`.

□ `SORTLIB=filename`

specifies a sort `TXTLIB` that needs to be globalized, where `filename` is a keyword indicating the library. Valid specifications are:

```
SORTLIB=SYNCSORT for SYNCSORT
SORTLIB=VMSLIB for VMSORT
SORTLIB=CASORT$C for CASORT
SORTLIB=DFSRTLIB for DFSORT.
```

Note: Syncsort customers must have at least Syncsort CMS Release 6.4C installed, or Syncsort support fix P0364082 must be installed to support SAS Release 8.2.

The SAS System is shipped with a default of `SORTPGM=BEST`. If `SORTLIB` is also specified, the SAS System will choose the appropriate SAS- or host-supplied sort for

each data set. For complete information on the SORT procedure and associated options, refer to the *SAS Companion for the CMS Environment*.

Six translate tables used by PROC SORT for determining a collating sequence are provided. All tables are in the SASHELP library in the HOST catalog.

The provided tables are ASCII, DANISH, EBCDIC, NATIONAL, REVERSE, and SWEDISH. The NATIONAL table is a machine-native collating sequence that you can customize. If you use an alternate SORT collating sequence, make sure all users use the SORTSEQ option on their PROC SORT statements. Consult the *SAS Procedures Guide* for more information on PROC SORT.

Windowing Environment PF Key Settings

Determine the most appropriate PF key settings and use the PFKEY= option to specify these settings. The PFKEY= option allows you to specify a primary set of 12 program function keys or key definitions. Acceptable values are PRIMARY, ALTERNATE, 12. The default setting is PRIMARY. These values enable you to choose sets of function keys, as follows:

PRIMARY	PF13 - PF24	=	Version 5 settings (command keys)
	PF1 - PF12	=	Version 6&7 settings (window management)
ALTERNATE	PF13 - PF24	=	Version 6&7 settings (window management)
	PF1 - PF12	=	Version 5 settings (command keys)
12	PF1 - PF12	=	Version 5 settings with only 12 function keys active

If the PFKEY= option does not meet your needs for changing function keys, see "Customizing Globally Used Catalog Members" below.

Customizing Globally Used Catalog Members

Many catalog members affect all users of the SAS System at a site. Catalog entries such as key settings, ASSIST defaults, and default print forms can be controlled by altering the appropriate catalog member of the SASHELP library. See the appendix "Post-Installation Setup for SAS/GRAPH Software" on page 79 for details on managing SAS/GRAPH device catalogs.

Please keep track of all customizations in case you need to remake them for a future release.

Customizing Default Forms

You can customize the FORM window. In addition to the information given below, read the information available in the *SAS Language Reference: Dictionary* and the *SAS Companion for the CMS Environment* on the FORM window.

The printer selection list can be customized to display only those printers available for your site. To find out more about customizing the printer selection list, issue the following command from the windowing environment command line:

```
AF C=SASHELP.BASE.PDEVICE.CBT
```

This command displays a series of Help windows that describes the process you need to follow. To receive help for a choice, tab to that choice and press `ENTER`.

If you have SAS/AF installed, the `PRINT FILE PARAMETERS` program can be customized for your site.

Customizing Default Display Manager Keys

You can globally change the program function keys for Display Manager for all users. If using the `PFKEY=` option does not meet the needs of your site, you can change the SASHELP catalog member `CORE.DMKEYS.KEYS`. To alter this catalog, do the following:

1. Invoke SAS with DMS option enabled.
2. On the command line, enter `KEYS` to display the `KEYS` window.
3. Modify the keys and issue a `SAVE` from the command line of the `KEYS` window to save the keys to `SASUSER.PROFILE`.
4. In the Program Editor, enter:

```
PROC CATALOG C=SASHELP.CORE;
CHANGE DMKEYS.KEYS=OLD.KEYS;
RUN;
```

This renames the existing `DMKEYS.KEYS` member.

5. Copy the new keys from `SASUSER` to `SASHELP`, using the following:

```
PROC CATALOG C=SASUSER.PROFILE;
COPY OUT=SASHELP.CORE;
SELECT DMKEYS.KEYS;
RUN;
```

Using the NEWS and SAS HELPCMS Files

- The NEWS File

The `NEWS` option displays the contents of a file in the `SASLOG` when the SAS System is invoked. You can uncomment the line `NEWS = "NEWS SAS *"` in the `SASV8SYS CONFIG` file, or add it to a user-defined `SASV8 CONFIG` file.

□ The SAS HELPCMS File

You can add a help file for the SAS command to the CMS HELP facility. You should copy the file, called SAS HELPCMS, to a CMS system disk so that users can issue a HELP SAS command without linking to the SAS System disk. You can add instructions to this help file about how to link to the SAS System disk (you may want to do this if you are not putting the SAS EXEC on a commonly accessed disk).

SASMAIL REXX Exec

The SASMAIL REXX exec may need to be customized with site-specific information that is pertinent to your mail system. Comments at the beginning of the exec guide you through the customizations.

Sample Libraries

The sample programs are in CMS macro libraries:

SAMPBASE	MACLIB	sample library for the Base SAS product
SAMPSTAT	MACLIB	sample library for the SAS/STAT product
SAMPETS	MACLIB	sample library for the SAS/ETS product
SAMPGIS	MACLIB	sample library for the SAS/GIS product
SAMPGRP	MACLIB	sample library for the SAS/GRAPH product
SAMPIML	MACLIB	sample library for the SAS/IML product
SAMPOR	MACLIB	sample library for the SAS/OR product
SAMPQC	MACLIB	sample library for the SAS/QC product
SAMPCONN	MACLIB	sample library for the SAS/CONNECT product
SAMPShar	MACLIB	sample library for the SAS/SHARE product
SAMPINTR	MACLIB	sample library for the SAS/IntrNet product (experimental)
SAMPORAC	MACLIB	sample library for the SAS/ACCESS Oracle product
SAMPSTAT	MACLIB	sample library for the SAS/STAT product

These maclibs contain samples of SAS source code. You may want to separate the sample programs into individual CMS files. The UNMACLIB EXEC, provided on the installation tape, will do this for you. See the appendix "Maclibs on the Installation Tape" on page 109 for information about this exec.

Some of the members of the maclibs use supplied catalogs. These catalogs are shipped as members of the SAMPPIO library. All members of this library have a filetype of SAMPPIO. Some products also have sample SAS data sets.

Defining Printers

SAS predefines several generic printers for use with ODS or Universal Printing. These generic printers write PostScript, PCL, or PDF output to CMS disk files. You may wish to create additional printer definitions that correspond to specific printers at your site, so that you can send output directly to the printer without the intermediate step of writing to a disk file. Use PROC PRTDEF to do this.

PROC PRTDEF is documented in the online help. In the material below, we discuss only CMS specifics for the most commonly used cases. To set up printer definitions for all users at your site, obtain write access to the SASHELP library and run PROC PRTDEF with the LIBRARY=SASHELP option.

If you have installed SASHELP in a saved segment, re-save the segment after running PROC PRTDEF with this option.

CMS specifics for variables needed by PROC PRTDEF

DEVICE -- For printing to a CMS disk file, use a device of DISK. For printing directly to a printer use a device of PIPE or PUNCH. (Other devices are also supported for specialized purposes.)

DEST -- For a DISK device, DEST is the name of the CMS disk file. For a PIPE device, DEST is SASPRT with optional arguments. SASPRT is further documented below. For a PUNCH device, DEST is blank.

HOSTOPT -- For a DISK device, HOSTOPT should be *recfm v*. For PIPE or PUNCH, HOSTOPT should be blank.

PROTOCOL -- For producing output in ASCII, PROTOCOL is *Ascii*. For EBCDIC output, PROTOCOL is *None*. It is preferable to produce output in ASCII to minimize translation problems.

TRANTAB -- For ASCII output, TRANTAB should be *SASGTAB0*. For EBCDIC output, TRANTAB should be blank.

LRECL -- For a PUNCH device, LRECL must be *80*. For DISK or PIPE, LRECL can be defaulted.

Considerations for printing disk files

If you write PostScript or PCL output to a disk file, you may be able to print the file by using a tool such as the PPS command provided with IBM's RSCS V3R2, or the LPR command provided with IBM's TCP/IP for VM.

For transferring a disk file to another platform, you can FTP an ASCII file in BINARY mode, or an EBCDIC file in ASCII mode.

Sending output directly to a printer

SAS provides a CMS Pipelines command named SASPRT to support printing PostScript or PCL output directly to a network printer if you have IBM's RSCS V3R2 installed. PostScript files (ASCII or EBCDIC) can be printed to an RSCS LPR link that uses the LPRXPSE exit, and PCL files (ASCII only) can be printed to an RSCS LPR link that uses the LPRXONE exit.

To use the SASPRT command, define a SAS PostScript or PCL printer with a device type PIPE. The DEST value is the SASPRT command line, including any options. The

options that should be specified in any particular case will depend on your RSCS configuration.

Options can be passed to SASPRT from three sources:

1. Defaults can be saved in the GLOBALV group PPS. Use IBM's PPS command (provided with RSCS V3R2) to set these defaults.
2. Options can be specified on the SASPRT command line in the DEST value that is part of the printer definition.
3. If the NICKNAME option is specified either via GLOBALV or the command line, options will be retrieved from the nickname entry in RSCS NAMES.

When an option is specified in more than one place, a command line option overrides a GLOBALV option, and a GLOBALV option overrides an RSCS NAMES option.

The following are supported options:

FORM	PREFIX	SUFFIX
HOSTID	PRINTER	TAG
HOSTNAME	PRINTQUEUE	TCPXLBIN
NICKNAME	RSCS	
NODE	SEP	

These options have the same meaning as in the PPS EXEC that is provided by IBM as part of RSCS V3R2. Consult the VM/RSCS Operation and Use manual for documentation.

Here is a sample job showing the definition of three printers that are assumed to be configured in RSCS, where *LPRLINK* is the name of an LPR link that uses the LPRXFORM exit to identify printers.

```

data names;
length name $8 desc $32;
input name $ desc $&;
datalines;
hpljno01 HP LaserJet number 1
hpljno02 HP LaserJet number 2
hpljno03 HP LaserJet number 3
;

data printers;
set names;

model      = 'PostScript Level 1 (color)';
device     = 'PIPE';
dest       = 'sasprt printer lpmlink form ' || name;
protocol   = 'ascii';
trantab    = 'SASGTAB0';

proc print;run;

```

```
proc prtdef list replace data=printers;
run;
```

Alternatively, you could define just one generic PostScript printer with `device='PIPE'` and `dest='sasprt printer lprlink'` and require each user to set a personal default printer via the `FORM` option of the `PPS` command.

Using the PUNCH device for printing

Here is a sample job showing the SAS definition of a printer named 'pspunch' that uses the `PUNCH` device type:

```
data punch;
  name      = 'pspunch';
  desc      = 'Virtual punch';
  model     = 'PostScript Level 1 (color)';
  device    = 'PUNCH';
  dest      = '';
  protocol  = 'Ascii';
  trantab   = 'SASGTAB0';
  lrecl     = 80;
run;

proc prtdef list replace data=punch;
run;
```

To use the `PUNCH` device, users must explicitly issue appropriate `SPOOL` and `TAG` commands prior to printing. Depending on your `RSCS` configuration, you may require options that can not be specified via `SPOOL` and `TAG`. Use the `SASPRT` pipeline if these considerations make the `PUNCH` device disadvantageous.

Here is an example of using the `PUNCH` device to print PostScript output:

```
options printerpath=pspunch;

cms cp spool punch to rscs form hpljno02;
cms cp tag dev punch mynode lprlink;

ods printer;
proc print;run;
ods printer close;
```

Chapter 3, Installing the CMS SAS® System in Saved Segments

This section describes how to install and tune CMS SAS software in saved segments. This is an optional, but strongly recommended final step of the installation. Before carrying out any of the steps described here, you should have completed all other installation steps and verified that the installed SAS System is functional.

The information in this section is for the SAS Installation Representative and the VM Systems Programmer.

All the tools that are required for installing the SAS System in saved segments are placed on the SAS System disk by the installation procedure (GETSAS). You must have at least read-access to the files on the SAS System disk.

Overview

Saved segments are a feature of the VM Operating System that lets many users share the same data in real storage. This sharing conserves real storage, thus reducing the rate of paging and the amount of paging storage required. When a product such as the SAS System is installed in saved segments, many users can share a single copy of the product code; each user can immediately access the code without having to read it from disk; and, since the saved segment can reside outside the user's virtual machine, required virtual machine size can be reduced. Without saved segments, every user must load a separate copy of the code into their own virtual machine. The more concurrent executions of SAS there are at your site, the greater the benefits of running SAS from saved segments.

We strongly recommend that the SAS System be installed in saved segments. Here is a brief listing of the benefits for system performance:

- Reduced system paging
- Reduced SAS System initialization time
- Reduced I/O to the SAS System disk
- Reduced virtual machine size requirements.

These can result in improved VCPU and execution times for all applications at your site.

You can also install **SAS libraries** and the **SAS message database** in saved segments. The advantages of doing so are similar to the advantages of installing the SAS System code in saved segments. In the case of SAS libraries, there is the additional advantage that you do not need to access the minidisk or SFS directory where the original copy of the library is stored. However, you need to take into account that the copy in the saved segment is read-only. Libraries that are good candidates for

putting into saved segments include those that are read frequently (especially by multiple users) and updated infrequently (or at regularly scheduled intervals). We strongly recommend installing the `SASHELP` library in a saved segment.

There are five steps in installing the SAS System in saved segments. They are summarized here, and detailed below. A *Procedure Checklist* is also provided to list each specific operation that you must carry out.

1. Planning for the Saved Segments

You will need to decide what pieces of the SAS System will be included in segments, where the segments will reside in virtual storage, and what the segments will be named.

2. Defining the Saved Segments

You will set parameters in a `CMSSEGS` control file and execute the `CMSSEGS` command to generate another set of control files.

3. Creating the Saved Segments

You will execute the `DEFXSAS EXEC` generated by the `CMSSEGS` command. `DEFXSAS` will issue the `DEFSEG` commands to create skeleton segments.

4. Storing and Saving the SAS System in the Saved Segments

You will execute the `SASCSEG` command, which will load the SAS System into the newly created segments and issue the `SAVESEG` command to save them.

5. Setting Option Defaults to use the Saved Segments

You will set options in `SASV8SYS CONFIG` so that the saved segments will be used by default.

Step 1: Planning for Installing the SAS System in Saved Segments

There are three major decisions to be made during the planning phase of installing the SAS System in saved segments. Your installation tape includes a configuration file (`SAS CMSSEGS` which will be described in more detail below) that provides a particular set of defaults for some of these decisions. We recommend using this default set unless some special situation requires doing otherwise. You should copy it to your A disk, and use it as the starting point for your customizations. A *CMSSEGS Worksheet* for recording your decisions is at the end of this section.

A. *What pieces of the SAS System are to be installed in saved segments?*

The SAS System consists of multiple products. All customers receive the Base SAS portion of the system, but the complete set of software that you receive depends upon which of the optional products, if any, you licensed. The

instructions in this section include the information necessary to install all SAS products into segments. However, the procedure works correctly for any subset of products.

B. What will the addresses of the saved segments be?

You must choose the virtual address at which each segment will be loaded. To make these choices you will need to know how large your saved segments will be. You can determine their sizes by selecting arbitrary starting addresses and doing a trial run of the CMSSEGS procedure described below.

Segments do not necessarily need to reside above virtual machines that will use them. If you anticipate that some of your users' virtual machines may overlap the SAS saved segments, you need to take the following considerations into account.

The SAS System must be able to issue a `SEGMENT RESERVE` command for the address space of each segment. For this command to succeed when the address space of some of the saved segments would lie within the virtual machine, it is necessary that none of the address space is already in use for another purpose, and that no segment lies partly inside and partly outside the virtual machine.

It is possible to have the `SASxMAP1` and `SASxSUPR` segments both within the virtual machine, but it is not possible to have `SASxMAP1` within the virtual machine and `SASxSUPR` outside. Since CMS uses storage at the top of the virtual machine, it is generally necessary to define your virtual machine size at least 1M larger than the ending address of the highest segment.

Saved segment address ranges must not overlap those of any other saved segments that will be used during a SAS session. For this reason, consultation between the SAS Installation Representative and the VM Systems Programmer may be necessary to determine suitable address ranges.

The following are examples of other products whose segments you might need to take into consideration:

- CMS itself
- a sort package (if in segments)
- ISPF (if you use ISPF in conjunction with the SAS System)
- VSAM (if you use VSAM files within the SAS System)
- GDDM (for certain graphic devices)
- DB2 VM (if you are licensed for SAS/ACCESS to DB2 VM)

C. What will the segments be named?

It is often desirable to have more than one version of the SAS System installed in saved segments. For example, you might install a test version of a new release of the SAS System so that you can test it without interrupting users of the production system. Since any one user can only run one version of the SAS System at a time, it does not matter whether or not different versions have their segments defined at overlapping addresses. But since each segment must have a

name that is unique system-wide, each version of the SAS System must have uniquely named segments.

To accomplish this, each SAS saved segment has a series identifier, a single letter or digit that uniquely identifies the series of segments as being different from any other existing version of SAS software. At segment definition time, this series letter is specified in the CMSSEGS file, or is prompted for by the CMSSEGS command. When you run the SAS System, you specify which series is used by the SERIES= option on the command line, in a SASV8SYS CONFIG file, or in a SASV8 CONFIG file.

In this chapter, where filenames containing an 'x' are mentioned, the 'x' represents a particular series letter. For example, the file referred to as DEFxSAS EXEC will actually be generated as DEFASAS EXEC for series A, DEFBSAS EXEC for series B, etc. Similarly, a segment name such as SASxMAP1 will actually be generated as SASAMAP1, or SASBMAP1, etc.

CMSSEGS Worksheet:

Select values for each of the following parameters to be coded in your CMSSEGS file. Full details on each parameter, as well as additional optional parameters, can be found in a later section.

SERIES _____ Series letter to be used in segment names

LOADADDR _____ What is the starting address for the SAS System code segments?

If the message database is being saved in a segment, then the following two parameters must be supplied:

MSGDB _____ The filename and filetype (and optionally, filemode) of the message database disk file. The filename and filetype will normally be MSGDB MSG.

MSGLADDR _____ The starting address for the message database segment

If you choose to save the message database in a DCSS, specify the actual address for MSGLADDR. If you choose to save the message database in a member saved segment, specify 0.

If the SASHELP library is being saved in a segment, then the following two parameters must be supplied:

LIBLADDR _____ The starting address for the SASHELP library segment.

If you choose to save the SASHELP library in a DCSS, specify the actual address for LIBLADDR. If you choose to save the SASHELP library in a member saved segment, specify 0.

LIBRARY SASHELP. See the detailed description of CMSSEGS parameters.

Step 2: Defining the Saved Segments

Execute the CMSSEGS filename command, where filename is the filename of your customized CMSSEGS file. In order to run CMSSEGS successfully, you must have the SAS System disk accessed, and you must have a Read/Write disk or SFS directory accessed as A.

When you are defining segments for the SAS System, not just SAS libraries, the CMSSEGS command produces six files on your A disk. We recommend that you preserve these files on the SAS System disk even after installation is complete. (We

suggest using the SAS System disk as your A disk when running CMSSEGS.) SAS Technical Support may request information from these files if you require assistance.

When you run CMSSEGS, you may see messages about members not found. These messages pertain to products that you have not installed, and can be ignored.

CMSSEGS reports the total number of pages that will be saved in the segments. You can use this figure to estimate the size of the NSS files that will be generated. On a small system, the NSS files could take up a significant portion of the system spool space, and you might need to increase your spool space. You should evaluate the impact of the NSS files on your system's spool capacity before proceeding.

CMSSEGS also reports the virtual machine size required for saving the segments. You will need to define your virtual storage to at least this size in order to run the SASCSEG command. This size estimate is an absolute minimum; depending on your environment, you could have storage conflicts that would require an even larger virtual machine size.

Files generated by CMSSEGS command

The following files are generated by the CMSSEGS command:

1. DEFxSAS EXEC

where 'x' is your chosen series letter. This is an exec that will issue the required DEFSEG commands.

2. SASxMAP1 MODULE

contains data that will be loaded into a saved segment

3. SASCSEGS LOADINFO

control file required by the SASCSEG6 command

4. CMSSEGS MSORTED

work file used only by CMSSEGS

5. CMSSEGS MEMBERS

work file used only by CMSSEGS

6. CMSSEGS MAP

contains a map of the segments

If you are defining segments only for SAS libraries or the message database, CMSSEGS produces only two files, DEFxSAS EXEC and SASCSEGS LOADINFO (for which you can choose alternate names; see the Reference section below for the

CMSSEGS control file statements EXECNAME and LOADINFO). We also recommend preserving these files.

Step 3: Creating the Saved Segments

To create the saved segments, complete the following steps:

1. Run `DEFxSAS QUERY` to determine whether segments with the chosen name already exist.
2. Run `DEFxSAS PURGE` to delete old segments. This is not required, since creating new segments will automatically purge old segments having the same names. However, if you are changing the structure of your segments, you may be creating a different set of segment names so you should explicitly purge the old segments to avoid leaving orphaned segments on the system. Note: this means that you will `DEFxSAS PURGE` using your *old* `DEFxSAS EXEC` before you run `CMSSEGS` to define the new segments and create a new `DEFxSAS EXEC`.
3. Run `DEFxSAS DEFINE` to define skeleton segments. This is required.
4. Again run `DEFxSAS QUERY` to verify that the segments have been correctly created.

Note that the `DEFxSAS EXEC` issues `CP DEFSEG`, `CP QUERY NSS`, and `CP PURGE NSS` commands, which can be successfully executed only by an authorized userid. Normally CP privilege class E is required, unless privilege classes have been restructured at your installation. You may need to request your VM Systems Programmer to execute this step if you do not have a userid with the necessary privilege class.

Step 4: Storing and Saving the SAS System in the Saved Segments

This step can only be executed by a userid with the privilege class required to execute the `SAVESEG` command. Normally, the required privilege class is E. You may need to ask your VM Systems Programmer to execute this step if you do not have a userid with the necessary privilege class.

Ensure that you have enough virtual storage to contain the storage for the segments within your virtual machine. If necessary, issue the command `CP DEFINE STORAGE ___M`, specifying a value that will make your virtual machine large enough to include all the segments that you want to save. You should ensure that you have at least 1M of storage above the highest segment.

If you do change your virtual machine size with the `DEFINE STORAGE` command, you will then need to re-IPL CMS. In any case, we recommend re-IPLing CMS to reduce the likelihood of storage conflicts.

Ensure that you have all the same minidisks and/or SFS directories accessed in the same order as when you ran `CMSSEGS`.

Execute the `SASCSEG` command. This command will load data into storage and issue the `SAVESEG` command to save the segments. By default, `SASCSEG` processes the `SASCSEGS LOADINFO` file. If you have generated your `LOADINFO` file with a different filename, specify the filename on the `SASCSEG` command.

If you now issue the `DEFXSAS QUERY` command, you should see output for the new segments.

The saved segments are now ready to use. Redefine your virtual machine to its normal size, re-IPL CMS, and invoke SAS with the `SSEG` and `SERIES` options to test them.

Step 5: Setting Option Defaults to use the Saved Segments

Two system options control the use of saved segments, `SSEG` and `SERIES=`. If the `SSEG` option is specified, the SAS System will execute from code in saved segments, will use the `SASHELP` library from a saved segment, and will use the message database from a saved segment. The `SERIES=` option specifies which segment series is used.

These options can be specified either at invocation or in a configuration file. We recommend, once you have verified that the installation in segments is working correctly, that you specify these options in `SASV8SYS CONFIG`. For example, if you have chosen "A" as your series letter, your `SASV8SYS CONFIG` might include:

```
SSEG
SERIES=A
```

If you temporarily need to force the SAS System not to execute from segments, specify the `NOSSEG` option. `NOSSEG` disables the use of saved segments that are loaded at SAS initialization: the SAS System segments, the `SASHELP` library, and the message database. It does not affect the ability to use user libraries in segments.

If you specify `NOSSEG`, or if you have chosen not to install the `SASHELP` library in a saved segment, the `SASHELP` system option must specify the disk-resident location of the `SASHELP` library. Normally, the correct specification will be

```
SASHELP=*
```

If you specify `NOSSEG`, or if you have chosen not to install the message database in a saved segment, the `MSG` system option must specify the disk-resident location of the message database. Normally, the correct specification will be

```
MSG=*
```

We recommend including the specifications of SSEG, SERIES=, SASHELP=, and MSG= in the SASV8SYS CONFIG file. For more information on these options, refer to the chapter "System Options" in *SAS Companion for the CMS Environment*.

To cause a user library in a saved segment to be used, specify the SEGMENT=YES option on the LIBNAME statement. For more information, refer to the description of the LIBNAME statement in *SAS Companion for the CMS Environment*.

Procedure Checklist:

✓ ***Planning for Installing the SAS System in Saved Segments.***

Choose a filename for your CMSSEGS control file. We suggest using the name of the sample file, with the series letter appended, e.g., SASA CMSSEGS.

Access the SAS System disk.

Copy the prototype CMSSEGS file to your A disk, using your chosen filename, e.g., COPYFILE SAS CMSSEGS B SASA = A.

Complete the CMSSEGS worksheet, then edit your CMSSEGS file and change parameters using information recorded on the worksheet.

✓ ***Defining the Saved Segments***

Execute the command CMSSEGS filename, where "filename" is the filename of your customized CMSSEGS file.

✓ ***Creating the Saved Segments***

Execute the following DEFxSAS commands, where DEFxSAS is the exec created by CMSSEGS:

```
DEFxSAS QUERY
DEFxSAS PURGE (optional, see text)
DEFxSAS DEFINE
DEFxSAS QUERY
```

✓ ***Storing and saving the SAS System in the Saved Segments***

If necessary, issue the command CP DEFINE STORAGE ___M

Recommended: issue the command CP IPL CMS (or whatever CMS is named at your installation), then re-access your minidisks and/or SFS directories in the same order as when you ran the CMSSEGS EXEC.

Execute the SASCSEG command.

Execute the DEFxSAS QUERY command.

✓ ***Setting default options***

Enter the options

```
SSEG
SERIES=value
```

in your SASV8SYS CONFIG file.

Sample CMSSEGS file

The following is an example of a completed CMSSEGS file. For this example, assume that the following decisions have been made:

- ❑ The series letter is B.
- ❑ The segments will load at address 2000000 hexadecimal (32M).
- ❑ The message database will be stored in a member saved segment in the supervisor segment space.
- ❑ The message database exists in a file named MSGDB MSG.
- ❑ The SASHELP library will be stored in a member saved segment in the supervisor segment space.

Now that these choices have been made, copy the prototype SAS CMSSEGS file from the SAS System disk to your A disk, and rename it SASB CMSSEGS: COPYFILE SAS CMSSEGS * SASB CMSSEGS A. Then, XEDIT SASB CMSSEGS A, filling in the site parameters as shown below.

```

* This file contains information needed to generate saved segments
* in which to save SAS Release 8.2 code.
* SASHELP and messages are member saved segments in the supervisor segment space.

SAY    Generating segments from SASB CMSSEGS Control file
*
*--- general information
*
SERIES    B                ; which series to generate (1 character)
*
*--- at what address do we start loading things
*
LOADADDR  2000000          ; 2000000 equals 32 Meg load point.
*
*--- the list of LLIST files to process
*
LLIST     ALLSAS           ; loadlist file for all of SAS
*
*--- specifications for the message segment
*
MSGLADDR  0                ; Use 0 to save SASxMSG as a member saved segment. Use a
*                          ; specific address to save it as a DCSS
MSGDB     MSGDB MSG       ; Name of the message database file
*
*--- specifications for the help library segment
*
LIBLADDR  0                ; Use 0 to save SASxHELP as a member saved segment. Use a
*                          ; specific address to save it as a DCSS.
*
LIBRARY   SASHELP         ; name of the SASHELP LIBRARY

```

CMSSEGS File for Saving Only User Libraries

A CMSSEGS file that is only for loading libraries and/or the message database differs from a CMSSEGS file that is used for loading SAS System code in that it does not contain LLIST statements or the LOADADDR statement. A CMSSEGS file that is only for loading user libraries (not including the SASHELP library) does not require a SERIES statement.

In this example, user libraries are saved in two ways:

- ❑ User libraries ABC and DEF are saved as member saved segments in segment space XYZSPACE, which will begin at address 0C000000.
- ❑ User libraries DC01 and DC02 are saved as DCSS. The DC01 segment will load at 0C400000, and the DC02 segment will load at the lowest megabyte boundary above the end of DC01.

The EXECNAME MYLIBS statement causes the generated exec to be named MYLIBS EXEC instead of the default DEF_SAS EXEC.

The LOADINFO MYLIBS statement causes the generated LOADINFO file to be named MYLIBS LOADINFO instead of the default SASCSEGS LOADINFO. To save the segments, specify this filename on the SASCSEG command:

```
SASCSEG MYLIBS
```

The statements found in this last example are explained in more detail in the Reference section, below.

```
* This file contains information needed to generate saved segments
* in which to save SAS Release 8.2 libraries.
*
EXECNAME MYLIBS
LOADINFO MYLIBS
*
LIBLADDR 0C000000
LIBRARY  ABC * ABC XYZSPACE
LIBRARY  DEF * DEF XYZSPACE
*
LIBLADDR 0C400000
LIBRARY  DC01
LIBRARY  DC02
```

Reference

This section provides information for more advanced customization of the saved segment installation.

Types of Saved Segments

The SASxMAP1 and SASxSUPR segments are saved as member saved segments in the SASxMAP segment space.

You can choose to save the message database either in a member saved segment in SASxMAP or in a DCSS.

You can choose to save SAS libraries in member saved segments in SASxMAP, member saved segments in user-defined segment spaces, or in DCSS.

Tailoring What Goes Into Segments

You should save the SAS System using the supplied configuration at least once. You may then decide that you want to remove some less frequently used products or procedures from segments due to either address range or disk space constraints.

To omit items from segments, make a copy of the ALLSAS LLIST file and, in the copy, comment out one or more procedures or functions to prevent that particular piece of code from being loaded into segments.

We recommend that you proceed carefully, as a mistake can cause a dramatic decrease in performance. To help you determine what to omit, you can use the SGSTAT option to collect usage information on the various pieces of code that make up SAS software.

Collecting Segment Usage Data Using the SGSTAT option

If you want to reduce the amount of storage occupied by SAS saved segments, you must decide which pieces of code, or which products, can be omitted from segments without causing performance degradation. SAS software provides a performance and tuning option, SGSTAT, to help collect this information. When you invoke the SAS System with the SGSTAT option, the system sends information via SMSG to a disconnected virtual machine named SAS6ACT. You can specify the SGSTAT option on the command line when invoking the SAS System, in a SASV8 CONFIG file, or in the SASV8SYS CONFIG file.

You must define the SAS6ACT virtual machine and provide a tool to run in it if you want to collect this information. You can then analyze the information to see which routines are most heavily used but not in segments, and which routines are in segments but not used often.

Three type of records are sent, `FOUND`, `NOTFOUND`, and `OVERLAY`. Their formats are as follows:

- ❑ `FOUND` `segname` `funcname`
- ❑ `NOTFOUND` `funcname`
- ❑ `OVERLAY` `segname` `funcname` `usagecount`

`FOUND` records a function that was found in a segment, and the segment name in which it was found. `NOTFOUND` records a function that was not found in a segment. `OVERLAY` records a function that was found in a segment but could not be used because some other SAS segment was already attached (with a non-zero usage count) at the same address. (Note: Starting with Version 7, `OVERLAYS` are no longer supported.) By analyzing the data in these records, you can determine which functions are heavily used and which are less used. The simplest strategy is to try to fit as many heavily used functions into segments as possible.

There is some overhead in sending this data using `SMMSG`; also, it generates a body of data very quickly. We recommend that the option only be turned on for the purpose of collecting data and then turned off. The easiest way to turn the option on is to add it to selected `SASV8 CONFIG` files or to the system `SASV8SYS CONFIG` file.

CMSSEGS Control Files

Customization parameters for defining CMS SAS saved segments are entered in control files identified by having a filetype of `CMSSEGS`. `CMSSEGS` control files are processed by the `CMSSEGS` command. Your installation tape contains the SAS `CMSSEGS` file, which loads the entire SAS System into saved segments

A `CMSSEGS` file contains a series of statements or comments:

- Statements must start with one of the keywords documented below.
- Comments begin with an `*`, or are entirely blank.
- A statement can be followed by a semicolon (`:`). The remainder of the line will be considered a comment.

If you prefer, you can enter a `?` as the parameter for all keywords other than `LLIST` or `LIBRARY`. You will then be prompted to supply the value when you run the `CMSSEGS` command. The provided prototype `CMSSEGS` file has `?` coded for most parameters, so you could use it without modification. However we recommend that you code your actual settings in a private `CMSSEGS` file as a means of preserving a record of your choices.

When `CMSSEGS` prompts for a value, you can terminate `CMSSEGS` with any of these replies: `bye`, `end`, `quit`, or `stop`.

□ CMSSEGS keywords

The following is the complete list of CMSSEGS keywords:

- **SERIES** a single letter or digit that will be used in the names of SAS saved segments. This allows you to define more than one series of segments. If you specify a letter or digit that is already in use for a series of SAS saved segments, you will replace that series. Be extremely careful not to do this unintentionally.

The SERIES letter is also used in the filenames of some files generated by the CMSSEGS command, for example DEFxSAS EXEC, where the "x" is replaced by your series letter. After you run CMSSEGS, issue the DEFxSAS QUERY command to determine whether segments already exist with your chosen series letter.

A series letter is not required if you are only saving SAS libraries (other than SASHELP, which does require a series letter). In this case the series letter appears only in the name of DEFxSAS EXEC, and defaults to "_" (underscore).

- **LOADADDR** specifies the address (in hexadecimal) at which the SAS System segments will be loaded.
- **LLIST** specifies the name of an LLIST file that will be processed. There can be any number of LLIST statements.
- **CMS** specifies a CMS command to be executed when CMSSEGS is invoked. One use for this feature might be to send a signal message to another virtual machine.
- **SAY** echoes the remainder of the statement to the console when CMSSEGS is invoked. If the string &THISFILE appears in the statement, it will be replaced with the actual name of the CMSSEGS file when the statement is echoed to the console.
- **LOADINFO** specifies the name of the LOADINFO file generated by CMSSEGS. The default is SASCSEGS. We recommend using the default when you are saving the SAS System but using an alternate name if you are only saving libraries. When you run the SASCSEG command, specify the filename of the LOADINFO file on the command line.
- **EXECNAME** specifies the name of the exec generated by CMSSEGS. The default is DEFxSAS. We recommend using the default when you are saving the SAS System but using an alternate name if you are only saving libraries.

- **MSGLADDR** specifies the address at which the message database segment will be loaded. If MSGLADDR is 0 and the CMSSEGS file also contains LLIST statements (i.e., you are also saving SAS System code) then the message database will be saved as a member saved segment in the supervisor segment space. If you specify an explicit address for MSGLADDR then the message database will be saved as a DCSS at that address. If the CMSSEGS file contains no LLIST statements, the message database must be saved as a DCSS and you should specify a value for MSGLADDR because CMSSEGS may have no basis for selecting a reasonable default.
- **MSGDB** specifies the filename and filetype, and optionally filemode, of the disk file containing the message database.
- **LIBLADDR** Each LIBLADDR statement specifies a starting point for load addresses of libraries to be specified on subsequent LIBRARY statements. There can be multiple LIBLADDR statements if multiple starting points need to be used.

If LIBLADDR is specified as '0', then the segment defined by the next following LIBRARY control record will load at the next available address. Also, if more than one LIBRARY control record follows a LIBLADDR control record, the segments defined by the second and subsequent LIBRARY control records will load at the next available address after the end of the segment specified by the previous LIBRARY control record.

In this context, "next available" means

1. For the first member of a segment space or for a DCSS, the next megabyte boundary above the end of the previously processed segment ('0' if no previous segment)
2. For non-first members of a segment space, the next page boundary above the end of the preceding member

Libraries are not necessarily assigned addresses in the same sequence in which they appear in the CMSSEGS file. All member saved segments of the supervisor segment space are mapped first, then the overlay segments if any, then all remaining library segment spaces, then the message DCSS, then all library DCSSs.

- **LIBRARY** filetype <filemode <segmentname <segmentspace>>>

The LIBRARY statement takes from one to four parameters. All but the first parameter are optional but any optional

parameters must be specified in the sequence shown and none can be skipped.

1. Filetype is the filetype of a SAS library to be saved in a segment. Filetype is the only required parameter.
2. Filemode is the filemode where the SAS library files are accessed (the default is the first accessed filemode where a file with the specified filetype is found). You can specify "*" as a placeholder to select the default explicitly.
3. Segmentname is the name of the segment in which the library will be saved. The default is the same as the filetype. LIBRARY SASHELP is a special case; the segment name will be SASxHELP whether segmentname is specified or not.
4. Segmentspace is the name of a segment space in which the library segment will be a member saved segment. The default for segmentspace is explained in the following paragraphs.

A library can be saved in one of three segment types: a member saved segment in the SAS supervisor segment space, a member saved segment in a user-specified segment space, or in a DCSS.

If segmentspace is specified on the LIBRARY statement, then the library segment will be a member saved segment in that space.

If segmentspace is not specified, follow the steps below:

1. If the last preceding LIBLADDR statement specified a non-zero address, or if there are no LLIST statements in the CMSSEGS file, then the segment will be a DCSS.
2. If the last preceding LIBLADDR statement specified an address of 0, or if there is no preceding LIBLADDR statement, then the segment will be a member saved segment in the supervisor segment space.
3. If the last preceding LIBLADDR specified '?', then if the response to the prompt for an address is non-zero, then the library segment will be a DCSS, otherwise it will be a member saved segment in the supervisor segment space.

LLIST Files

Control files with filetype LLIST are used to specify which pieces of code are to be loaded into each SAS saved segment.

The contents of an LLIST file are a series of keyword records, data records, or comments. For all records, only the first 72 columns are read, and any text following a semicolon (;) is regarded as a comment. Comment records must begin in column 1 with an '*', or be entirely blank. The format of keyword and data records is given below.

Keyword records must begin in column 1 with one of the keywords documented below.

- ❑ `BOOT`
specifies the name of the "bootstrap" code and determines which "bootstrap" module will be available at SAS runtime, depending upon which environment SAS is run under. There should be only one `BOOT` record among all the `LLIST` files processed from any one `CMSSEGS` file and it should *NOT* be changed by the user.
- ❑ `SEGMENT OVERLAY=PRI.level segname`
This statement should not be changed by the user.
- ❑ `LABEL descriptive text`
specifies a description of the segment being created.
- ❑ `ALIGN number`
creates enough space in the current segment to force the next item to begin at the specified boundary.

Data records must begin in a column greater than 1 and have the following format:

```
filename LOADLIB filemode membername (share-status)
```

This specifies the filename, filetype, and filemode of a `LOADLIB` and the name of a member of that `LOADLIB` to be loaded into the segment specified by the last previous `SEGMENT` statement. The `share-status` must be (S) or the member will not be loaded.

Chapter 4, Maintaining the SAS[®] System Under CMS

- Part I Updating Product Licenses Using SETINIT
- Part II Applying Maintenance to SAS Software Products
- Part III SAS Notes and Zaps
- Part IV Installing Additional SAS Software Products

Part I, Updating Product Licenses Using SETINIT

In order to run each software product that is part of the SAS System, your installed SAS System must have up-to-date information about your licensing agreement with SAS. This licensing information in SAS software is called SETINIT information. SAS software will not run without valid SETINIT information.

Running PROC SETINIT will update your SETINIT information in the SAS System. Your SETINIT must be updated in the following circumstances:

- Renewal of your product authorization.
- Authorization of additional SAS software.
- Change of hardware, CPU model or serial number
- Nonfunctioning SETINIT for whatever reason

Any change requests for your license parameters may either be called in or submitted in writing on your company's official stationery to our Customer Service Department.

Note: Only the authorized SAS representative should change the SETINIT information. You designated the SAS representative when you licensed the SAS System.

Running PROC SETINIT

To update your SETINIT information, complete the following steps:

1. Get write access to your SAS System disk.
2. If your new SETINIT information was not sent on a tape (you have a paper SETINIT):
 - A. Copy SETNAME SAS, which is on your SAS System disk and contains the original SETINIT information, to a file called SETNAME SASO for backup purposes.
 - B. Update SETNAME SAS with the new SETINIT information. **Be sure that the file matches the information you were sent character-for-character.** Save the new file.
3. Execute the new SETNAME SAS program by entering the following SAS command:

```
SAS SETNAME (SETINIT NOSSEG
```

This command writes a file called SETNAME SASLOG to the same disk that contains the SETNAME SAS file (or the first R/W disk if SETNAME SAS is on a disk accessed R/O). You should see the following message in the SASLOG when the SETINIT has been applied successfully:

Note: Siteinfo data have been updated.

If you find any error messages, verify the information in the SETNAME SAS file and re-execute as previously described.

4. Finally, if your site has installed the SASHELP library in segments, you will need to resave segments. Refer to Chapter 3, "Installing the CMS SAS System in Saved Segments" on page 31 for details on resaving segments.

Updating the SAS Registry

You will have additional registry components if you are licensed for any of the following products:

- AF
- CONNECT
- Graph
- IML
- INTTECH
- MDDB

Unless your tape was shipped with a paper SETINIT, these registry components will be installed automatically.

If your tape was shipped with a paper SETINIT, or if any problem prevented the automatic application of the SETINIT or the registry updates during the install, you will need to manually apply these registry updates after manually applying the SETINIT. To determine whether you have registry updates that have not yet been applied, check whether any files with filetype SASXREG have been installed on your SAS system disk. If so, you must perform the manual registry update.

To perform the manual registry update, follow these steps:

1. Make sure you have write access to your SAS System disk.
2. At the CMS Ready prompt type:

```
SAS SASREG (NOSSEG
```

This procedure may take a few minutes. After it completes successfully, you can erase the SASXREG files.

3. If your site has installed the SASHELP library in segments, you will need to resave segments. Refer to Chapter 3, "Installing the CMS SAS System in Saved Segments" on page 31 for details on resaving segments.

For More Information

If you are licensing additional SAS software, please refer to Chapter 4, Part IV, "Installing Additional SAS System Products" on page 54 for more information.

If you have questions about your SETINIT data (for example, expiration date, CPU serial number, etc.), please call our Customer Service Department at (919) 677-8003 between 9:00 a.m. and 8:00 p.m. Eastern Time.

If you encounter problems applying your SETINIT, please call our Technical Support Division at (919) 677-8008 between 9:00 a.m. and 8:00 p.m. Eastern Time. Ask the Technical Receptionist for a CMS consultant. Please have your site number ready when you call.

Part II, Applying Maintenance to SAS Software Products

Maintenance is delivered on a MAINT tape, which contains all available maintenance for all products that have been licensed for your site. When you install a MAINT tape, replacement or delta files are loaded to disk for all products installed at your site. A MAINT tape alone does not represent a full SAS System. Instead, it contains only those files that have received changes since the production release of the product.

Maintenance for LOADLIBS and Catalogs is shipped as delta files on the MAINT tape. The other files contained on the tape will be replacement files. The install will merge the new executables into their proper LOADLIBS and perform a compress at the conclusion of the merge process. Catalog processing will check for any user-modified catalogs and write them to a library called SETASID. At the completion of the maintenance installation, check the SETASID library and reapply any local modifications.

The maintenance on the tape is cumulative. It is not necessary to install previous maintenance in order to update to the current maintenance release.

Note: The MAINT tape should be installed over a **copy** of the existing Release 8.2 of the SAS System. During the maintenance install process, files loaded from the tape will replace files on the specified minidisk/shared file system directory and new files may be added. Once validation jobs have been run and you are satisfied with the behavior of the new maintenance release, you can make it the production version.

When setting up the new disk to house a copy of your existing Release 8.2 of the SAS System, to which the maintenance will be applied, remember to factor in 200 additional cylinders. This is the minimum needed to accommodate the LOADLIB COMPRESS process.

To determine disk space requirements, decide what sections and products you will unload from the tape. Run GETSAS without loading files and let it calculate disk space requirements.

If an error occurs during the installation process, you must correct the problem first then restart the procedure from the beginning. Do not attempt to manipulate any file that came off the MAINT tape with the error condition, because GETSAS will apply the maintenance properly when it is rerun.

Part III, SAS Notes and Zaps

Using SAS Notes

SAS Usage Notes will no longer be shipped with install media. Instead, the SAS Web Site on the Internet can be used to access SAS Usage Note information (now termed simply "SAS Notes" or "Technical Notes") that is updated **daily**. This immediate access to the most current information made the shipping of SAS Notes on tape obsolete.

You can access the most current SAS Notes by going to `http://www.sas.com`, clicking on the `Technical Support` link, and then clicking on `Technical Notes`.

You can also do a keyword search on SAS Notes by using the search dialog box at the top of the `Technical Support` home page.

If you do not have web access, our `Technical Support` staff is on hand to help you Monday through Friday, 9 a.m. – 8 p.m. Eastern Time at 919-677-8008.

Applying ZAPs for Maintenance

SAS Institute may occasionally recommend that you apply maintenance to correct a particular problem. Sometimes, this maintenance is in the form of a `ZAP`. The `ZAP` is applied to one or more members of a `LOADLIB`.

Included in the `SUPPORT TOOLS` section of the tape is a `maclib` called `ZAP800 MACLIB`. This `MACLIB` contains `ZAPs` that fix specific problems. The members included in this `maclib` have names that begin with `Z800` and end with a 4-digit number that corresponds to a Usage Note number and the last four digits of `MODULEN` in the `USAGE.USAGE SAS` data set. For example, `Z8000003`, is a `zap` for a problem referred to in Usage Note 3 for Release 8.2 of the SAS System. **Before applying any ZAPs, read the corresponding Usage Note to see if the ZAP is applicable to your system.**

Should you need to get a `ZAP` out of the `MACLIB`, an `exec` called `UNMACLIB EXEC` is provided on the tape to remove all members at once or individual members. See the appendix, "MACLIBs on the Installation Tape" on page 109 for information on manipulating `MACLIBs`.

When it is necessary to apply a `zap`, the syntax for application can be found in the header of the `zap` file. The `CMS ZAP` command should be used. For complete information on the `CMS ZAP` command, please see the appropriate IBM documentation. An example use of the `ZAP` command is as follows:

```
ZAP LOADLIB libname1 libname2 (INPUT filename NOPRINT
```

where :

<code>libname#</code>	is the filename of the loadlibs to be zapped
<code>filename</code>	is the filename of the file that contains the zap to be applied (the filetype should be <code>ZAP</code>)
<code>NOPRINT</code>	tells the <code>ZAP</code> command not to send information on the <code>ZAP</code> application to the printer.

To apply zap maintenance, follow these steps:

1. Remove the `zaps` from the `maclib`, as described above.
2. Copy the `LOADLIB` member(s) to be zapped to a temporary `LOADLIB` using the `LOADLIB COPY` command.
3. Apply the zaps to the temporary `LOADLIB`.
4. Run your test job with the `SASLOAD=` and `NOSSEG` options set properly to access the temporary `LOADLIB`.
5. When you are satisfied that the maintenance is correct, `LOADLIB COPY` from the temporary `LOADLIB` to the production `LOADLIB` to replace the old members.
6. Resave your SAS segments if a zap has been applied to code residing in the saved segments.

Saved Segment Implications

After a `ZAP` is applied to a member of a `LOADLIB`, it no longer matches the code that is in the saved segment. This has the following consequences:

- You cannot test the change by running the version of the SAS System that is in segments.
- You must eventually re-save the code in segments to give users access to the `ZAP`.

`$NOSEGL` is an option **for testing purposes only** that allows you to force SAS software to use the specified member from a `LOADLIB` rather than from segments.

The syntax is as follows:

```
$NOSEGL=<image name>
```

You can specify only one name per `$NOSEGL` option, but you can specify multiple `$NOSEGL` options.

Once you have tested the `ZAP`, you must be sure to re-save the segments to keep the `LOADLIB` concurrent with the code in segments.

Part IV, Installing Additional SAS System Products

Note: If you are not familiar with the terms and concepts defined in “Installation Objectives” and “Tape Contents” starting on page 2 in Section I, *Introduction and Orientation*, you should read that information before continuing with this section.

What is an Add-On?

If you license an additional SAS Software product after you receive your initial SAS System, you will receive an ADDON tape that contains your new product. To request an additional SAS System product, contact SAS Customer Service at (919) 677-8003 between 9 a.m. and 8 p.m. Eastern Time.

Instructions for Installing Your ADDON Tape

Complete the following steps to install your ADDON tape:

1. To load the new software from tape to disk, follow the instructions in Chapter 2, “Installing the SAS System Under CMS” starting on page 7. The instructions for loading the tape to disk are the same for an ADDON tape as they are for a new SAS site. The “Rerunning GETSAS” section on page 19 will be particularly helpful for installing your add-on.

You can install your new SAS software directly to your production SAS System. If you do not want to do this, make a copy of your 8.2 SAS disks and install your add-on over the copy.

Some SAS software products require other SAS products in order to run. If you attempt to install a product without the other required products, GETSAS displays an error message indicating that additional products are needed. If this happens, you need to install the required product before or at the same time as the new product.

Note that the GETSAS EXEC automatically merges any autocall macro libraries for the new product into SASAUTOS MACLIB. For more information, please see “Enabling the SAS Autocall Macro Libraries” on page 21.

2. Update the SAS System SETINIT by running the SETINIT included on the update tape. The SETINIT information is in a file called SETNAME SAS, which was loaded to disk from the tape. Execute the program by entering this SAS command:

```
SAS SETNAME (SETINIT NOSSEG
```

For more information about updating product licenses, please see “Updating Product Licenses Using SETINIT” on page 48.

If the ADDON tape includes Registry product pieces, these will be merged automatically after the SETINIT is applied. If the SETINIT is manually applied, please refer to "Rebuilding the SAS Registry File" on page 49.

3. If you have a test program for the new product, run it to verify the installation. For more information about testing your installation, please see "Installation Verification Testing" on page 22.
4. If the SAS System was previously installed in saved segments, you may need to create new segment information using the CMSSEGS EXEC. For more information about this exec, see Chapter 3, "Installing the CMS SAS System in Saved Segments" on page 31.
5. Read the appropriate appendices for information about tailoring your new product. Note that not all products require post-installation modification.

Appendix A, Implementing SAS/ACCESS[®] Interface to ORACLE[®]

In this step you complete the installation of the SAS/ACCESS Interface to ORACLE.

To interface SAS/ACCESS software to ORACLE, access the production disk for the ORACLE database machine and then complete the following steps:

Oracle Releases

With every new release of ORACLE database software, the SAS/ACCESS software to ORACLE must link in ORACLE release-specific `textlibs`. The SAS/ACCESS software to ORACLE you received has ORACLE release 7.3.3.6 `textlibs` linked into the `sasora` member in the `sasaccoc` load library. If you need an earlier release of ORACLE software supported for your SAS/ACCESS to ORACLE software, contact technical support for more information.

Assigning Login Defaults

Note: This task is optional.

ORACLE allows you to specify login information several ways:

1. First ORACLE looks for information in the login string.
2. If the information is not specified in the login string, ORACLE next looks in the `LASTING GLOBALV` file. You can set your login string defaults in the `LASTING GLOBALV` file using the `ORASET EXEC`. Refer to the *Oracle User's Guide* for more information.
3. If no `LASTING GLOBALV` information for ORACLE login exists, the information is taken from the first `SYSTEM OracleID` files found on an accessed minidisk. Refer to the *Oracle User's Guide* for more information.
4. If no login information is supplied, you will receive the following error from ORACLE:

```
ERROR: ORACLE connection error: ORA-01017: invalid
username/password; logon denied.
```

Creating and Loading the Sample Tables (optional)

Refer to *SAS/ACCESS[®] Software for Relational Databases: Reference, First Edition, (Part 6, SAS/ACCESS Software: Appendices, Sample Data)* for coding examples based on sample

ORACLE tables that can be created at your site. Creating these tables will help the users at your site in learning how to use the SAS/ACCESS Interface to ORACLE product.

The program to create these sample tables is in the `SAMPLE` library member `GENDBL`. Before running `GENDBL`, you must first adapt the `DBMACS` and `DBPTMACS` macro file for your ORACLE set-up.

Appendix B, Tailoring SAS/ACCESS® Interface to DB2 Server for VM

To interface SAS/ACCESS software to DB2 for VM, access the production disk for the IBM DB2 for VM database machine and then complete the following steps:

1. Link-edit step.

SAS/ACCESS Interface to DB2 Server for VM software uses the IBM DB2 for VM resource manager stub routine, ARIRVSTC, to communicate with IBM DB2 for VM. The following step, though not required, is recommended to ensure that loading the resource manager does not conflict with the loading of other software.

Note: If you choose to do this step, you must execute this step each time a new version of IBM DB2 for VM is installed at your site.

- A. Access the IBM DB2 for VM disk (read-only) that contains the resource manager stub routine. This is a text file with a filename of ARIRVSTC and should be on the production minidisk for the IBM DB2 for VM data base machine. Refer to *IBM DB2 Server for VM System Administration* (GC09-2405-00) for additional information on the production minidisk for a data base machine.
- B. Using the following CMS command to link-edit the text file

```
LKED ARIRVSTC (RENT.
```

The resulting loadlib can reside on the A-disk or on a disk common to users of SAS/ACCESS software. Be sure to add the loadlib reference to the SASV8SYS CONFIG file with the SASLOAD= option.

2. Execution step.

The remaining steps of the installation process may require the cooperation of an IBM DB2 for VM database administrator or installer. Please contact these individuals with questions concerning any of the following steps.

- A. Make sure you have defined a virtual machine to run IBM DB2 for VM. Refer to *IBM DB2 Server for VM System Administration* for additional information on defining IBM DB2 for VM user machines.
- B. The IBM SQLINIT exec should be used to establish the name of your database. Be sure to use the PROTOCOL option for DRDA environments. Refer to *IBM DB2 Server for VM Interactive SQL Guide and Reference* (SC09-

2409-00) for additional information on the production minidisk for a database machine.

- C. You must create an access package by precompiling the SASQD8 source unloaded from the tape. The SQL preprocessor is invoked with the following command:

```
SQLPREP ASM PP(PREP=SASSQD8,USER=SQLDBA/pswd,DATE(ISO)) IN(SASSQD8 ASM fm)
```

where `fm` indicates the filemode of the SAS disk that contains SAS/ACCESS Interface to DB2 for VM software.

Be sure to specify the `SQLDBA` user and its correct password. Additional parameters for the `SQLPREP` command may be required at your installation. If you are unsure of the options you should use, talk to your IBM DB2 for VM data base administrator or refer to *IBM DB2 Server for VM Application Programming (SC09-2392-00)* for additional details on the IBM DB2 for VM preprocessor.

- D. UNLOAD the packages from a native DB2 for VM system and RELOAD them to a DRDA resource.

Note: If you are invoking the precompiler from this particular userid, `SQLDBA`, you do not need to use the preprocessor option `USERID=` and therefore this UNLOAD/RELOAD step may not be necessary.

Below is an example of the syntax and the steps necessary for an UNLOAD/RELOAD. In the example the package is called `SASSQD8` and is located on a DB2 for VM system called `DB2VM`. It is to be loaded to a DRDA resource called `DB2MVS`.

- (1) Invoke `SQLINIT` to the DB2 for VM system using the `SQLDS` `PROTOCOL`. This is necessary to perform the UNLOAD operation.

```
SQLINIT DB (DB2VM) PROTOCOL (SQLDS)
```

- (2) Issue a `filedef` to describe the file to which the package will be unloaded.

```
FILEDEF ACC DISK SASSQD8 PACKAGE A (RECFM FB LRECL
80 BLKSIZE 800)
```

- (3) Invoke the database utility `SQLDBSU` and issue the following command:

```
UNLOAD PACKAGE (SASSQD8) OUTFILE(ACC);
```

- (4) Invoke `SQLINIT` again to change the `PROTOCOL` option to `AUTO`. This is necessary to perform the RELOAD operation on the DRDA resource.

```
SQLINIT DB (DB2VM) PROTOCOL (AUTO)
```

- (5) Invoke the database utility SQLDBSU again and issue the following commands:

```
CONNECT TO DB2MVS ;  
RELOAD PACKAGE (SQLDBA.SASSQD8) REPLACE KEEP INFILE (ACC) ;  
COMMIT WORK ;
```

For more information concerning the SQLDBSU utility or DRDA, consult with your IBM DB2 for VM data base administrator, or refer to the following documents:

- *IBM DB2 Server for VM DBS Utility (SC09-2394-00)*
 - *IBM DB2 VMDatabase Services Utility for IBM VM Systems (SH09-8088-03)*
 - *IBM Distributed Relational Database Architecture Evaluation and Planning Guide (SC26-4650-01)*
 - *IBM Distributed Relational Database Architecture Connectivity Guide (SC26-4783-02)*
3. Before you can use SAS/ACCESS software, you must grant run authority for the programs the software uses. You can do this by using `ISQL` and the `GRANT` command. Specify the following:

```
GRANT RUN ON SASSQD8 TO userid
```

or

```
GRANT RUN ON SASSQD8 TO PUBLIC
```

If you have DRDA protocols, you should connect to the DRDA database system and reissue the previous `GRANT` statements.

Note: The `GRANT` command must be executed by someone with `GRANT` authority. If you are unsure of your IBM DB2 for VM authority, you should ask your IBM DB2 for VM data base administrator before attempting to issue the `GRANT` command.

Appendix C, SAS/ACCESS[®] Interface to SYSTEM 2000[®] Software

Perform the following installation steps only if you are licensed for the SAS SYSTEM 2000 product and are interested in accessing SYSTEM 2000 data through the SAS System.

1. Verify your SYSTEM 2000 release level.

You must have installed SYSTEM 2000 Release 11.6, Field Fix 1 or higher.

2. Review your shared segment assignments.

Both the SAS System and SYSTEM 2000 Multi-User software use CMS shared segments. It is important that the memory assignments be non-overlapping. You may need to adjust your memory assignments for the two systems. Information on the SYSTEM 2000 DCSS is contained in the installation instructions for the SYSTEM 2000 Multi-User Software.

3. Install and customize the SASS2K EXEC.

Use the SASS2K EXEC to access SYSTEM 2000 databases using the SAS System. This exec allocates all the files that are necessary to access the SYSTEM 2000 databases. The files are written in REXX and provided for you on the installation tape.

Note: The SASS2K EXEC supplied with this release of the SAS System may be different from the SASS2K EXEC from previous releases. Do not attempt to use an old SASS2K EXEC to invoke this release of the SAS System. If you modified an old SASS2K EXEC, be sure you transfer any important modifications to the new exec. In addition, you can not execute the SAS System from an old style exec (beginning with &CONTROL).

The SASS2K EXEC has the same parameters as the SAS exec. The exec prompts you for all additional information that is unique to SYSTEM 2000 software. If you want to bypass the prompts, you should create another exec that queues arguments to SASS2K and then calls the exec. For single-user SYSTEM 2000 execution, supply arguments for the names of the databases and the file modes for the database files, then terminate with a null argument. For multi-user execution, supply a null argument.

For example,

Single-User Mode	Multi-User Mode
MAKEBUF	MAKEBUF
QUEUE "data base1 filemode"	QUEUE " "
QUEUE "data base2 filemode"	EXEC SASS2K
...	DROPBUF
QUEUE "data basen filemode"	EXIT
QUEUE " "	
EXEC SASS2K	
DROPBUF	
EXIT	

4. Optionally, install sample data.

SAS/ACCESS Interface to SYSTEM 2000 Data Management Software: Usage and Reference, Version 6, First Edition contains a number of coding examples based on the `EMPLOYEE` data base that is delivered with the SYSTEM 2000 product. These examples help the users at your site learn how to use the SAS/ACCESS Interface to SYSTEM 2000 Software. Refer to the SYSTEM 2000 installation instructions for information on restoring this database. Other programs used as examples in the SYSTEM 2000 user's guide can also be found in the sample `maclib`, `SAMPS2K` `MACLIB`.

Appendix D, Post-Installation Setup for SAS/ASSIST® Software

An exec called ASSIST EXEC is on the tape for your convenience. This exec automatically brings up the SAS System in ASSIST mode.

Profile Changes and Enhancements

To enhance the profile feature for Query and Reporting, some changes have been made to the profile structure. A new profile option, `Query exit`, has been added to the user profile in `Type:Query`. Another new profile option, `Additional Information`, has been added to `Type:Query Manager`. This enables users, who are running queries against SAS data, to obtain customized information about the data they are accessing.

Appendix E, Post-Installation Setup for SAS/CONNECT® Software

For documentation about the configuration of SAS/CONNECT software, refer to *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software*.

Appendix F, Installing SAS/GIS[®] Census Tract Maps

This document contains instructions to guide you through your installation of SAS/GIS Census Tract Maps in Release 8.2 of the SAS System under CMS. SAS/GIS Census Tract Maps are provided on media separate from your SAS System media.

This document provides step-by-step instructions for the installation process. It contains the following sections.

- Overview and Checklist of the SAS/GIS is a concise list of the installation steps.
- Pre-Installation Considerations helps you prepare for the installation, including instructions for loading the installation aids from tape to disk.
- Loading the SAS/GIS from Tape to Disk explains how to use the installation aids (specifically the `GETGIS EXEC`) to load the SAS System to disk.

Installation Requirements

In order for the SAS/GIS Census Tract Maps to be installed properly, you must have read/write access to the Release 8.2 SAS System disk.

Tape Contents

The tape that accompanies this document contains SAS/GIS Census Tract Maps for Release 8.2 of the SAS System under CMS. The installation tape is a non-labeled tape and the files are in IBM's `TAPE DUMP` format. Except for the installation aids, the files are blocked `VB` and must be read using `BLOCKTAP`, which is included with the installation aids. The tape's external label indicates the density of your tape.

Note: This document refers to the installation tape, rather than tapes. It is one logical tape, although it can extend over two or more physical cartridges. If you received more than one physical tape, the external labels indicate the number of cartridges. For example, the labels for a two-cartridge tape read *1 of 2* and *2 of 2*.

You can list the products that are on the installation tape and their space requirements by loading the installation aids from the tape, and then with the tape still attached as virtual device 181, issuing:

```
GETGIS TAPEMAP
```

Detailed instructions for doing this are included later in this document. If you do not have the correct tape, contact the SAS Distribution Center for a replacement by

sending email to `distrib@vm.sas.com` or by calling (919) 677-8000, extension 7850, Monday through Friday between 9 a.m. and 8 p.m., Eastern Time.

The installation tape is a collection of CMS files that are logically divided into the following sections:

❑ Installation Aids

This section contains files used to install the SAS/GIS Census Tract Maps, including the `GETGIS EXEC`. These files are loaded to disk using the `CMS TAPE LOAD` command. The `GETGIS EXEC` is the *driver* for the rest of the installation process. This is what you run to load the rest of the SAS System from tape to disk.

❑ TGRMAPS

This section contains all the maps by state.

Contacting SAS

If you have difficulty with this document or any of the procedures described in it, contact the SAS Technical Support Division at (919) 677-8008, between 9 a.m. and 8 p.m., Eastern Time.

Part I, Overview and Checklist of the SAS/GIS Census Tract Maps Installation Steps for CMS

The following is an overview of the steps you follow to install SAS/GIS Census Tract Maps for the CMS environment.

Read the installation instructions before attempting to install SAS/GIS Census Tract Maps.

- ❑ Obtain write access to a small amount of empty disk space to hold the installation aids.
- ❑ Obtain read/write access to the Release 8.2 SAS System disk.
- ❑ Mount the installation tape and have the tape drive attached to the userid.
- ❑ Load the installation aids (including the `GETGIS EXEC`) from tape to disk using `TAPE LOAD`.
- ❑ Determine the disk space requirements. This can be done by running `GETGIS EXEC`.
- ❑ Obtain the required resources to hold SAS/GIS Census Tract Maps, including empty disk space to which you have write access and access to where Release 8.2 of the SAS System was originally installed.

- ❑ Install the SAS/GIS Census Tract Maps from tape to disk by running `GETGIS EXEC`. `GETGIS EXEC` prompts you for all necessary information, then loads the maps from the tape to disk.

Part II, Pre-Installation Considerations

Before you begin installing SAS/GIS Census Tract Maps, perform the following tasks:

- ❑ Obtain write access to a small amount of empty disk space (600 4K blocks should do). The disk space can be either a minidisk or CMS Shared File System directory. This will be used to contain the installation aids or you may use the same disk that was used for the SAS System installation aids.

The installation aids should go on their own disk, rather than putting them on the same disk as the rest of the SAS System or other software products. The installation aids are specific to the tape they are on, so the installation aids from one tape (or set of tapes) should not be used to install another tape (set).

- ❑ Have the installation tape mounted on a tape drive and have the drive attached to your virtual machine at address 181. The tape is non-labeled and the density is indicated on the external physical label. Mount the tape in read-only mode.
- ❑ Load the installation aids (including the `GETGIS EXEC`) from tape to disk. Ensure that you are at the beginning of the tape by issuing:

```
TAPE REW
```

Next, load the files from tape to disk by issuing the following command:

```
TAPE LOAD * * filemode
```

where *filemode* indicates the filemode of the minidisk or Shared File System directory that will contain the installation aids. Do not use 'A' for this filemode.

When this step is completed, `GETGIS EXEC` and all other files from the installation aids section have been loaded to disk. Leave the tape mounted so that you can continue with the next step.

- ❑ Determine the disk space requirements. You can have the disk space requirements calculated for you by running `GETGIS EXEC`. With the tape still mounted and attached as virtual device 181, issue:

```
GETGIS
```

Answer the selection prompts and exit (F3) from the screen showing disk requirements. This screen also prompts for filemodes, but as long as you exit from this screen, none of the pieces of SAS/GIS Census Tract Maps get loaded off the installation tape. A soft copy of the disk space requirements based on your selections can then be found in the file `GETGIS AUDIT` on the installation aids disk. The numbers given are the recommended number of 4K blocks,

including a minimal amount of padding. See the online help (F1) on the disk requirements screen if you want more details.

Your answers to the prompts will be saved and can be used as defaults in a subsequent invocation of `GETGIS` (for example, when you are ready to do the actual loading of SAS/GIS Census Tract Maps from tape to disk). Doing this makes running `GETGIS` a two-pass process. On the first pass, you select what you want to install and you are given summarized disk space requirements. Then, you set up your disk space. On the second pass, you can have your previous selections used as default. Then, you specify where you want things to go and SAS/GIS Census Tract Maps get loaded from tape to disk.

Note: For the most current notes on using `GETGIS`, see the online help associated with each screen (including general information available from the help on the very first screen).

Other methods can be used to determine disk space requirements. First, decide what sections and products you will unload from the installation tape. Then, look up the corresponding sizes in the screen output that is displayed when you issue:

```
GETGIS TAPEMAP
```

with the tape still mounted and attached. You can then manually add up each of the sizes to find out the total disk space requirements. Be sure to include some padding if you are manually calculating disk requirements. The numbers in the *System Requirements* document no longer include any padding. The recommended minimum padding is 150 4K blocks.

Notes: When you install SAS/GIS Census Tract Maps for Release 8.2 of the SAS System under CMS, you can install everything from your installation tape or you can select only certain states off the tape.

Obtain the required disk space. You need to have write access to this disk space and the disk space should be empty. The disk space that you set up should be used exclusively for SAS/GIS Census Tract Maps, rather than installing SAS/GIS Census Tract Maps to a location containing other software products.

Throughout this document, references to CMS minidisk can be replaced by the accessed CMS Shared File System directory.

Part III, Loading SAS/GIS Census Tract Maps from Tape to Disk Using the Installation Aids

Running GETGIS EXEC

Use `GETGIS EXEC` to install SAS/GIS Census Tract Maps. `GETGIS` is a full-screen application that prompts you for necessary information, such as what you want to install and where you want it installed. Based on your answers, the appropriate files are then loaded from tape to disk.

Before running `GETGIS EXEC`, perform the tasks listed in Part II, "Pre-Installation Considerations" starting on page 10. For example, you need to load the installation aids (including `GETGIS EXEC`) from tape to disk and obtain write access to obtain sufficient disk space to hold the SAS System.

You must also have the installation tape mounted on a tape drive and have the tape drive attached to your virtual machine at address 181. The installation tape is non-labeled and the density is indicated on the external physical label. Mount the tape in read-only mode.

To perform the install, run `GETGIS` from a CMS ready prompt by issuing:

```
GETGIS
```

Answer the prompts and let `GETGIS` do the rest.

`GETGIS EXEC` includes the following features.

- Automatic disk space requirements generation and checking
- Ability to page backward through the screens (and change previous answers)
- Screens requiring filemodes allow you to go into CMS subset mode
- Expanded online help - select F1 from any screen to receive help
- An audit trail is kept that records your answers to the prompts and information about your run-time environment
- Answers from a previous invocation can be automatically used as defaults

As part of `GETGIS` processing, some files are created with the naming convention `TGRMAPS KEEP_ME`. As the filetype implies, **do not** erase this file or move it to a different disk. If you do, subsequent installs (for example, for reruns, maintenance, or add-ons) will fail.

For the most current notes on using `GETGIS`, see the online help associated with each screen (including general information available from the help on the very first screen).

Note: GETGIS EXEC turns on BLOCKTAP and NUCXLOADS it. When GETGIS terminates, it turns off BLOCKTAP and NUCXDROPS it.

The GETGIS User Interface

GETGIS operates in full-screen mode. When you issue the GETGIS command, the following message appears:

NOTE: Positioning tape and loading tape map.

Next, the following screen appears.

```

CMS SAS Release 8.2 TS M0                                     1
Thank you for licensing the SAS GIS product for the CMS system.
Copyright(c) 1999 by SAS Institute Inc., Cary, NC USA.
See HELP for an overview of the installation procedure.
Press ENTER to continue
Enter=Continue F1=Help F3=Quit F12=Backtrack

```

Press Enter to continue.

GETGIS then prompts you for information including which states you want to install. Because of the number of states to choose, the default is set to install some. On the State display screens, select F6 to display a description of each product.

GETGIS then shows you the disk space requirements based on your selections and prompts you for the disks to which you want SAS/GIS Census Tract Maps installed. Disk space checking is performed and you will receive messages if the available disk space is not sufficient for what you have selected. Selecting F2 from the Disk Requirements and Filemode Selection screen puts you in CMS subset mode. Type RETURN and press Enter to resume GETGIS.

Any time you run GETGIS, your answers to the prompts are kept even if you exit one of the screens (F3). The next time you run GETGIS, you can choose to have these answers used as the defaults.

The following screen asks if you want to see the name of all files being loaded. This is the last screen GETGIS displays. The files are loaded to disk only if you continue on from this screen:

```

CMS SAS Release 8.2 TS M0                                     10

As files are loaded from the tape, the filename, filetype,
and destination filemode can be displayed.

Do you want fileids displayed as files are loaded (Y|N)? Y

This is the last prompt before files are loaded from the tape.
When you press Enter, loading will begin.

Enter=Continue F1=Help F3=Quit F12=Backtrack

```

After you have responded to all prompts, GETGIS loads or bypasses the appropriate files from the tape to the specified disks. This process takes several minutes.

If you directed GETGIS to display the file information as the files are loaded from tape, information similar to the following is displayed:

```

NOTE: Verifying tape...

NOTE: Positioning tape to first file...
Loading ...
C27001 GISMAPS C1
C27003 GISMAPS C1
C27005 GISMAPS C1
C27007 GISMAPS C1
C27009 GISMAPS C1
C27011 GISMAPS C1
C27013 GISMAPS C1
C27015 GISMAPS C1
C27017 GISMAPS C1
C27019 GISMAPS C1
C27021 GISMAPS C1
C27023 GISMAPS C1
.
.

End-of-file or end-of-tape
GETGIS: Loading from tape is complete. The tape is being
rewound.

GETGIS: installation completed successfully.

Ready;

```

When all requested sections and products are loaded, the tape is rewound (but not detached or dismounted).

Installing from Multiple Tapes

If your installation tape extends over two or more physical tape cartridges, the files selected may not exist on the current tape. When this happens, GETGIS prompts you to mount the next tape with the following sequence of messages:

- End-of-file or end-of-tape
- NOTE: The current tape has been completed and is being rewound.
- GETGIS will now put you in CMS Subset mode so that you can mount the next tape.
- The tape that needs to be mounted is externally labeled nnnnnn.
- When the tape is ready, enter the RETURN command to resume GETGIS.
- Waiting for tape rewind to complete before entering CMS subset...
- Tape is rewound.
- CMS subset

The installation tape needs to be attached as virtual address 181, so you may need to drop the previous tape by issuing:

```
DETACH 181
```

Once the next installation tape has been mounted and attached and you have issued the RETURN command from CMS subset, GETGIS continues the loading process. Information similar to the following is displayed if it was requested:

- NOTE: Verifying tape...
- NOTE: Positioning tape to first file...
- Loading...
- C30001 GISMAPS E1

Full-Screen Key Definitions

One or more of the following key definitions may be active and appear on a screen:

- Enter=Continue

Press ENTER to continue to the next screen. GETGIS validates the data you have entered. If there is invalid data, it will not continue to the next screen.

If the screen you are using has fewer lines than GETGIS is attempting to display, Enter will scroll down to show the rest of the information. Every attempt has been made to avoid this situation.

- F1=Help

Press F1 or F13 for screen-sensitive online help. This includes general help on the initial screen.

- F2=CMS Subset

On the disk requirements and filemode selection screen, press F2 or F14 to go into CMS subset mode. Use the RETURN command to return to GETGIS.

- ❑ F3=Exit

Press F3 or F15 to terminate GETGIS.

- ❑ F6=Describe

On screens that list SAS products, press F6 or F18 to see the product descriptions.

- ❑ F7=Scroll up

Within Help screens, press F7 or F19 to scroll up.

- ❑ F8=Scroll down

Within Help screens, press F8 or F20 to scroll down.

- ❑ F12=Backtrack

Press F12 or F24 to return to the previous screen. Within a help screen, this returns you to the screen from which you issued help.

Rerunning GETGIS

After running GETGIS to load files from tape to disk, you may sometimes need to rerun GETGIS. A typical reason for doing this includes wanting another state that you did not previously select. In these situations, simply rerun GETGIS and select the state(s) that you want to load to disk.

Any time you rerun GETGIS, you need to have write access to all disks that contain any part of your SAS/GIS Census Tract Maps installation.

GETGIS keeps track of what has been previously installed and sets selection defaults to N for those selections/products.

As part of GETGIS processing, a file is created with the naming convention TGRMAPS KEEP_ME. As the filetype implies, this file should not be erased. If you do, subsequent installs (for reruns, maintenance, or add-ons for example) will fail.

Every time you run GETGIS, a file named GETGIS AUDIT is written that contains an audit trail of the execution. If you want to keep the version of this file from a previous execution, you must rename it or make a copy of it before rerunning GETGIS.

Disk Space Implications

If you select anything that was not previously installed, use the information from the disk space requirements screen to help you determine how much disk space you need.

If you want to completely re-install everything you previously installed, erase all files that `GETGIS` loaded to disk before you rerun `GETGIS`. In this case, be sure to erase the `TGRMAPS KEEP_ME` file, but do **not** erase the installation aids.

If you want to reinstall a subset of what you previously installed, you still need to ensure that you have enough disk space to reload the files. When `TAPE LOAD` overwrites an existing file, it temporarily needs space for both the new and the old copies.

There are two ways you can ensure that you have enough disk space:

- ❑ Erase any files that you plan to reload that are bigger than what is currently available on the disk, and/or
- ❑ Get enough additional disk space to hold an extra copy of the largest file that you plan to reload.

Post-Install

In the `SASV8SYS CONFIG` file there is an entry for `GISMAPS`. If reference to this library will be made within the SAS System, remove the `*` (comment).

Return Codes from GETGIS

Specific descriptive messages are issued to the console for all non-zero return codes.

0	Executed successfully
24	Invalid option specified with <code>GETSAS</code> . <i>or</i> <code>GETSAS</code> was issued from within <code>CMS SUBSET</code> . <i>or</i> Unrecognized tape type <code>XXXX</code> . <i>or</i> Invalid <code>GETSAS TAPEMAP</code> file (possible invalid SAS distribution tape).
28	Required installation aids file not found or not unique. <i>or</i> Tape not attached as 181 or wrong tape. <i>or</i> Could not turn on blocktap.
36	Write access to the SAS installation tools disk is required but not available.
100	<code>TAPE LOAD</code> failed or other <code>TAPE</code> command error
101	Abend

Appendix G, Post-Installation Setup for SAS/GRAPH® Software

If you install SAS/GRAPH software from your tape, you get the SAS/GRAPH procedures and device drivers. This appendix explains how to interface SAS/GRAPH with various graphics devices. Please read the sections that apply to the graphics devices that you will be using with SAS/GRAPH. Save these instructions for later reference if you obtain additional devices. The following sections are included in this appendix:

- ❑ Client-Side Components
- ❑ Managing Device Catalogs
- ❑ Using SAS/GRAPH with ASCII Devices
- ❑ Installing the Linkable Driver
- ❑ Using SAS/GRAPH with IBM 3270-Type Terminals and 3270 Emulators
- ❑ Using SAS/GRAPH with IBM 3287, 3268, and 4224 Printers
- ❑ Using SAS/GRAPH with GDDM

Client-Side Components

Using SAS/GRAPH Software, Release 8.2 you can produce interactive charts and plots for Web publishing. The GCHART, GCONTOUR, GMAP, GPLOT, and G3D procedures can produce scripted ActiveX Controls or Java Applets in HTML pages using the SAS/GRAPH Java or ActiveX drivers and the Output Delivery System (ODS). The DS2GRAF, DS2CSF, and META2HTM macros can also be used to generate HTML output with embedded ActiveX Controls or Java Applets.

The following controls and applets are available:

ContourApplet (ctrapp.jar)

A scriptable Java applet for visualization of Contour and Surface plots in a Web browser. The applet supports outline and filled modes and interactive exploration of the data. ContourApplet is supported by the SAS/GRAPH Java driver with ODS.

GraphApplet (graphapp.jar)

A scriptable Java applet for visualization of 2D and 3D charts in a Web browser. The applet supports Bar charts, Pie charts and Scatter Plots, and interactive exploration of the data. GraphApplet is supported by the SAS/GRAPH Java driver with ODS, and also by the DS2GRAF macro.

MapApplet (mapapp.jar and related map data jar files)

The Java Map Applet is a scriptable Java graphics control that allows the user to embed interactive spatial data in a Java 1.1 compliant Web page. The Web page is created with PROC GMAP and uses the map jar files that correspond to the SAS Map Data Sets. MapApplet is supported by the SAS/GRAPH Java driver with ODS.

MetaViewApplet (metafile.zip)

A Java applet for displaying SAS/GRAPH metagraphics data. MetaViewApplet is supported by the SAS/GRAPH metagraphics driver and the META2HTM macro.

RangeViewApplet (rvapplet.jar)

A Java applet for displaying a Critical Success Factor (CSF). A CSF is a graphic that visually represents the position of some value in a range of data. RangeViewApplet is supported by the DS2CSF macro.

SAS/GRAPH Control for ActiveX

This ActiveX control enables you to embed interactive graphs in Web pages and OLE documents (in Microsoft Office products), as well as in applications written in Visual Basic, C++, HTML, and JavaScript. When the graph is displayed, you can point-and-click to rotate, change, or further investigate the graph.

In Release 8.2 of SAS software, the SAS/GRAPH Control for ActiveX supports the following graph types:

- Area plots
- Bubble plots
- Line plots
- Regression plots
- Surface plots
- Bar charts
- Contour plots
- Maps
- Scatter plots
- Box-and-Whisker plots
- High-Low plots
- Pie charts
- Standard Deviation plots

SAS/GRAPH Control for ActiveX is supported by the SAS/GRAPH ActiveX driver with ODS, and also by the DS2GRAF macro.

Installing the Client-Side Components

Documentation for the SAS/GRAPH Client Components can be found on the SAS Web site:

<http://www.sas.com/rnd/webgraphs>.

Documentation for the HTML Formatting Tools (including the DS2GRAF, DS2CSF, and META2HTM macros) can be found at

<http://www.sas.com/rnd/web/format>.

If you wish to publish SAS/GRAPH output on a Web server or create SAS/IntrNet applications using SAS/GRAPH, you may need to install these SAS/GRAPH clients

on your Web server. The clients can be installed from *the SAS Client-Side Components* CD included with your SAS Software distribution. Please see the installation instructions on the *SAS Client-Side Components* CD for more information on installing SAS/GRAPH clients on a Web server.

Managing Device Catalogs

Setting Up and Modifying Device Catalogs

After installing SAS/GRAPH software, you may need to create or modify device catalog entries in order to customize device driver output to your site's needs. This section gives a brief explanation of device catalogs, and how to handle situations where catalog entries may need to be modified. For complete details on managing device catalogs, refer to "The GDEVICE Procedure," in *SAS/GRAPH Software: Reference*.

How Device Catalogs are Used

In Release 8.2 of SAS/GRAPH software, when you specify the name of a graphics device driver, the name you specify corresponds to an entry in a device catalog. Device catalog entries contain default characteristics (such as graph size, picture orientation, and default colors) used by the driver. You can change the characteristics used by a driver either by modifying its entry in the device catalog or by specifying `GOPTIONS` that override settings in the catalog entry.

For example, if you specify `DEVICE=HP7550`, the SAS/GRAPH procedure attempts to find an entry named `HP7550` in available device catalogs. The parameters found in the entry (such as the default graph size, graph orientation, or output destination) are used in generating the graph. If you want to change the way the driver produces output, you can use the `GDEVICE` procedure to modify parameters in the catalog entry, or you can override them with a `GOPTIONS` statement.

In general, if you want to change defaults for a single session or job, you can use a `GOPTIONS` statement. If you want to permanently change the default parameters used by a driver, you can create a new device entry or modify an existing one.

A supplied device catalog, `SASHELP.DEVICES`, is installed and made available as part of the overall SAS/GRAPH installation process. This catalog contains over 300 entries, covering each graphics device and model that SAS/GRAPH software supports. Individual users or groups can also create their own device catalogs. These are given the names `GDEVICE0.DEVICES`, `GDEVICE1.DEVICES` and so on, through `GDEVICE9.DEVICES`.

When a driver name is specified in a SAS program, the SAS System looks for the corresponding entry in `GDEVICE0.DEVICES`, `GDEVICE1.DEVICES`, etc. If the entry is not found in any of the user catalogs (or the catalogs do not exist), the supplied catalog, `SASHELP.DEVICES`, is searched.

How and When do You Modify Catalog Entries

How you change a device entry depends on the number of users affected by the change. If the change affects only one user, that user should create a device catalog (`GDEVICE0.DEVICES`), copy the device entry from `SASHELP.DEVICES`, and make the changes to the entry in `GDEVICE0.DEVICES`. Note that `GDEVICE0.DEVICES`, being a *personal* catalog, is usually a different catalog for each user. If the modification affects a large number of users, the SAS Installation Representative or SAS Software Representative can make modifications to an entry in `SASHELP.DEVICES`.

Use the following guidelines when creating or modifying device entries:

- ❑ Only the SAS Installation Representative or SAS Software Representative should add or modify entries in `SASHELP.DEVICES`. End users should not have update access to `SASHELP.DEVICES`. If individual users need to make modifications, they should create their own `GDEVICE0.DEVICES` catalog.
- ❑ If a catalog entry needs to be modified, create a new entry (with a different name) and modify the new entry. By renaming modified entries, users are ensured that the original entries supplied in the `SASHELP.DEVICES` catalog use default settings.
- ❑ Any options specified in a `GOPTIONS` statement override equivalent parameters in device catalogs. If a parameter needs to be changed only for a single session, it is probably easier to use a `GOPTIONS` statement than to create a new device entry.

Example

The following example illustrates how to use device catalog parameters to spool output directly to a hardcopy device. The example first illustrates `GOPTIONS` and `FILENAME` or host statements that can be used to produce output, and then shows how equivalent parameters can be specified in a device entry, eliminating the need for the statements in the end-user's programs.

Spooling Directly to a Graphics Device

Suppose you want to use the PS300 driver and send the output directly to a PostScript printer attached to an AGILE 6287 protocol converter. The destination name for the printer is `PSPRINT`, and you want to send output directly to the printer through RSCS by using `CP SPOOL` and `CP TAG` commands.

The following `GOPTIONS` and `FILEDEF` statements are used to send graphics output to the PostScript printer.

```
/* define fileref and file attributes for graphics stream
file */
CMS CP SPOOL PRINT TO RSCS;
CMS CP TAG DEV PRINT PSPRINT;
CMS FILEDEF GSASFILE PRINT;
```

```

/* specify device driver, fileref for GSF, */
/* protocol converter, and record length */
goptions dev=ps300 gaccess=gsasfile gprotocol=sasgpagl
gsflen=64;

```

You can achieve the same results by creating your own driver with the `GDEVICE` procedure and specifying host file options. The following displays show the Host File Options window and the Host Commands for the modified device entry `MYPS300`. You can enter these values using `GDEVICE` windows or with line-mode `GDEVICE` statements.

```

GDEVICE: Host File Options
-----
Command === >
          Catalog: GDEVICE0.DEVICES      Entry: MYPS

Gaccess:  GSASFILE

Gsfname:  _____      Gsfmode: REPLACE      GSflen:      64
Trantab:  _____      Devmap:  _____      Devtype:    PRINTER
Gprotocol: SASGPAGL

Host file options:
-----
-----
* Close file at end of driver or procedure termination
° Close file at end of each graph

                                           ZOOM  _R

```

```

GDEVICE: Host Commands
Command === >
          Catalog:  GDEVICE0.DEVICES      Entry:  MYPS

Driver Initialization:
  1  CP SPOOL PRINT TO RSCS
  2  CP SPOOL TAG DEV PRINT PSPRINT

Pre-Graph commands:
  1  _____
  2  _____

Post-Graph commands:
  1  _____
  2  _____

Driver Termination:
  1  _____
  2  _____

```

ZOOM — R

The CP SPOOL and CP TAG commands are issued at driver initialization and direct the fileref GSASFILE to RSCS PRINT. The value of GSASFILE for the GACCESS parameter causes driver output to be sent to the fileref of GSASFILE.

Using SAS/GRAPH with ASCII Devices

This section describes the steps necessary to initialize the SAS/GRAPH System if you will be using supplied drivers for ASCII terminals, plotters, and printers. If you do not use ASCII devices, skip this section and go on to the sections that are relevant to your device(s).

The first section on setting up a graphics translate table should be followed if you will be using asynchronous terminals and/or attached plotters (in “eavesdrop” mode) on ASCII or TTY lines. Graphics commands must be translated from ASCII to EBCDIC, and back to ASCII again. This requires both system and SAS/GRAPH translation tables.

Note: You do not have to complete this step if all of your asynchronous devices are used with supported 3270- or 3287-type protocol converters.

The second section describes how to use ASCII terminals and ASCII terminal emulators with SAS/GRAPH. Also included is a list of 3270-style protocol converters supported by SAS/GRAPH.

The third section describes how to use ASCII printers, plotters, and cameras in "spooled" mode, attached to a 3287-style protocol converter supported by SAS/GRAPH.

Setting up a SAS/GRAPH Translate Table

A translate table is only needed if you are using ASCII terminals or attached plotters interactively on ASCII lines. You do not have to worry about translate tables if all of your asynchronous devices are used with 3270-type or 3287-type protocol converters.

If you encounter problems using SAS/GRAPH software with ASCII terminals or attached plotters interactively on ASCII lines, contact SAS Technical Support.

Using SAS/GRAPH with ASCII Terminals and ASCII Terminal Emulation Software

If you encounter problems using SAS/GRAPH software with ASCII terminals, and PCs running ASCII terminal emulation software, contact SAS Technical Support.

Using SAS/GRAPH with ASCII Printers

When using SAS/GRAPH software with ASCII printers, the typical GOPTIONS statement that you need to specify in your SAS/GRAPH program is as follows:

```
GOPTIONS DEVICE = driver-name
          GPROTOCOL = protocol-module
          GSFNAME = GSASFILE
          GSFMODE = REPLACE
          GSFLEN = 80;
```

These GOPTIONS tell the device driver to direct the graphics output to the FILENAME of GSASFILE. To use a FILENAME statement to assign the fileref of GSASFILE to a permanent data set, specify the following:

```
FILENAME GSASFILE 'filename filetype filemode';
```

To direct the output to RSCS PRINT, for example, specify the filename statement as follows:

```
FILENAME GSASFILE PRINTER;
```

The following are some valid values for the `protocol-module` value shown above:

SASGPASC	Formats the graphics data stream as straight ASCII.
SASGPSTD	Formats the graphics data stream as straight EBCDIC.
SASGPAGL	Use with an AGILE protocol converter.
SASGPISI	Use with an ISI 87 converter (as well as some AGILE converters).
SASGPLCL	Use with converters from Andrew Corporation (as well as older converters from KMW and Local Data).

SASGPAXI	Use with converters from AXIS Corporation.
SASGPVAT	Use with converters from Avatar.
SASGPIDX	Use with converters from IDEX Corporation.
SASGPNET	Use with converters from NetCommander, I-Data, and several others.
SASGPDCA	Use with IRMAprint, IRMAprint2, TEK 4512, and QMS AFPLink converters.
SASGPCAB	Use with converters from Cablenet.
SASGPCHK	Use with a Cherokee protocol converter.
SASGPIOC	Use with converters from I/O Corporation.

For more information about using SAS/GRAPH with ASCII printers, consult the on-line help for SAS/GRAPH device drivers. If you encounter problems using SAS/GRAPH software with ASCII printers, contact SAS Technical Support.

Installing the Linkable Driver

The Linkable device driver is a special SAS/GRAPH device driver that makes calls to vendor-supplied 'CalComp compatible' plotting routines. Before you can use the Linkable driver, parts of it must be compiled and link-edited with your plotting routines.

Devices that typically require the use of the Linkable driver include Xerox 9700, 9790, and 8700 printers. If your site does not have a device that uses vendor-supplied subroutines, you do not have to worry about building a Linkable driver. If you do have devices that use vendor-supplied subroutines, contact SAS Technical Support for more information.

Using SAS/GRAPH with IBM 3270-Type Terminals and 3270 Emulators

If you are using a display device that supports mainframes and you are running the SAS Display Manager system, output is automatically displayed on your screen when you run a SAS/GRAPH procedure. Therefore it is not necessary to specify a SAS/GRAPH device driver.

If you encounter problems displaying SAS/GRAPH output on the screen of your 3270 display device, the problem is usually because your device is not properly configured to support mainframe graphics. To determine the graphics capability of your 3270 display device, invoke the SAS Display Manager and issue the `TERMSTAT` command from any command line. The `TERMSTAT` command writes device configuration information to the log window. In the SAS log, check that the following two lines appear under the `DEVICE FEATURES` section:

```
Extended Data Stream
Vector Graphics
```

If these two lines do not appear under the `DEVICE FEATURES` section of the `TERMSTAT` information, check the following:

- ❑ If you have a real IBM graphics terminal, check to be sure that it supports mainframe graphics.
- ❑ If you have a PC running 3270 emulation software, make sure that your 3270 emulation software supports host graphics and that your 3270 host session has been configured to support host graphics.
- ❑ Make sure that your 3270 display device is defined to VM and/or VTAM as a device that supports extended data streams.
- ❑ Make sure that the control unit that your 3270 display device is attached to supports (or is configured to support) 3270 graphic data streams.

For more information, consult the on-line help for SAS/GRAPH device drivers. If you encounter problems using SAS/GRAPH software with 3270 display devices, contact SAS Technical Support.

Using SAS/GRAPH with IBM 3287, 3268, and 4224 Printers

SAS/GRAPH software supports IBM graphics printers using either native (non-GDDM) device drivers or GDDM device drivers. If you encounter problems using SAS/GRAPH with IBM graphics printers, contact SAS Technical Support.

Using SAS/GRAPH with GDDM

SAS Institute provides a set of drivers that interface with IBM's GDDM base product and can be used to direct output to any device supported by GDDM. The set of GDDM drivers is included with SAS/GRAPH software; special action is required to install the GDDM device drivers. Some of the GDDM device drivers are provided as an alternative to the SAS/GRAPH device drivers. For example, both the GDDMPCG and IBM3179 drivers produce graphics on an IBM 3179 Model G graphics terminal. The advantages of one over the other vary and will depend on a particular site's requirements. On the other hand, some graphics devices require the use of a GDDM driver. These devices include IEEE-attached plotters (IBM 7372, IBM 6180, etc.) and IBM 3800-type laser printers (IBM 3800, 3812, 3820, etc.).

To use any of the GDDM device drivers, the GDDM base product must be installed on your system.

Note: The Presentation Graphics Feature (PGF) and the Interactive Chart Utility (ICU) are not required to use any of the GDDM interface drivers.

Before using any of the drivers, make sure that the required GDDM TXTLIB files are made available. To do this, issue the following CMS command **before** invoking the SAS System:

```
GLOBAL TXTLIB ADMRLIB ADMGLIB
```

For more information about using SAS/GRAPH software with GDDM, consult the on-line help for SAS/GRAPH device drivers. If you encounter problems using SAS/GRAPH software with GDDM, contact SAS Technical Support.

Appendix H, Post-Installation Setup for SAS/MDDDB Server[®] Software

SAS/MDDDB Server Software includes an OLE DB provider, Open OLAP Server Software. The Open OLAP Server allows you to access, update, and manipulate MDDDB data on your SAS System from OLE DB and ADO compliant applications on Windows platforms.

The Open OLAP Server is packaged as a self-installing program for Windows platforms and is available on the *SAS Client-Side Components* CD included with your SAS Software distribution. Please refer to the *SAS Client-Side Components* CD for installation instructions.

Appendix I, Post-Installation Setup for the Metabase Facility

Starting with Version 7 of the SAS System, the SAS/EIS Metabase facility was converted to the new Common Metadata Repository. The Common Metadata Repository is a general-purpose metadata management facility that provides common metadata services to various metadata-driven applications. The Common Metadata Repository enables applications to share metadata between SAS System products.

Using the Common Metadata Repository requires a one-time setup. If the repository manager was set up in a previous release, it may not need to be set up again. The steps in the following sections should be completed before you attempt to use the Metabase Facility. For Metabase Facility users who were using a release prior to Version 7, using the Common Metadata Repository requires a conversion.

Setting Up the System Repository Manager Files

Complete the following steps to set up the necessary system repository manager files. You must have write access to `SASHELP` in order to specify the system repository manager.

1. Create a directory that will be dedicated exclusively to the storage of repository manager files. This directory should not be used to store other SAS files.
2. At a SAS command line, type `REPOSMGR` and then select `Setup Repository Manager`.
3. In the `Repository Manager Setup` window, `Library` will default to `RPOSMGR`. For `Path`, specify the path from Step 1, above, and then select the `Write values to system registry` check box. Then select `OK`.
4. In the resulting dialog window, select `Yes` to generate the necessary repository manager files.

This completes the setup for the System Repository Manager. You can create additional repository managers (a user repository manager, for example) by repeating the steps above and then using a different path.

Note: This step sets the default location for the repository manager for your site. Individual users may specify their own repository manager location by following the steps above and not selecting the `Write values to system registry` check box.

Registering the SASHELP Repository in the Repository Manager

The SASHELP repository is used in various samples. Before beginning the steps below, a repository manager must be created (see previous section). Complete the following steps to register the SASHELP repository in the Repository Manager:

1. At a SAS command line, type REPOSMGR and then select Repository Registration.
2. In the Repository Registration window, select New.
3. In the Register Repository (New) window, type SASHELP (in uppercase) in the Repository field, and then type the full directory path where the CORE catalog is located in the Path field.
4. In the Description field, you can type any character string (for example, SASHELP Repository). Select OK to close the Register Repository (New) window. Select Close to exit the Repository Registration window.

Note: Repositories cannot span multiple directories because the path cannot contain concatenated directories. If you have existing metabases in concatenated directories, you should copy the metabases to a single path that will be referenced as a repository.

Appendix J, Post-Installation Setup for SAS[®] OLAP Server Software

Setting up Access Control without SAS/EIS Software on Your Server

Please keep in mind that Access Control Setup consists of three steps:

1. Set your Access Control Key - modifies `SASHELP.MB`
2. Set your Access Control environment (`aclroot`, `ac_active` flag, etc.) - modifies `SASHELP.AC`
3. Create your Access Control definitions (users, groups, the actual ACL) – data sets `PASSWD`, `GROUPS`, and `ACL` in `aclroot`

Each of these steps can be performed interactively in a set of windows (where available), or programmatically.

Starting the Access Control Setup Dialog Window

Use the following command:

```
AF C=SASHELP.EISSRV.STARTAC.SCL <USER=uid PASSWD=password>
```

Follow the instructions in Help or the description in *SAS/EIS Software: Administrator's Guide - Access Control Tasks* to assist you through the setup process. On 3270 platforms, where the Access Control Setup GUI is not very comfortable to use, you may prefer to do your AC setup programmatically. Please see the following section on more information on that.

Setting Your Access Control Key and Environment Programmatically

Setting the Access Control Key

The Access Control Key is stored in the entry `SASHELP.MB.ACLAPWM.SCL`. You need write access to this entry in order to change the Access Control Key. Please refer to *How to set up write access to SASHELP.AC and SASHELP.MB* for more information on that.

Using a command

Use the following command to set the Access Control Key (for setting the Access Control Key to `ADMIN`):

```
AF C=SASHELP.EISSRV.SETAPW.SCL PW=ADMIN
```

- To reset the Access Control Key to its initial status (no key set), pass in an empty string ("").
- Use the special value "0" to use no Access Control Key.
- Use the option ECHO=Y to dump the settings in the log.

Using a statement

Use the following command to submit the command as a SAS statement:

```
DM 'AF C=SASHELP.EISSRV.SETAPW.SCL PW=ADMIN' ;
```

Using SCL

Within SCL code, you can use the following method call to set the Access Control Key:

```
CALL METHOD ('SASHELP.MBEISSRV.APWUTIL','CREAAPWM', flag, pw-value, rc);
```

where

- `flag` is 0 or 1. 0 indicates to not use a control key; 1 indicates to use a control key.
- `pw-value` is the value of the new control key. If flag is 0, this value is ignored.
- `rc` is 0 if the update was successful; 1 if it was not successful.

Setting the Access Control Environment

The Access Control Environment information is stored in the entry `SASHELP.AC.ACLINIT.SCL`. You need write access to this entry in order to change the Access Control Environment settings. Please refer to *How to set up write access to SASHELP.AC and SASHELP.MB* for more information.

Using a command

Use the following command to set the Access Control Environment:

```
AF C=SASHELP.EISSRV.SETAC.SCL
APW=access control key
ACTIVE=Y/N
ACLROOT=access control root path
ACLSERV=server
LOGIN=login application
AUTOUSER=Y/N
LIBSEC=Y/N
PW_ENCRYPT=Y/N
DISP_CLASS=access control start class
QUERY_CLASS=access control query class
SERVER_CLASS=access control server class
ADMIN_CLASS=access control administration class
ECHO=Y
```

Using a statement

submit the previous command using a DM statement, e.g.,

```
DM 'AF C=SASHELP.EISSRV.SETAC.SCL APW=ADMIN ACTIVE=Y
ACLROOT="path" ' ;
```

Using SCL

Within SCL code, you can use the following method call to set the Access Control Environment:

```
CALL METHOD ('SASHELP.EISSRV.ACLUTIL', 'CREAACLI', rc, flag
active, aclroot, aclserv, login_window, autouser_enabled,
libsec, pw_encrypt, disp_class, query_class, server_class,
admin_class);
```

Key	Description
APW	The Access Control Key (required for <code>setac</code>)
ACTIVE	Y/N to switch access control on or off
ACLROOT	The path of a directory that holds the ACL files
ACLSERV	The name of the remote session or share server for <code>ACLROOT</code> . If the session is local, this parameter should be blank.
LOGIN	The four-level name of the AF application or <code>APPLSCR</code> to use as a login dialog. The default is <code>SASHELP.EISSRV.GATE_KPR.FRAME</code> - a dialog with entry for User ID and Password, and OK and Cancel buttons. There is one other login dialog provided with the system, <code>SASHELP.EISSRV.GATE_KP2.FRAME</code> , which has an additional Change Password button.
AUTOUSER	Y/N to indicate whether to allow the use of the <code>USER=</code> and <code>PASSWD=</code> options on the <code>EIS</code> , <code>RUNEIS</code> , and <code>METABASE</code> commands. If these options are given, no login dialog appears (Default=Y)
LIBSEC	Y/N to indicate when the temporary library to access access control files is to be allocated 'Y' (default): the <code>ACLTMP</code> library is allocated before and deallocated after each access to the ACL files. Use this setting to assure that the ACL files don't show up in the SAS Explorer. 'N': the <code>ACLTMP</code> library is allocated once at access control server initialization and deallocated at access control server termination.

PW_ENCRYPT	Y/N indicates whether to encrypt the user password stored in the <code>PASSWD</code> file. (Default=Y).
DISP_CLASS	The class used to start the access control subsystem and optionally display a login dialog. The default is <code>SASHELP.EISSRV.ACLDISP.CLASS</code>
QUERY_CLASS	The class used to satisfy queries on the current access control permissions. The default is <code>SASHELP.MB.ACLMAIN.CLASS</code> .
SERVER_CLASS	The class used for loading and persisting <code>acl</code> information. The default is <code>SASHELP.EISSRV.ACLSERV.CLASS</code> .
ADMIN_CLASS	The class used managing user and group information and for updating the ACL. The default is <code>SASHELP.MB.ACLADMIN.CLASS</code> .
ECHO=Y	Dump the current and updated settings in the LOG.
RC	(<code>creaacli</code> only) a flag that indicates if the update was successful, where '0' indicates that the update was successful and '1' indicates that it was not.

Doing Your Access Control Definitions (Users, Groups, ACL) Programmatically

To do your Access Control definitions programmatically, you need to know some basics about the storage of User and Group information and the actual Access Control List.

The Access Control definitions are stored in three data sets in the Access Control Root Path. The data sets are password-protected and encrypted using the Access Control Key.

User definitions are stored in the `PASSWD` data set. Group definitions are stored in the `GROUPS` data set. The Access Control List is stored in the `ACL` data set.

To do your definitions, proceed using the following steps:

1. Set up the Access Control Key and Environment page 97
2. Set a libname ACL on your `aclroot` path page 97
3. Define the groups page 97
4. Define the users page 98
5. Create your metabase registrations page 100
6. Create your ACL page 100
 - A. Initialize partial ACL data sets page 100
 - B. Edit the partial ACL data sets page 101
 - C. Merge the partial ACL data sets page 104

1. Set up the Access Control Environment

Here is a simple example of how to Set up the Access Control Environment. Choose an Access Control Key, and create a location where you want to store your AC definitions. Then submit:

```
DM 'AF C=SASHELP.EISSRV.SETAPW.SCL PW=access control key';
DM 'AF C=SASHELP.EISSRV.SETAC.SCL
  APW=access control key
  ACLROOT="access control root path"
  PW_ENCRYPT=N';
```

Note: By default, user passwords stored in the `PASSWD` data set are encrypted using the `_encryptPassword` method of the `ACLSERV` class. This adds an additional layer of protection to the information stored in the `PASSWD` data set. To be able to store plain text passwords in the `PASSWD` data set when managing the user setup outside of the Access Control definition dialogs, use `PW_ENCRYPT=N` option when setting up the access control environment.

2. Set a libname ACL on your aclroot path

```
LIBNAME ACL "access control root path";
```

3. Define the groups

The `GROUPS` data set holds the names and descriptions of the access control groups. The data set has one record for each group defined to the system. When the `GROUPS` data set is initially created, two additional records are also added, one for the `SYSTEM` (Administrator) and another for the `USERS` (Users) group.

A Group name can have from three to eight characters. Group names begin with a letter, and are followed by letters, numbers, or underscores. Letters must be in upper case.

The `GROUPS` data set has the following structure:

GROUP	\$8	Group Name (needs to be upper case!)
DESC	\$32	Group Description

You can edit the `acl.groups` data set by using an interactive facility, like `FSEDIT`, or `FSVIEW`, or data management tools like the data step.

Example for using a data step:

```
data work.groups;
  infile datalines;
  length group $8 desc $32;
  input group / desc &;
  datalines;
  SALES
```

```

Sales Staff
MKT
Marketing
MGMT
Management
;
proc sort data=work.groups;
by group;
proc sort data=acl.groups(pw=access control key);
by group);
data acl.groups(pw=access control key);
merge acl.groups(pw=access control key) work.groups;
by group;
run;

```

Group names must be upper case valid SAS names, of three to eight characters length.

Please note that when you first activate Access Control (using either the Access Control Setup window, or the SETAC facility) a GROUPS data set is created in your aclroot path, with the two groups SYSTEM and USERS already defined. That is why the previous data step merges your new definitions with the already existing ones.

4. Define the users

The PASSWD holds the definitions for the access control users. The following information is stored for each user:

User id	A 32-character string that must start with a character, followed by characters, numbers, or underscores. The <code>userid</code> is stored in upper case.
Description	Mixed case, free format descriptive string.
Groups	Names of the groups a user belongs to, in upper case, separated by commas.
Password	A 16-character string that must start with a character, followed by characters, numbers, or underscores. By default, this password is stored encrypted using the <code>_encryptPassword</code> method of the <code>ACLSERV</code> class. Use <code>PW_ENCRYPT=N</code> when setting up the access control environment to use unencrypted passwords. Unencrypted passwords are stored in upper case.
Creation date/time	A SAS datetime value indication the creation time of the user's record.

The data set holds one record for each user of the system. When the `PASSWD` data set is initially created, one record for the `ADMIN` user (password `ADMIN`) is added.

The `PASSWD` data set has the following structure:

USERID	\$32	User ID (upper case)
FULLNAME	\$32	User Description
GROUP	\$198	User Groups
PASSWORD	\$16	User Password
C_DATET	\$8	DateTime

You can edit the `acl.passwd` data set by using an interactive facility, like `FSEDIT`, or `FSVIEW`, or a data management tool like the data step.

Example for using the data step:

```

data work.passwd;
  infile datalines dsd;
  length userid fullname $32 group $198 password $16 c_datet
  8;
  format c_datet datetime16.;
  c_datet=time();
  input userid / fullname & / group / password ;
  datalines;
  MJONES
  Markus Jones
  SALES
  MJONES1
  OFIELDS
  Oscar Fields
  MKT
  OFIELDS1
  ABEAN
  Abraham Bean
  SALES,MKT,MGMT
  ABEAN1
  ;
  proc sort data=work.passwd;
  by userid;
  proc sort data=acl.passwd(pw=admin);
  by userid;
  data acl.passwd(pw=admin);
  merge acl.passwd(pw=admin) work.passwd;
  by userid;
  run;

```

Userids must be upper case valid SAS names, of 3 to 32 characters length. Passwords must be upper case valid SAS names, of 3 to 16 characters length.

Please note that when you first activate Access Control (using either the Access Control Setup window, or the SETAC facility) a PASSWD data set is created in your `aclroot` path, with the user ADMIN (password ADMIN) already defined. That is why the previous data step merges your new definitions with the already existing ones.

5. Create your metabase registrations

If you have not already done so, create your metabase registrations now. An Access Control definition is always linked to an existing metabase registration.

Use the METABASE command to invoke the Metabase GUI.

6. Create your ACL

Now, for each group/metabase registration combination, fill a data set with the structural information from the metabase registration, and, if needed and available, with the class column value combinations. There is a utility, FILLACL, that does that for you. Then edit those partial data sets to set your access control tags. And finally, merge the partial ACL data sets back into ACL.ACL.

A. Initialize partial ACL data sets

Note: The FILLACL utility uses the SAS OLAP Server classes to access the data. If you do not have SAS/EIS software, you might have to run the following utility first, to make sure the correct data model classes are being used:

```
DM 'AF C=SASHELP.EISSRV.SET_OLAP_CLASSES.SCL
  MODMGR=SASHELP.EISSRV.MODMGR.CLASS
  MODMGRE=SASHELP.EISSRV.MODMGRE.CLASS
  EMDDDB_C=SASHELP.EISSRV.EMDDDB_C.CLASS';
```

Use the FILLACL utility to create a data set with the same structure as the ACL data set, and initialize it with information from the registration and the data.

```
DM 'AF C=SASHELP.EISSRV.FILLACL.SCL
  APW=access control key
  OUTDS=partial ACL data set name
  GROUP=groupname
  REP="repository name"
  REG="registration name"
  LEVEL=ALL/DIMSONLY';
```

using a different OUTDS= value each time, and setting the other options accordingly.

FILLACL accepts the following named parameters:

Key	Description
APW	The access control key. This is required.
OUTDS	The data set where the partial ACL file should be written. If the data set exists, it will be overwritten.
GROUP	Name of the user group for initializing the GROUP column (upper case!)
REP	The name of the repository in which the registration is stored. Use quotes if the repository name contains blanks or special characters.
REG	The name of the metabase registration to use. Please note that the typical registration name has the form LIB.MEM, e.g., SASHELP . PRDMDDB. By default, a metabase registration has the name of the SAS file (data set or MDDB) that was registered.
LEVEL	ALL/DIMSONLY. ALL is the default. DIMSONLY only reads out the structural information, no data values.

B. Edit the partial ACL data sets

Edit each data set created by `FILLACL`, using an interactive facility like `FSEDIT` or `FSVIEW`. Usually, you would only edit the `TAG` column. Use `TYPE`, `VALUE`, and `ITEM` to identify the element for which a tag will be set.

For example, to drop the `COUNTRY` variable, find the record with `TYPE=CL`, `VALUE=COUNTRY`, and set a 'D' in the `TAG` column.

Do not set any values for `TAG` in those records that you do not want to restrict. These records will be removed when merging the partial ACL data sets in the next step.

Please refer to *SAS/EIS Software: Administrator's Guide - Using Access Control* for information on how to use Access Control tags.

Each partial ACL has the following structure:

GROUP	\$8	Group Identifier
TARGET	\$17	Target Identifier
TYPE	\$2	Information Type
ITEM	\$42	Information Item
VALUE	\$200	Information Value
TAG	\$1	Access Control Tag

The columns hold the following information:

Column name Description - values

GROUP	Name of the group to which the access control definitions in the current record apply (upper case!)	
TARGET	The ID of the metabase registration to which the access control definitions in the current record apply, or #A, for applications/application databases, or #F, for application functions.	
TYPE	Record type: If TARGET is a metabase registration ID:	
	T	Table
	H	Hierarchy
	HL	Hierarchy Level
	A	Analysis variable (ANALYSIS, COMPUTED)
	S	Statistic
	C	Category variable (CATEGORY)
	CL	Category variable level (data value)
	If target is #A:	
	AP	for application
	AD	for application database
	If target is #F: Always F	

VALUE	Depending on TYPE, value can be:	
	Type	Value
	T	TABLE (dummy value when the whole table is being dropped)
	H	Hierarchy name
	HL	Hierarchy level name
	A	Analysis variable name
	S	Statistic keyword
	C	Category variable name
	CL	Category variable value. Special value #T for <u>_Total_</u>
	AP	Application name (2-level)
	AD	Application Database name (4-level)
	F	Function id.
ITEM	Additional identifier, set to identify what the VALUE refers to for	
	HL	to identify the hierarchy
	CL	to identify the Category variable
	S	to identify the Analysis variable

TAG	Access Tag. This is the only column you would typically edit. Valid tags are as follows:	
	D	Drop
	K	Keep
	I	Initial
	H	Hide
	S	Show
	Valid TAGs by TYPE:	
	T	D
	A	D, K, H
	S	D, K
	H	D, K
	C	D, K
	CL	D, K, I, H, S
	HL	D, K, I
	AP	D, K
	AD	D, K
	F	D, K

C. Merge the partial ACL data sets

The result is a collection of data sets. To merge them and remove the unneeded observations (the ones with TAG= ' '), submit:

```
data acl.acl(pw=access control key encrypt=yes);
set work.one
work.two
.
.
;
if tag = ' ' then delete;
run;
```

Write Access to SASHELP.AC and SASHELP.MB

1. Choose an empty library or path for use as a playpen. Later you can either merge it into your SASHELP library, or concatenate it in front of your SASHELP path.

```
libname playpen 'path';
proc catalog;
  copy in=sashelp.ac out=playpen.ac;
  copy in=sashelp.mb out=playpen.mb;
  select aclapwm.scl;
run;quit;

catname sashelp.ac (playpen.ac);
catname sashelp.mb (playpen.mb sashelp.mb);
```

2. Set up your Access Control Key and Environment. When setting up your SAS application server, or distributing the application to your users, make sure the modified catalogs are concatenated in front of your SASHELP path by modifying the SAS CONFIG file, or the SAS clist accordingly.

Specifying OLAP Classes

The following utility program can be used to override the default OLAP Server classes and specify your custom OLAP Server classes.

```
DM `AF C=SASHELP.EISSRV.SET_OLAP_CLASSES.SCL
MODMGR=
MODMGRE=
EMDDB_C=
DP=
MDVIEWER=
MDMODEL=
`;
```

SET_OLAP_CLASSES accepts the following named parameters:

Key	Description
MODMGR	The 4-level name of the model manager class.
MODMGRE	The 4-level name of the model manager engine class.
EMDDB_C	The 4-level name of the model coordinator class.
DP	The 4-level name of the data provider class.
MDVIEWER	The 4-level name of the OLAP metadata viewer class.
MDMODEL	The 4-level name of the OLAP metadata model class.

Appendix K, Implementing SAS/SHARE[®] Software

For documentation on implementing SAS/SHARE Software, refer to *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software*.

Note: In order to run a secure SAS/SHARE server using the TCP/IP access method, you must provide a user exit for authenticating userid/password combinations.

Also note that the server will require authorization, either as a CP privilege class, or through your external security manager, in order to be able to authenticate passwords.

The password authentication user exit is not used for the IUCV and APPC access methods. There is no need to authenticate IUCV connections, and APPC authentication is performed by APPC.

For any access method, you can optionally provide another user exit for controlling file access authorization. The default, if you do not provide this exit, is to permit access to all files to any connected client.

User exits are documented in the appendix “User Exits in the SAS System under CMS” on page 113.

The sample user exit file `SASUSER ASSEMBLE` includes three sample password verification exits (vector slot number 62). One interfaces with Sterling Software’s VMSECURE security product. To activate this routine, change slot 62 from A(0) to A(RACFPW). The third sample uses CP’s diagnose 84 to verify passwords. To activate this routine, change slot 62 from A(0) to A(DIAG84).

Client-Side Components

SAS/SHARE software includes client components that are used outside of your SAS installation. These components are listed below:

SAS/SHARE Data Provider

The SAS/SHARE data provider enables you to access, update, and manipulate SAS data using OLE DB- and ADO-compliant applications on Windows platforms.

SAS ODBC Driver

The SAS ODBC driver enables you to access, update, and manipulate SAS data from ODBC-compliant applications on Windows platforms.

SAS/SHARE Driver for JDBC

The SAS/SHARE driver for JDBC enables you to write applets, applications, and servlets that access and update SAS data. The Java Tools package that includes the SAS/SHARE driver for JDBC also includes the SAS/CONNECT driver for Java. If you are writing Java programs using these interfaces, you may also want to use the tunnel feature. This optional feature can be used with the Java applets you write to solve some common configuration problems

SAS/SHARE SQL Library for C

The SAS SQL Library for C provides an application programming interface (API) that enables your applications to send SQL queries and statements through a SAS/SHARE server to data on remote hosts.

Client Components

SAS/SHARE client components are available on the *SAS Client-Side Components* CD included with your SAS Software distribution. Please refer to this CD for more information.

Appendix L, MACLIBs on the Installation Tape

Several of the files on the SAS installation tape are CMS MACLIBs. A MACLIB is a collection of files, each of which forms a member in the MACLIB. MACLIB members always have fixed-length, 80-character records.

Use the `CMS MACLIB` command to add, replace, or delete members of MACLIBs. For example, to delete member `FREQ` from `EXAMPLE` MACLIB, enter the following:

```
MACLIB DEL EXAMPLE FREQ
```

Files that are to be put into a MACLIB should have a filetype of `COPY`. For example, to add a file `TEST SAS` to `SAMPBASE` MACLIB, first, copy (or rename) `TEST SAS` to `TEST COPY`. Then enter the following:

```
MACLIB ADD SAMPBASE TEST
```

To replace an existing member `FREQ` in `SAMPBASE` MACLIB with a new copy of `FREQ SAS`, copy (or rename) `FREQ SAS` to `FREQ COPY`. Then enter the following:

```
MACLIB REP SAMPBASE FREQ
```

The `CMS MACLIB` command can also be used to create new MACLIBs, compress existing MACLIBs, and list the members in a MACLIB.

The `CMS MOVEFILE` command can be used to create a separate file from each member in a MACLIB. For example, suppose `EXAMPLE` MACLIB has three members, `REG1`, `REG2`, and `REG3`. To copy these members into separate files having the filetype `DATA`, enter the following:

```
FILEDEF INMOVE DISK EXAMPLE MACLIB *  
FILEDEF OUTMOVE DISK DUMMY DATA filemode  
MOVEFILE (PDS
```

where `filemode` is the filemode of the minidisk on which the new files are to be written. The filetype used in the `OUTMOVE FILEDEF` determines the filetype of the newly created files.

An exec has been provided on the SAS installation tape that performs this operation. The exec, called `UNMACLIB EXEC`, is installed on the system disk. To use this exec, enter:

```
UNMACLIB filename filetype location
```

where `filename` is the filename of the MACLIB, `filetype` is the filetype you want the loose files to be (`DATA` in the example above), and `location` is either a filemode or an SFS directory name where the new files are to be written.

To use this exec to remove selected members of a MACLIB, enter the following:

```
UNMACLIB filename filetype location (MEMBER names)
```

where `names` is a list of one or more members to be copied.

For more information about the `CMS MACLIB` and `MOVEFILE` commands, refer to *IBM VM/ESA: CMS Command Reference*.

Appendix M, CMS Storage Dumps

When a CMS SAS System task abends, it generates a message similar to:

```
Illegal instruction in task [MAIN]
```

or

```
Segmentation violation in task [MAIN]
```

and the return code from the SAS System is set to 16.

To resolve such problems, you should send SAS Institute a storage dump of the virtual machine at the time the abend occurred. To obtain a suitable storage dump, first issue these CP commands:

```
SET RUN OFF  
TRACE PROG
```

Then, rerun the program until the abend occurs. A line similar to:

```
*** 010FEBE6    PROG      0005 -> 00F3DC88    ADDRESSING
```

is displayed and the status area shows the CP READ message.

To obtain a storage dump, issue the following command:

```
VMDUMP 0-END DSS FORMAT CMS * brief statement of problem
```

This creates a dump file in your virtual reader. Use the DUMpload command to copy the dump to disk.

When you send SAS a storage dump, please include a copy of the program, the SASLOG, the terminal console log, and the data used in the program.

Appendix N, User Exits in the SAS® System under CMS

Using User Exits

User exit points are provided to support certain kinds of monitoring or control of a SAS session. For example, the SAS initialization and termination exit points make it possible for you to provide an exit to send accounting records at the beginning and end of each SAS session. The `userid` authentication exit point allows you to interface SAS with your particular security manager. Most exits are optional, but you must provide a `userid` authentication exit in order to support TCP clients on secured servers, and you must provide a file authorization exit if you wish to exercise control over which files a user may access via a server.

SAS is shipped with a default user exit load module named `SASUSER`. The source code for this module is provided in the file `SASUSER ASSEMBLE` installed on the SAS System disk. The sample source includes several example exits that are disabled. You can use the sample exits simply by enabling them, or you can code your own exits.

Note: The set of supported exits, some details of the programming interface, and the sample exits have changed between SAS releases and could change again in the future. If you have exits that you have used with a previous SAS release you should review them for compliance with the current interface.

Available User Exit Points

The following user exit points are available:

- ❑ SAS Initialization. This exit is called at initialization of the SAS System.
- ❑ SAS Termination. This exit is called at termination of the SAS System.
- ❑ Task Initialization. This exit is called at initialization of each task performed in the SAS System. Procedures and the data step are examples of tasks, among others.
- ❑ Task Termination. This exit is called at termination of each task in the SAS System.
- ❑ Physical Image Load. This exit is called each time SAS loads an image. If you have installed the SAS System in saved segments, you may want to determine how often the modules in the saved segments are being used. The Physical Image Load exit point allows you to record the names of each SAS module

loaded. However, you may prefer instead to use the `SGSTAT` option described on page 41 in Chapter 3, "Installing the CMS SAS System in Saved Segments."

- ❑ **File Authorization.** This exit is called by a server to determine whether an authenticated client is authorized to access a specified CMS file.
- ❑ **User Authentication.** This exit is called by a server to verify the `userid/password` of a client that is connected via the TCP access method. IUCV and APPC clients are authenticated by the operating system so this exit is not called for clients connected via those access methods.

User Exit Source Code Specifications

The following are the exit source code specifications:

- ❑ A user exit module must contain a vector of pointers to the entry points of the separate exits. Use a value of '0' in the vector as a placeholder for unimplemented exits. The main entry point of the module must place the address of this vector in general register 15 and then return. The main entry point is called one time only, early in SAS initialization.

Please see the sample source code in `SASUSER ASSEMBLE` for model code for the main entry point. You can implement any of the example exits by placing its address in the appropriate slot in the exit vector.

- ❑ All code in user exits must be reentrant and reusable.
- ❑ When an exit is entered, it must save the general purpose registers of the calling program, and when the exit terminates, it must restore the saved register contents prior to returning control to that program.
- ❑ Although you can create your own program linkage, we recommend that you use the `UXPRLOG` and `UXEPLOG` macros and the `DSA DSECT` that are provided in `SASUSER ASSEMBLE` and used in the sample exits.
 - The `UXPRLOG` macro (User eXit PRoLOG) is used to define the entry point of your user exit. `UXPRLOG` allocates a `DSA` (Dynamic Storage Area, mapped by the `DSA DSECT`) that includes space for a standard register save area, as well as additional space for automatic local variables if needed. It saves the registers, defines its label as an entry point, defines a base register for the user exit code, and loads the `DSA` address in register 13.

These are the valid `UXPRLOG` parameters (all optional):

```
BASE=reg, DSA=length
```

`BASE=` specifies the base register for the user exit. The default is `BASE=5`. The address of the entry point will be loaded into this register.

DSA= specifies the amount of storage to be allocated for the DSA. The default is DSA=*, which allocates the minimum storage for just a save area. To include space for automatic local variables in your DSA, define the space at the end of the DSA DSECT, define an EQUATE to specify the length (in bytes) of the DSA including the extra storage, and specify that EQUATE on the DSA= parameter. SASUSER ASSEMBLE includes several examples of doing this. For addressability to this storage, include a USING DSA,R13 statement in your user exit code.

- The UXEPLOG macro (User eXit EPiLOG) is used to define the return point of your user exit. UXEPLOG restores the registers, sets a return code, drops the base register, and returns to the calling program.

These are the valid UXEPLOG parameters:

BASE=reg,RC=

BASE= specifies the base register to be dropped. The default is BASE=5.

RC= specifies the return code. You must specify this parameter. The return code can be specified either as a register (in parentheses) that contains the return code, e.g., RC=(R15) or it can be specified as an integer value, e.g., RC=0.

- Entry conditions. When a user exit is called, register 1 contains a pointer to a control block that is mapped by the WEHEADER DSECT, which is included in SASUSER ASSEMBLE:

```
WEHEADER DSECT
CODE      DS      F
TASKNAME  DS      CL8
USERID    DS      A
USERLEN   DS      F
LOGPTR    DS      A
EXITGCB   DS      A
EXITCB    DS      A
```

CODE

is an integer value that corresponds to the slot position in the exit vector. It is possible to use a common entry point for more than one user exit, then use this value to determine which exit has actually been called.

TASKNAME

is the name of the active task

USERID

is a pointer to the name of the userid running the SAS session.

USERLEN

is an integer specifying the length of the name pointed to by USERID.

LOGPTR

is a pointer to a function that writes a null-terminated string to the SAS LOG. Call via BALR 14,15 with R1 pointing to a one-word PLIST containing the address of the string.

Note: Messages to the log are queued so the physical display of the message may not yet have occurred upon return from this call.

EXITGCB

is a location where a userword can be stored by exit code. It is initialized to '0' and not further modified by SAS code. If an exit updates the value, the updated value is passed on subsequent exit calls.

EXITCB

is a pointer to an exit-specific control block. See SASUSER ASSEMBLE for the structure of this control block for each supported exit. Only the Physical Image Load, File Authorization, and User Authentication exit points supply an EXITCB; for the other exit points EXITCB is either '0' or a pointer to '0'.

These are the supported EXITCB structures:

- Physical Image Load exit point

```
IMAGENAME DS CL8 Name of image being loaded
```

- File Authorization exit point

```
FNPTR DS A pointer to a (null terminated) complete CMS
* file specification, including filename, filetype,
* and filemode or directory name.
FNLEN DS F length of the file specification pointed to by
* FNPTR (excluding null)
WEXUSER DS CL8 userid whose authorization for the file is to be
* checked
WEXMODE DS F type of access requested for WEXUSER
READACC EQU 1 Read access
WRITACC EQU 2 Write access
RDWRACC EQU 3 Read/write access
```

- User Authentication exit point

```
SECUSERL DS XL1 actual length of SECUSER
SECUSER DS CL8 userid (blank padded to 8 bytes)
SECPWL DS XL1 actual length of SECPW
SECPW DS CL8 password (blank padded to 8 bytes)
```

- ❑ Return codes. Normally a user exit should supply a return code of '0'. The authentication and authorization exits should return '2' when the request is rejected. You can also use return code 1 to cause the SAS session to be aborted.

How to Create a User Exit Load Module

Once you have coded your user exits, you must assemble the source code and create a load module.

Before assembling the exit module, GLOBAL any required MACLIBs. Assembly of the original sample SASUSER ASSEMBLE requires the DMSGPI and OSMACRO MACLIBs.

Assuming that your source file is named SASUSER ASSEMBLE, you can use these commands to assemble it and then linkedit the TEXT file into a LOADLIB member named SASUSER in SASUSER LOADLIB:

```
GLOBAL MACLIB DMSGPI OSMACRO
ASMAHL SASUSER
LKED SASUSER ( MAP
```

How to Test Your User Exits

To use your user exits, you must make your load module available to the SAS System. There are two ways to do this in test mode (i.e., without replacing the production user exit load module).

1. If both your production and test user exit load modules are named SASUSER, you must ensure that SAS finds your test version before it finds the production version. To do this you must specify two system options, either on the command line or in the config file. Assuming that you link-edited your module into SASUSER LOADLIB, use option SASLOAD='SASUSER LOADLIB *' to cause SAS to search SASUSER LOADLIB before it searches other loadlibs. If you have SAS installed in saved segments, also use option NOSEGL=SASUSER to keep SAS from using SASUSER from segments and to load it from a loadlib instead.

For example, issue the following command:

```
SAS (SASLOAD='SASUSER LOADLIB *' NOSEGL=SASUSER
```

2. You can give your user exit load module a name other than the default SASUSER, and tell the SAS System to use your test module via the USEREXITS= option. If, for example, you name your user exit load module TESTUSER for testing, and you linkedit it into TESTUSER LOADLIB, you can issue the command

```
SAS (SASLOAD='TESTUSER LOADLIB *' USEREXITS=TESTUSER
```

Note that the value specified on USEREXITS= is the name of a loadlib member, which need not be the same as the name of the loadlib that contains it. You need both the SASLOAD= and USEREXITS= options to specify both the loadlib and the member.

How to Make the User Exit Load Module a Permanent Part of the SAS System

Once you have tested your user exit module, in order to make it a permanent part of the SAS System, you will have to modify the `SASBASE LOADLIB` and resave the SAS System segments.

We recommend that you name your production user exit module `SASUSER` in order to avoid the need to specify the `USEREXITS=` option in your system config file, although you can do so if you choose.

Copy the `SASUSER` module from the `SASUSER LOADLIB` into your production `SASBASE LOADLIB`. You will then not need a separate `SASLOAD=` option (because there is already a `SASLOAD=' SASBASE LOADLIB * '` in your system config file). To copy the load module, link to the SAS System minidisk with write access and issue the following commands:

```

Computer  You
Prompts:  Enter:
-----  -----
Ready     LOADLIB COPY SASUSER LOADLIB fm SASBASE LOADLIB fm
          (REPLACE
ENTER     SELECT
ENTER     SASUSER(R)

```

After entering `SASUSER(R)`, you will see another prompt. Simply press the `ENTER` key in response to this prompt, to terminate the `LOADLIB` command.

Compress `SASBASE LOADLIB`. Since the `LOADLIB COMPRESS` command creates a copy of the `loadlib`, be sure that you have enough room on your minidisk to hold temporarily a second copy of the `SASBASE LOADLIB`. To compress the `loadlib`, issue the following command:

```
LOADLIB COMPRESS SASBASE LOADLIB fm
```

SASUSER and Saved Segments

If you have installed the SAS System in saved segments, then each time you update your `SASUSER` load module you must resave the segments.

If you have installed the SAS System in saved segments, but prefer not to include `SASUSER` in the saved segments, edit the `ALLSAS LLIST` file before resaving the segments. Locate the following line:

```
SASBASE LOADLIB * SASUSER      (S)
```

To have `SASUSER` omitted from the segments, place an asterisk (*) in the first column of that line. For `SASUSER` to be included in the segments, the first column must be blank.

Federal Government Rights Notice

If your installation is a federal government site or a federal government prime contractor site, contractual requirements may have included a usage rights notice, which you should examine. This notice is file `GOVT NOTE`, and is loaded to disk automatically at installation.

Please review the rights notice and provide a way for other users at your installation to review it also. For example, you could add the notice as a `NEWS` item or use the `ECHO=` option telling your users to read the file. Please see the appropriate section of the installation process or consult the host documentation for the choice that is best for your site.

CMS SAS® Installation Feedback Form

We can make the SAS System even easier to use and install with your help. Please complete the following installation feedback form and return to the address listed below.

Installation: _____

Installer: _____

Products Installed:

How long did it take you to load SAS Software from the tape ? _____

How many hours did it take you to enable the SAS System so that you could use it? _____

Did you install the SAS System in segments? _____

How long does it take you to convert to a new release? _____

How many hours was it before your users could use the new release? _____

What level of CMS are you running ? _____

If you made any performance comparisons, please share your figures below:

Additional comments on the instructions, installation, segments, etc.

Return to:

SAS
CMS Systems Development
SAS Campus Drive
Cary, NC 27513-2414