Installation Instructions and System Manager's Guide for the SAS® System under CMS, Release 6.09 Enhanced, TS465 and above

Table of Contents

Chapter 1, Introduction and Orientation

How This Book is Organized

This document contains instructions to guide you through your installation of Release 6.09 Enhanced 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. 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* gives a concise list of the installation steps.

  - *Preinstallation 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 Using the Installation Aids* 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* provides 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.

- ❑ *Tailoring the SAS Notes* 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 appropriate for the products you install.

Note: Be sure to read the *Cover Letter, System Requirements,* and *Alert Notes* that are included in your software package. Keep these items and this document as a 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 off with the installation aids. It can be useful to search for special instructions; it is not formatted or meant to be printed.

Installation Objectives

Your installation objective is one of the following 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 6.09 Enhanced site including established SAS System site running a release prior to 6.09 Enhanced

  You want to install Release 6.09 Enhanced of the SAS System under CMS for the first time. Your tape contains Base SAS software, all other products you have licensed, and current SAS Notes. The external label on the tape reads "PRODUCT & SAS NOTES."

  SAS Notes are a collection of reports on current SAS software problems, compatibility issues, and common user problems.

  Even if you already have a prior release of the SAS System installed on CMS, you need to install Release 6.09 Enhanced to 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 System site wanting current SAS Notes

  You currently have Release 6.09 Enhanced of the SAS System under CMS installed and you have requested the current SAS Notes. Your tape contains SAS Notes. The external label on the tape reads "SAS NOTES."

If this is your objective, you should finish reading this chapter and then follow the instructions in Chapter 4, *Maintaining the SAS System Under CMS*.

❑ Established SAS System site requesting additional products

You currently have Release 6.09 Enhanced of the SAS System under CMS installed and you want to install an additional SAS System product. Your tape contains the new product. It also contains current SAS Notes. The label on the tape reads "PRODUCT & SAS NOTES."

If this is your objective, you should finish reading this chapter and then follow the instructions in Chapter 4, *Maintaining the SAS System Under CMS*.

Tape Contents

The tape that accompanies this document contains Release 6.09 Enhanced of the SAS System under CMS. The installation tape is a non-labeled tape that was created under VM/SP Release 5 using IBM's `TAPE DUMP` command and `BLOCKTAP`.* The tape's external label indicates the density of your tape.

Note: This document refers to the installation tape in the singular. It is one logical tape, although it can extend over two or more physical reels. 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 software installation aids from the tape, and with the tape still attached as virtual device 181, issue:

```
GETSAS TAPEMAP
```

Detailed instructions for doing this are covered later in this document. If you do not have the correct tape, contact the Distribution Center at SAS Institute 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.

* `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. Many thanks to Steve for his good work.

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. It is what you run to load the rest of the SAS System from tape to disk.

- 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.

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

  Installation verification programs are also in this section. These test programs are described later in this document in Part IV, *Customizing and Verifying the SAS System*, in the section, *Installation Verification Testing* of Chapter 2.

- 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 need not 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. Refer to the SAS Technical Report P-196, *SAS/GRAPH Software: Map Data Sets, Release 6.06* for more information.

- TUTOR

  If your site has licensed one or more SAS/TUTOR software courses, the courses will be placed in this section.

- SAS NOTES

  This section contains the SAS Notes, documentation notes, and the SAS programs needed to process the SAS Notes. SAS Notes are a collection of reports on current SAS software problems, compatibility issues, and common user problems. This section can also contain files for short-term emergency software maintenance, most commonly in the form of ZAPs.

References

The following SAS Institute manuals can assist you and your users with Version 6 of the SAS System. These manuals can also provide helpful information for completing your system set up and product implementation:

- *SAS Language: Reference, Version 6, First Edition*
- *SAS Procedures Guide, Version 6, Third Edition*
- *SAS Language and Procedures: Usage, First Edition*
- *SAS Language and Procedures: Usage 2, First Edition*
- *SAS Companion for the CMS Environment, Version 6, First Edition*
- SAS Technical Report P-268: *Changes and Enhancements to the SAS System through Release 6.09 Enhanced, for the CMS Environment*
- *SAS/GRAPH Software: Reference, Version 6, First Edition, Volumes 1 and 2*
- *SAS/GRAPH Software: Using Graphics Devices in the CMS Environment, Version 6, First Edition*
- SAS Technical Report P-196, *SAS/GRAPH Software: Map Data Sets, Release 6.06*
- SAS Technical Report P-197, *The TRANTAB Procedure, Release 6.06*
- *SAS/CONNECT Software: Usage and Reference, Version 6, Second Edition*
- *SAS/ACCESS Interface to SYSTEM 2000 Data Management Software: Usage and Reference, Version 6, First Edition*

Contacting SAS Institute

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

Chapter 2, Installing the SAS® System Under CMS

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

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

Note: The install procedure for Release 6.09 Enhanced under CMS has been rewritten. New functionality includes automatic calculation of disk space requirements and the ability to have answers from a previous invocation used as the defaults. Read all of the *Installation Instructions* before attempting to install SAS software.

- ❑ Review all the items in your product package.
- ❑ Ensure your operating system and hardware meet the specifications indicated in the *System Requirements* contained in your package.
- ❑ Obtain write access to a small amount of clean disk space to hold the installation aids.
- ❑ Mount the tape and have the tape drive attached to the userid.
- ❑ Load the SAS software installation aids (including the GETSAS EXEC) from tape to disk using TAPE LOAD.
- ❑ Determine the disk space requirements for CMS SAS software. This can be done by running the GETSAS EXEC.
- ❑ Obtain the required resources to hold the SAS System, including clean disk space to which you have write access.
- ❑ 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.
- ❑ Enable the SAS Autocall Macro Libraries.
- ❑ Run CMS SAS programs for installation verification.
- ❑ Edit and customize the SAS EXEC.
- ❑ Review and customize installation defaults for SAS System options.

- ❑ Customize catalog members used globally by users.
- ❑ Review and customize the SITEINFO file.
- ❑ Review and customize the NEWS file.
- ❑ Add Help File to CMS HELP Facility.
- ❑ Review the information on sample libraries.
- ❑ Tailor the SAS Notes (optional). See the section *Tailoring the SAS Notes* in Chapter 4 for more information.
- ❑ Perform product-specific installation tasks:
    - ❑ If you have IBM's SQL/DS software and SAS/SQL-DS software, link the IBM SQL/DS resource manager stub routine with the SAS/SQL-DS software. See Appendix A, "Tailoring the SAS/ACCESS Interface to SQL/DS."
    - ❑ If you have SYSTEM 2000 software, tailor the SASS2K EXEC. See Appendix B, "SAS/ACCESS Interface to SYSTEM 2000 Software."
    - ❑ If you have SAS/ASSIST software, see Appendix C, "Post-Installation Setup for SAS/ASSIST Software."
    - ❑ If you have SAS/CONNECT software, download the SAS/CONNECT scripts and configure for APPC support. See Appendix D, "Post-Installation Setup for SAS/CONNECT Software."
    - ❑ If you have SAS/GRAPH software, tailor SAS/GRAPH software if required. See Appendix E, "Tailoring SAS/GRAPH Software."
    - ❑ If you have SAS/ACCESS to ORACLE software, see Appendix F, "Implementing SAS/ACCESS Interface to ORACLE."
    - ❑ If you have SAS/SHARE software, prepare the server machine. See Appendix G, "Implementing SAS/SHARE Software."
    - ❑ If you have SAS/TOOLKIT software, see Appendix H, "Implementing SAS/TOOLKIT Software."
    - ❑ If you have any SAS/TUTOR software courses, prepare startup information. See Appendix I, "Installing the SAS/TUTOR Software Courses."

    Note: The remaining SAS software products do not have product-specific installation instructions.
- ❑ If you need to monitor or control execution of the SAS System, see Appendix J, "User Exits in the SAS System under CMS."

- ❑ Create the CMS SAS saved segments. Segment usage has changed for Release 6.09 Enhanced and the documentation has been rewritten. See Chapter 3 for more information. We strongly encourage you to install your SAS software products in saved segments.

- ❑ Rerun CMS SAS programs for installation verification.

- ❑ Read the file GOVT NOTE (U.S. Federal Government employees only). This section appears 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 6.09 Enhanced will need to be remade.

After you have completely finished the installation, you can erase the installation aids. The installation aids are specific to the tape they are on, so the installation aids from this tape should not be used to install another tape.

Part II, Preinstallation 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 Institute 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 Installation Representative who in turn may need to contact SAS Institute.

- ❑ Ensure your operating system and hardware meet the specifications indicated in the *System Requirements* contained in your package.

- ❑ Obtain write access to a small amount of clean 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.

  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 non-labeled 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 are 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. New with the CMS install for Release 6.09 Enhanced 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* or the screen output from issuing:

```
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* 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, SAS/TUTOR, and SAS Notes).

  Notes:

  - When you install Release 6.09 Enhanced of the SAS System, you can install everything off your tape or you can select only certain sections of the tape (SYSTEM, SAMPLE, MAPS, SAS/TUTOR, and SAS Notes). You can also select specific products. This includes specific courses if you license SAS/TUTOR software and 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, SAS/TUTOR, and SAS Notes) to separate locations. Any combination of sections can be grouped into separate locations. You should not, however, install different products to different locations. The only exception is SAS/TUTOR software, which *should* go to a separate location. For example, you could put the SYSTEM files on one disk, the SAMPLES, MAPS, and SAS Notes on a second disk, and SAS/TUTOR files on a third disk.

- ❑ Obtain the required disk space. You need to have write access to this disk space and the disk space should be clean. 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* 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*.

  Note: Throughout this document, references to CMS minidisk can be substituted with 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, "Preinstallation Considerations." For example, you need to load the SAS installation aids (including the GETSAS EXEC) from tape to disk and obtain write access to obtain sufficient disk space to hold the SAS System.

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 non-labeled 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
```

Answer the prompts and let GETSAS do the rest.

The GETSAS EXEC has been rewritten for Release 6.09 Enhanced. Some new features include:

- ❑ 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.
- ❑ 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 6.09 Enhanced is a *new* release, it is not a 6.08 maintenance release. If you already have a prior release of the SAS System installed on CMS, install Release 6.09 Enhanced 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 6.09 Enhanced 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 add-ons) 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 6.09E TS 450                                              1



      Thank you for licensing SAS Software products for the CMS system.

           Copyright(c) 1996 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 prompts you for information including which products you want to install and which sections of the tape you want to install (SYSTEM, SAMPLE, MAPS, SAS/TUTOR, and SAS Notes). By default, all sections and products are installed. On the product, sample, and tutor selection screens, select F6 to display a description of each product.

Some SAS software products require other SAS products in order to run. For example, SAS/EIS software requires SAS/AF software, SAS/FSP software, and SAS/GRAPH software. If you attempt to install a product without the other required products, GETSAS displays an error message similar to the following:

```
SAS/EIS requires SAS/AF to function.
```

In this case, you need to install the required product too.

GETSAS then shows you the disk space requirements based on your selections, and prompts you for the disks to which you want the SAS System 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. Type `RETURN` and press `Enter` to resume GETSAS.

Any time you run GETSAS, your answers to the prompts are kept even if you exit one of the screens (F3). The next time you run GETSAS, 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 GETSAS displays. The files are loaded to disk only if you continue on from this screen:

```
CMS SAS Release 6.09E NEW TS 450                                        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...
 0CORE SASHELP        D1
 End-of-file or end-of-tape
 Loading...
 SASBASE     LOADLIB  D2
 SASBASE     LOADLIB  D2
 SASSMSG     ASM      D1
 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 VERIFY
 . . . ADDING FILEMAC
 THE SASAUTOS MACLIB IS COMPLETE.
...COMPRESSING SASAUTOS MACLIB
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 recommend 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 detached or dismounted).

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.

For post-installation tasks, see Part IV, "Customizing and Verifying the SAS System" and Chapter 3, "Installing the CMS SAS System in Saved Segments." 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 rewound.
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. 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. 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 where you issued help.

Rerunning GETSAS

After having run GETSAS to load files from tape to disk, you may sometimes find a need to rerun GETSAS. Typical reasons for doing this include wanting a product or sample that you did not previously select or reinstalling a product that has been erroneously modified. In situations like this, 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 Release 6.09 Enhanced CMS installation. This includes access to locations containing any SYSTEM, SAMPLE, MAPS, SAS/TUTOR, and SAS Notes files. One exception is that if you want to load SAS Notes only, you only need write-access to your SAS Notes 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 Part IV, "Customizing and Verifying the SAS System" and Chapter 3, "Installing the CMS SAS System in Saved Segments."

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. If you do, subsequent installs (for reruns, maintenance, or add-ons for example) will fail.

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.

  Two ways you can ensure that you have enough disk space are:

  - Erase any files that you plan to reload that are bigger than what is currently available 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, the only files needing extra space are:

Base SAS Software	SASBASE	LOADLIB	5040 4k blocks
	0FSP	SASHELP	1071 4k blocks
SAS/GRAPH Software	SASGRAPH	LOADLIB	1230 4k blocks
	COUNTIES	MAPS	2943 4k blocks
SAS/STAT Software	SASSTAT	LOADLIB	2331 4k blocks
SAS/ASSIST Software	0ASSIST	SASHELP	1498 4k blocks
SAS/EIS Software	0EIS	SASHELP	1097 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.
      or GETSAS was issued from within CMS SUBSET.
      or Unrecognized tape type XXXX.
      or Invalid GETSAS TAPEMAP file (possible invalid SAS
         distribution tape).

28    Required installation aids file not found or not unique.
      or Tape not attached as 181 or wrong tape.
      or 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 software) come with commonly used and helpful macros. These macros are shipped by product in macro libraries and referred to as autocall macros.

When you run GETSAS, after everything you have 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 SYSPROF SAS 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 [1]own MACLIB and add the name of the MACLIB to the SASAUTOS= option. For example:

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

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

---

*Prior to Release 6.09 Enhanced, the individual product MACLIBs had to be merged into the SASAUTOS MACLIB by manually running the SASAUTOSEXEC as a post-installation task.

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:

TESTBASE	SAS	test job input for Base SAS procedures
TESTORL	SAS	test job for SAS/ACCESS to ORACLE*
TESTETS	SAS	test job for SAS/ETS software
TESTGRAF	SAS	test job for SAS/GRAPH software**
TESTIML	SAS	test job for SAS/IML software
TESTIMLG	SAS	test job for SAS/IML software with graphics
TESTOR	SAS	test job for SAS/OR software
TESTQC	SAS	test job for SAS/QC software
TESTS2K	SAS	test job for SAS/ACCESS for SYSTEM 2000***
TESTSQD	SAS	test job for SAS/ACCESS for SQL/DS****
TESTSTAT	SAS	test job for SAS/STAT software

Note: When running these test programs, link the SAS System disk in read-only mode. This will prevent the SASLOG and LISTING files from being created on the SAS System disk. If they have been created on the SAS System disk, delete them.

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

```
SAS TESTBASE
```

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 6M. Note that the programs can take up to 20 minutes to execute. If a test program completes with a clean return code, the test was successful.

---

* See Appendix F, "Implementing SAS/ACCESS Interface to ORACLE," before running TESTORL.

** Read the comments in the TESTGRAF program before executing it.

*** See Appendix B, "SAS/ACCESS Interface to SYSTEM 2000 Software," before running TESTS2K.

**** See Appendix A, "Tailoring SAS/ACCESS Interface to SQL/DS," before running TESTSQD.

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.

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

The SAS EXEC supplied with this release of the SAS System is different from the SAS EXEC from previous releases. Do not attempt to use an old SAS EXEC to invoke this release of the SAS System. If you need modifications from an old SAS EXEC, be sure you transfer them to the new exec. In addition, you cannot execute the SAS System from an old style exec. Old style execs begin with `&CONTROL`.

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

Settings for SAS System Options

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

For descriptions of SAS System options, refer to the *SAS Language: Reference, Version 6 First Edition* and *SAS Companion for the CMS Environment, Version 6, First Edition,* and SAS Technical Report P-268, *Changes and Enhancements to the SAS System for the CMS Environment Through Release 6.09 Enhanced*. 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 found below on customizing the SYSPROF SAS file.

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

Customizing SYSPROF SAS

SAS Institute provides two files for changing SAS System options, SYSPROF SAS and CONFIG SAS. The SYSPROF SAS file is part of the SAS System disk and should only be changed by the site representative. A CMS editor can be used to edit SYSPROF SAS, which contains one or more free-format, 80-byte records. The options placed in this file affect all SAS users, whereas options placed in a CONFIG SAS file affect a user only when the file is found on a linked minidisk.

The LOADLIBs used by the SAS System are identified in the SYSPROF SAS 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 SYSPROF SAS, put an asterisk in column one.

Customizing CONFIG SAS

Users can create a CONFIG SAS 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 CONFIG SAS file and the SYSPROF SAS file, SAS uses the value specified in the CONFIG SAS file.

In releases of SAS software product prior to Release 6.06, all PROFILE SAS files were concatenated and used. However, this is not true for the CONFIG SAS and SYSPROF SAS files. Refer to the *SAS Companion for the CMS Environment, Version 6* for more information on the CONFIG= option and the CONFIG SAS file. To assist you in your own configuration, an example of a CONFIG SAS file is on the installation tape.

Note: When using an option in the SYSPROF or CONFIG SAS 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 routine, SASSORT, as part of the SAS System under CMS. The Institute 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 options available to you and decide which performs best in your environment. If your installation leases SYNCSORT, 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 routine that is efficient for data sets with fewer than 2500 observations.

Important: If you use CASORT with the SAS System, be sure to rename or delete the file SORT TEXT that comes with CASORT so that it is not used.

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

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

```
SORTPGM=SAS for SASSORT
SORTPGM=SORT for SYNCSORT
SORTPGM=SORT for VMSORT
SORTPGM=CASMA001 for CASORT
SORTPGM=SORT for DFSORT
```

or

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

The values of 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 globaled, 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.
```

The SAS System is sent with 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, Version 6, First Edition* and SAS Technical Report P-268, *Changes and Enhancements to the SAS System for the CMS Environment Through Release 6.09 Enhanced*.

Note: Messages from host sort programs can be controlled with the SORTMSG system option.

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

The Institute-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 need to modify any of the Institute-provided tables or create new tables, refer to SAS Technical Report P-197, *The TRANTAB Procedure, Release 6.06.*

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.

Display Manager 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 programmable function keys or key definitions, as defined by IBM's Common User Access (CUA). Acceptable values are `PRIMARY, ALTERNATE, 12, CUA, CUAALT,` and `CUA12`. 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 settings (window management)
`ALTERNATE`	PF13 - PF24	=	Version 6 settings (window management)
	PF1 - PF12	=	Version 5 settings (command keys)
`12`	PF1 - PF12	=	Version 5 settings with only 12 function keys active
`CUA`	PF13 - PF24	=	defined by CUA
	PF1 - PF12	=	Version 5 settings (command keys)
`CUAALT`	PF13 - PF24	=	Version 5 settings (command keys)
	PF1 - PF12	=	defined by CUA
`CUA12`	PF1 - PF12	=	defined by CUA
	PF13 - PF24	=	match PF1 - PF12

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

Using the IBM 3090 Vector Facility with the SAS System

If you have an IBM 3090 with the Vector Facility, some SAS procedures, most notably PROC GLM, uses the instructions provided for this facility. Specify the `NOVECTOR` SAS System option to disable use of the facility.

Customizing Globally Used Catalog Members

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

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

Customizing Default Forms

As the SAS consultant for your site, you can customize the `form` window displayed for your operating system. In addition to the information given below, read the information available in the *SAS Language Guide* 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 Display Manager 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.

To find out more about customizing the Print File Parameters program, issue the following command from the Display Manager:

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

You can print the CBTs by issuing the following program statements:

```
PROC BUILD C=SASHELP.BASE BATCH BROWSE;

             PRINT DISPLAY SELECT=PDEVICE.CBT; RUN;

PROC BUILD C=SASHELP.BASE BATCH BROWSE;

             PRINT DISPLAY SELECT=FORMSCPH.CBT; RUN;
```

or

```
PROC BUILD C=SASHELP.BASE BATCH BROWSE;

             PRINT DISPLAY SELECT=(PDEVICE.CBT FORMSCPH.CBT);
             RUN;
```

To route the output to a file, issue the following commands:

```
PROC BUILD C=SASHELP.BASE BATCH BROWSE;

          PRINT DISPLAY SELECT=(PDEVICE.CBT FORMSCPH.CBT)
          PRTFILE='file name'; RUN;
```

Customizing Default Display Manager Keys

You can globally change the programmable 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.

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 SITEINFO, NEWS, and SAS HELPCMS Files

The SITEINFO File

A file named SITEINFO SAS is provided to allow you to display site information from the SITEINFO window. The SAS Software Representative should edit this file to provide local information for users. Use SAS Institute's Display Manager System (DMS) or any other system editor to add this information.

The file currently contains labels for information that is useful to your user community, including names of consultants, training personnel, etc. A field is also designated for the Site Number and Site Name, which you must provide in all communications with SAS Institute Technical Support.

Any other information that is helpful or desirable for users can be added to this file. Activate the file by setting the `SITEINFO` option in the SYSPROF SAS file equal to the filename. If you change the name of this file, you must change the setting of the `SITEINFO=` system option in the SYSPROF SAS file. Edit the file and provide your users with site-specific information. To look at the results of your editing, issue the `SITEINFO` command from any display manager window command line.

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 SYSPROF SAS file, or add it to a user-defined CONFIG SAS 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).

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
SAMPAF	MACLIB	sample library for the SAS/AF product
SAMPETS	MACLIB	sample library for the SAS/ETS product
SAMPFSP	MACLIB	sample library for the SAS/FSP product
SAMPGRP	MACLIB	sample library for the SAS/GRAPH product
SAMPIML	MACLIB	sample library for the SAS/IML product
SAMPINSI	MACLIB	sample library for the SAS/INSIGHT product
SAMPOR	MACLIB	sample library for the SAS/OR product
SAMPQC	MACLIB	sample library for the SAS/QC product
SAMPRLIN	MACLIB	sample library for the SAS/CONNECT product
SAMPS2K	MACLIB	sample library for the SAS/ACCESS product for S2K
SAMPSQL	MACLIB	sample library for the SAS/ACCESS product for SQL/DS
SAMPTOOL	MACLIB	sample library for the SAS/TOOLKIT product
SAMPDBIO	MACLIB	sample library for the SAS/ACCESS product for ORACLE

These maclibs contain samples of SAS source code. In addition to these maclibs, the SAMPEXIT MACLIB is included with the base SAS product. This maclib contains assembler code used when creating user exits. Please see Appendix J, "User Exits in the SAS System Under CMS," for more information on this maclib.

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 Appendix K, "MACLIBs on the Installation Tape," for information about this exec.

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

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. Segment usage has changed for Release 6.09 Enhanced so this chapter has been rewritten.

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 one of the features 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, the 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 system there are at your site, the greater the benefits of running SAS system from saved segments.

We strongly recommend that the SAS System be installed in saved segments. The benefits for system performance can be summarized as:

- 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.

The implementation of saved segments is different in some respects between XA- and non XA-based VM releases. An XA-based system is any VM system whose CP supports the `DEFSEG` command. These include all VM/XA and VM/ESA systems (except for VM/ESA systems with the 370 feature). A non XA-based system is any VM system whose CP requires the use of a DMKSNT ASSEMBLE file. These include all VM/SP systems and VM/ESA with the 370 feature. Differences between XA- and non XA-based systems will be noted where relevant.

Beginning with Release 6.09 Enhanced of the SAS System, you can also install SAS libraries and the SAS message database in saved segments (on XA-based systems only). 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

On an XA-based system, you will execute the `DEFxSAS EXEC` generated by the `CMSSEGS` command. DEFxSAS will issue the `DEFSEG` commands to create skeleton segments.

On a non XA-based system, you will provide your VM Systems Programmer with the SASxSNT ASSEMBLE file generated by the `CMSSEGS` command. The VM Systems Programmer will need to incorporate this file into the system DMKSNT ASSEMBLE file and generate a new CP.

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

You will execute the `SASCSEG6` command, which will load the SAS System into the newly created segments and issue the `SAVESYS` or `SAVESEG` commands to save them.

5. Setting Option Defaults to use the Saved Segments

You will set options in SYSPROF SAS 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 three configuration files (SAS CMSSEGS, SASXA CMSSEGS, and ALLSEG CMSSEGS, which will be described in more detail later) each of which provides a particular set of defaults for some of these decisions. We recommend using one of these default sets unless some special situation requires doing otherwise. In any case, you should choose the appropriate one of these sample files, 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.

On non XA-based systems, the requirement that saved segments reside entirely below 16M while not conflicting with other saved segments means that it is not possible to install the entire SAS System in saved segments. The storage constraints are dealt with by loading only the most-used code in saved segments, and by using an overlay technique to map multiple SAS modules into the same address range. The SAS CMSSEGS sample control file uses this approach for non XA-based systems. The message database and SAS libraries can not be saved in segments on non XA-based systems.

On XA-based systems it is possible, and recommended, to install the entire SAS System in saved segments. The ALLSEG CMSSEGS sample control file uses this approach. However a more selective installation using overlays is also supported and the SASXA CMSSEGS sample control file uses this approach. If you have an XA-based system but the SAS System will be used by users with 370-mode virtual machines, you must install the SAS saved segments below the 16M line and thus should use overlays. If you have a mixture of 370-mode and XA-mode virtual machines, you could define two series of SAS segments, one below the line and one above. Defining multiple series will be further discussed below.

Release 6.09 Enhanced segments are larger than previous releases. More spool space may be consumed, especially if you save libraries in segments.

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 to be described later.

On non XA-based systems, saved segments must reside entirely below the 16M line and also must reside above the largest virtual machine that will use them.

On XA-based systems, saved segments can reside at any address. Also, segments do not necessarily need to reside above virtual machines that will use them.

If you have an XA-based system and 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. Requirements for this command to succeed when the address space of some of the saved segments would lie within the virtual machine include that none of the address space is already in use for another purpose, and that no segment can lie partly inside and partly outside the virtual machine. It is possible to have the SASxMAP1 and SASxSUPR segments within the virtual machine and the PROC segments outside the virtual machine, but it is not possible to have SASxMAP1 within the virtual machine and SASxSUPR outside, nor is it possible to have any other segment inside the virtual machine if SASxMAP1 is 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, it may require consultation between the SAS Installation Representative and the VM Systems Programmer to determine suitable address ranges. Examples of other products whose segments you might need to take into consideration are 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), or SQL/DS (if you are licensed for SAS/ACCESS to SQL/DS)

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. Or, as noted above, you might require one series for XA-mode users and another for 370-mode users. 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 a user runs the SAS System, which series is used is specified by the SERIES= option on the command-line, in a SYSPROF SAS file, or in a CONFIG SAS file.

In this document, 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, SASBMAP1, etc.

The naming convention for Version 6 SAS saved segments is different from that for Version 5 SAS saved segments so there is no potential for conflict between these SAS versions.

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

`GENXA` _____ Enter YES if the system is XA-based, NO if non XA-based.

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

The following four parameters are required only for non XA-based systems (`GENXA NO`) and pertain to the DASD volume on which the saved segments will be stored. You will need to consult with the VM Systems Programmer to obtain this information. For XA-based systems you can omit these parameters or leave their values as "?."

`DISK-TYPE` _____ The DASD type , e.g., 3370, 3380, etc.

`DISK-LABEL` _____ What is the disk label of the volume?

`DISK-CYL` _____ Cylinder at which the saved segments will begin (For CKD DASD only)

`DISK-PAGE` _____ Page at which the saved segments will begin.

The following parameters are supported only on XA-based systems:

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 M`.

`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. You must rename RXFILEIO -MODULE to RXFILEIO MODULE on your SAS System disk. It was installed with the file type -MODULE to avoid any possible conflict with an existing file on your system. If you like you can rename the file back to RXFILEIO -MODULE when you are done with CMSSEGS.

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 Institute 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.

Files generated by CMSSEGS command

The following files are generated by the CMSSEGS command:

- For non-XA based systems:

  SASxSNT ASSEMBLE

  where "x" is your chosen series letter. This is the file that you must send to your VM Systems programmer.

- For XA-based systems:

  DEFxSAS EXEC

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

- For all systems:

  1. SASxMAP1 MODULE
     contains data that will be loaded into a saved segment

  2. SASCSEGS LOADINFO
     control file required by the SASCSEG6 command

  3. CMSSEGS MSORTED
     work file used only by CMSSEGS

  4. CMSSEGS MEMBERS
     work file used only by CMSSEGS

5. CMSSEGS OVERLAY
   work file used only by CMSSEGS.

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 on *non XA-based systems*, send the SASxSNT ASSEMBLE file that CMSSEGS created to the VM Systems Programmer so the new segments can be defined to CP. You must wait for this to be completed before proceeding with the next step.

To create the saved segments on *XA-based systems*, complete the following steps:

Note: The DEFxSAS command accepts three parameters, QUERY, PURGE, or DEFINE.

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 (for example, switching between using overlays or not) 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 SAVESEG (XA) or SAVESYS (non-XA) commands. Normally the required privilege class is E. 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.

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. For XA-based systems 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.

You may need to `SET LDRTBLS` to a higher number than your default.

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

Execute the `SASCSEG6` command. This command will load data into storage and issue the SAVESEG (XA) or SAVESYS (non-XA) commands to save the segments.

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

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

Step 5: Setting Option Defaults to use the Saved Segments

Three system options control the use of saved segments, `SSEG`, `SERIES=`, and `SASHELP=`. If the `SSEG` option is specified, the SAS System will execute from code in saved segments. The `SERIES=` option specifies which segment series is used. If the `SASHELP=` option has a null value, the SAS System will use the SASHELP library from segments. `SASHELP=''` is the default so it does not have to be explicitly specified.

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 SYSPROF SAS. For example, if you have chosen "A" as your series letter, your SYSPROF SAS might include:

```
SSEG
SERIES=A
SASHELP=''
```

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 need only to force the SAS System not to read the SASHELP library from a segment, provide a library specification on the SASHELP option, for example `SASHELP=*`.

To cause user libraries in saved segments to be used, specify the SEGMENT=YES option on the LIBNAME statement. For more information on SSEG, SERIES=, SASHELP= and the SEGMENT=YES option on the LIBNAME statement, see the chapters on "SAS System Options" and "Optimizing Performance" in both *SAS Companion for the CMS Environment* and SAS Technical Report P-268, *Changes and Enhancements to the SAS System for the CMS Environment Through Release 6.09 Enhanced.*

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` or `ALLSEGA CMSSEGS`.

Access the SAS System disk.

Copy the prototype `CMSSEGS` file to your A disk, using your chosen filename,

e.g. `COPYFILE ALLSEG CMSSEGS B ALLSEGA = 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

non XA-based systems:

Send the `SASxSNT ASSEMBLE` file that `CMSSEGS` created to the VM Systems Programmer so the new segments can be defined to CP. You must wait for this to be completed before proceeding with the next step.

XA-based systems:

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 `SASCSEG6` command.

Execute the `DEFxSAS QUERY` command.

Setting default options

Enter the options

```
SSEG
SERIES=__
SASHELP=''
```

in your SYSPROF SAS file.

Sample CMSSEGS files

The following are three examples of completed CMSSEGS files, one for saving the SAS System on a non XA-based system, one for saving the SAS System on an XA-based system, and one for saving only user libraries.

CMSSEGS File for Non XA-Based System

For this example, assume that the following decisions have been made:

- The series letter is A.
- The segments will load at address 700000 hexadecimal (7M).
- The segments will be stored on 3380 DASD volume VM9999 at cylinder 450, page 1.

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 `SASA CMSSEGS: COPYFILE SAS CMSSEGS * SASA CMSSEGS A`. Then, `XEDIT SASA CMSSEGS A`, filling in the site parameters as shown below.

```
*  This file contains information needed to generate saved segments
*  in which to save SAS Version 6 code.
*
SAY Generating segments from SASA CMSSEGS control file
*
*--- general information
*
SERIES     A                ; which series to generate (1 character)
GENXA      NO               ; YES for XA based CP; NO for non-XA
DISK-TYPE  3380             ; disk type (i.e., 3380,...)       (non-XA)
DISK-LABEL VM9999           ; disk label for segments          (non-XA)
DISK-PAG   1                ; beginning page /CKD or FBA disks (non-XA)
DISK-CYL   450              ; beginning cyl  /CKD disks only   (non-XA)
*
*--- what address do we start loading things at
*
LOADADDR   700000
*
*--- the list of LLIST files to process
*
LLIST      SASSUPER         ; supervisor
LLIST      SASV6OP          ; startup-options
LLIST      SASAF            ; AF
LLIST      SASBPROC         ; base-PROC routines
LLIST      SASCALC          ; CALC
LLIST      SASDS            ; DATA STEP
LLIST      SASETS           ; ETS
LLIST      SASFSP           ; FSP
LLIST      SASGRAPH         ; GRAPH
LLIST      SASIML           ; IML
LLIST      SASOR            ; OR
LLIST      SASPRINT         ; PROC PRINT
LLIST      SASQC            ; QC
LLIST      SASSTAT          ; STAT PROCS (multiple segments)
```

CMSSEGS File for an XA-Based System

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 M.
- ❑ 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 ALLSEG CMSSEGS file from the SAS System disk to your A disk, and rename it `ALLSEGB CMSSEGS: COPYFILE ALLSEG CMSSEGS * ALLSEGB CMSSEGS A`.

Then, `XEDIT ALLSEGB CMSSEGS A`, filling in the site parameters as shown below.

```
*  This file contains information needed to generate saved segments
*  in which to save SAS Version 6 code. This file is specific to
*  loading all of SAS into segments above the 16 Meg line.
*  SASHELP and messages are member saved segments in the supervisor segment space

SAY    Defining XA segments from ALLSEGB CMSSEGS

*
*--- general information
*
SERIES     B
GENXA      YES               ; YES for XA based CP; NO for non-XA
*
*--- what address do we start loading things at
*
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                 ; Address of 0 specifies a member saved segment
MSGDB      MSGDB  M          ; Fileid of message database
*
*--- specifications for the help library segment
*
LIBLADDR   0                 ; Address of 0 specifies a member saved segment
LIBRARY  SASHELP
```

CMSSEGS File for Saving Only User Libraries

In this example,

- ❑ 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.

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, and does not require the GENXA statement. A CMSSEGS file that is only for loading user libraries (not including the SASHELP library) does not require a SERIES statement.

```
*  This file contains information needed to generate saved segments
*  in which to save SAS Version 6 libraries.
*
LIBLADDR 0C000000
LIBRARY  ABC * ABC XYZSPACE
LIBRARY  DEF * DEF XYZSPACE
*
LIBLADDR 0C400000
LIBRARY  DC01
LIBRARY  DC02
```

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

Reference

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

A. Types of saved segments

On non XA-based systems there is only one type of saved segment, DCSS (Discontiguous Saved Segments).

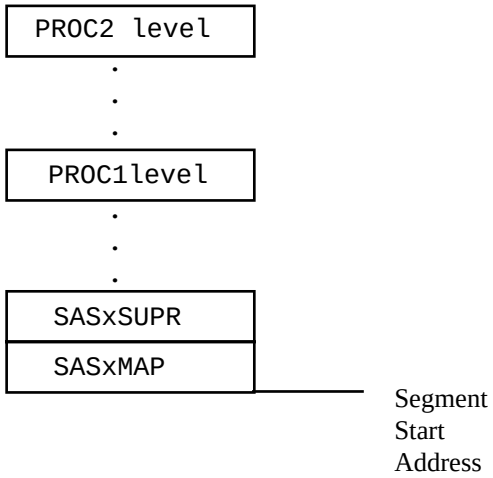
On XA-based systems, the SASxMAP1 and SASxSUPR segments are saved as member saved segments in the SASxMAP segment space. The PROC segments, if any, are saved in DCSS.

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.

B. Segment Layout

In virtual memory, SAS segments are laid out as follows:



where:

`SASxMAP1` is a binary search table of all the routines that are loaded into segments along with some other housekeeping information

`SASxSUPR` includes supervisor routines, common functions, and other common routines used by many other SAS procedures

PROCs include groups of SAS PROCedures generally collected in some logical grouping. In order to minimize the total virtual memory used, the PROCs are overlaid. There are multiple segments that load at the same address, or overlay each other.

Because some PROCs can call other PROCs, making it impossible for those PROCs to be at the same segment load address, there are two procedure levels.

When the ALLSAS LLIST file is used to load the entire SAS System, all PROCs are loaded into SASxSUPR. In this case the PROC1 and PROC2 levels are not used.

C. Tailoring What Goes Into Segments

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

You can omit items from segments using one of the following methods:

- ❑ Edit your CMSSEGS control file and comment out one or more of the LLIST statements to stop that entire LLIST file from being processed.

- ❑ Make a copy of one or more of the LLIST files 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.

D. 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 (for Version 6 Accounting data). You can specify the $SGSTAT option on the command line when invoking the SAS System, in a CONFIG SAS file, or in the SYSPROF SAS 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 types 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.

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 SMSG; furthermore 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 CONFIG SAS files or to the system SYSPROF SAS file.

E. 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 three prototype CMSSEGS files:

- ❑ SAS CMSSEGS for non XA-based systems
- ❑ SASXA CMSSEGS for XA-based systems; uses PROC1 and PROC2 overlays; loads a subset of the SAS System into saved segments

- ALLSEG CMSSEGS for XA-based systems; does not use overlays and loads the entire SAS System into saved segments

The contents of a CMSSEGS file are 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 files have "?" coded for most parameters, so you could use them 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.

Here 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) .

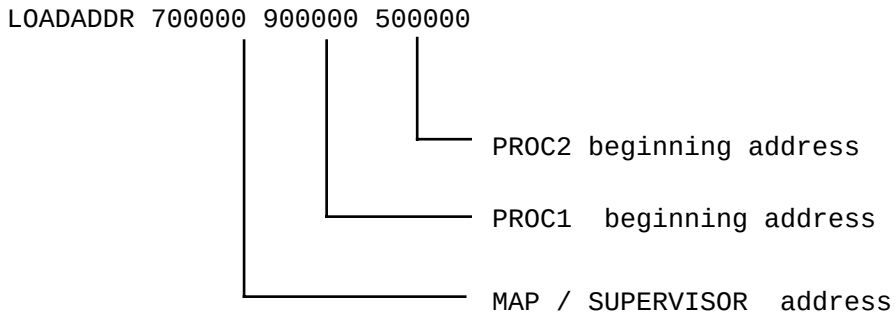
`GENXA` this must be YES if you are defining segments for an XA-based system, or NO if you are defining segments for a non XA-based system.

GENXA is not required if you are only saving SAS libraries or the message database; it defaults to YES in these cases.

`LOADADDR` specifies the address at which the SAS System segments will be loaded. You can specify up to 3 load addresses, which correspond to the beginning addresses for the SASxMAP1 segment and the PROC1 and PROC2 segments. Only the SASxMAP1 address is required; the PROC addresses default to the next available addresses. Specify the PROC addresses only if you need to have the segments located

elsewhere. The SASxSUPR address is not specified because it always immediately follows SASxMAP1.

For example, the following causes the SASxMAP1 segment to load at 7M, the PROC1 level at 9M, and the PROC2 level at 5M.



If you want to specify the SASxMAP1 and PROC2 addresses while allowing PROC1 to default to the next available address after SASxSUPR, specify 0 (zero) for the PROC1 address.

The following example specifies that the SASxMAP1 segment begins at 5M, the PROC1 level follows SASxSUPR, and the PROC2 level begins at 12M:

```
LOADADDR 500000 0 C00000
```

`DISK-LABEL` for non XA-based systems only, specifies the label of the DASD volume where the saved segments will be stored.

`DISK-TYPE` for non XA-based systems only, specifies the DASD type of the volume where the saved segments will be stored. Valid types are 3370, FBA, FBA-512, 3340, 2305, 2314, 2319, 3330, 3333, 3350, 3375, 3380, or 3390.

`DISK-CYL` for non XA-based systems only, specifies the starting cylinder where the segments will be stored on DISK-LABEL. Not used if DISK-TYPE specifies an FBA device.

`DISK-PAGE` for non XA-based systems only, specifies the starting page where the segments will be stored on DISK-LABEL.

`LLIST` specifies the name of a 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.

The remaining CMSSEGS statements are supported only on XA-based systems:

`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.

`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.

"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 DCSS's.

```
LIBRARY  filetype <filemode <segmentname <segmentspace>>>
```

The LIBRARY statement takes from 1 to 4 parameters. All but the first parameter is optional but any optional parameters must be specified in the sequence shown and none can be skipped.

filetype — is the filetype of a SAS library to be saved in a segment. Filetype is the only required parameter.

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.

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.

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, then:

- 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.
- 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.
- 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.

F. 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 a 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`

specifies at which overlay level this segment is to be loaded, as well as the name of the segment. The segment name can be a maximum of four characters. Level must be SASSUPR, SASPROC1, or SASPROC2 and should NOT be changed by the user.

`LABEL` descriptive text

specifies a description of the segment being created.

`SPACE` number

specifies the number of bytes to reserve at the end of the segment to allow for future expansion of the code (without having to re-gen the segment). SPACE is ignored when creating segments for an XA-based system because there is no gen process.

`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

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 the Institute. This licensing information in SAS software is called SETINIT information. SAS software will not run without valid SETINIT information.

When you first install the SAS System, the SETINIT information for your site is normally pre-applied, so you don't need to do anything to enable it. When you renew your license, license additional SAS software, have a change to your hardware, CPU model, or serial number, or if your SETINIT isn't working for some reason, you need to update your SETINIT information.

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"* for details on resaving segments.

Installing Additional SAS System Products

If additional products are licensed from SAS Institute after the initial installation of the SAS System, you will receive an add-on products tape.

After installing your new product, update your current SAS System license by running the SETINIT included on the new product tape. The SETINIT information is in a file called SETNAME SAS, which was loaded to disk from the tape. To update your SETINIT information, follow the steps in the previous section, *Running PROC SETINIT*.

For More Information

If you are licensing additional SAS software, see Part IV, "Installing Additional SAS System Products" 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 Standard 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 Standard 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 for SAS software products is distributed using the following two methods:

- ❑ For short-term emergency problems, some fixes are distributed on a SASNOTES tape in the form of zaps. However, the main purpose of SAS Notes is to provide information about known problems. Updated SASNOTES tapes are available every few months. See "Tailoring the SAS Notes" in Part III for more information.

- ❑ Periodically the problems corrected with zaps and other more complicated corrections are distributed on a MAINT tape in the form of replacement files (such as LOADLIBs, MODULEs, SAS catalogs, and MESSAGE files). This is by far the most common way of getting maintenance. Updated MAINT tapes are available less frequently than SAS Notes tapes. This type of maintenance is the focus of this chapter.

A MAINT tape contains all available maintenance for all products. When you install a MAINT tape, replacement files are loaded to disk for all products previously installed at your site. Maintenance files alone do not represent a full SAS System. Instead, MAINT tapes contain only those files that have had changes made to them since the production release of the product.

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

Note: The MAINT tape should be installed over a COPY of the existing Release 6.09 Enhanced of the SAS System. During the maintenance install process, files loaded from the tape will replace files on the specified minidisk 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.

The MAINTSAS EXEC will support loading a MAINT tape to disk. This process is slightly different than in previous releases. Refer to the Installation Instructions Addendum -- *The SAS System, Release 6.09 Enhanced (TS465), CMS* found in the installation package.

Part III, Tailoring the SAS® Notes

The SAS Notes section of a tape contains the Usage Notes, documentation notes, zaps, and the SAS programs needed to process the notes for your installation. SAS Usage Notes are a collection of reports on current SAS software problems, compatibility issues, and common misunderstandings.

A new master copy of the SAS Notes is created every few months and contains information on the current production release of SAS software. To obtain an updated copy, send a written request to the SAS Institute Distribution Center by sending email to distrib@vm.sas.com, or sending a letter to:

> SAS Institute Inc.
> Distribution Center
> SAS Campus Drive
> Cary, NC 27513

Fixes distributed as zaps are extremely limited. If you are interested in a more broad base set of fixes, see Part II of this chapter, "Applying Maintenance to SAS Software Products."

To load SAS Notes from tape to disk, complete the instructions in the following sections:

- Chapter 2, Part II, "Preinstallation Considerations"
- Chapter 2, Part III, "Loading the SAS System."

In addition, note that when running GETSAS to install only SAS Notes you should direct GETSAS to install `Some` (not `all`) of the tape, and answer `N` to all selections except for SASNOTES.

The files in the SAS Notes tape section are:

- A SAS Notes Library which consists of the SAS Usage Notes data sets and the screen and format catalogs necessary to access the Usage Notes. All of the files in this library have the filetype of USAGE6.

Instructions for Tailoring the Usage Notes

PARMS6 File

This file contains a list of parameters that must be modified in order to run the SELECT6, PRINT6, and ZAPREPT6 programs. Refer to the comment blocks in the `PARMS6 SAS` file for instructions on modifying the file. Be sure to read all the comment blocks before attempting to run any of the programs.

READPRM6 Program

This program is used by SELECT6, PRINT6, and ZAPREPT6 to read the `PARMS6` file and should not be modified.

SELECT6 Program

As distributed, the Version 6 Usage Notes data set (USAGE6.USAGE) contains notes for all SAS Institute products. You may not want notes that pertain to products that you do not have. The SELECT6 SAS file contains the SAS program that creates a subset of the Version 6 Usage Notes. Modify the parameters in the PARMS6 file to suit your needs before running the SELECT6 program. The SELECT6 SAS file should not be modified.

To invoke the SELECT6 program issue the following command:

```
SAS SELECT6
```

PRINT6 Program

The PRINT6 SAS file contains a program that prints a copy of the Usage Notes (USAGE6.USAGE). This program is available for your convenience and for sites that do not have SAS/FSP software for browsing notes in full-screen mode. Modify the parameters in the PARMS6 file to suit your needs before running the PRINT6 program. The PRINT6 SAS file should not be modified.

To execute PRINT6 SAS, issue the following command:

```
SAS PRINT6
```

ZAPREPT6 Program

The ZAPREPT6 SAS file produces a report that summarizes the available zaps by product and severity. Modify the parameters in the PARMS6 file to suit your needs before running the ZAPREPT6 program. The ZAPREPT6 SAS file should not be modified.

To execute ZAPREPT6 SAS, issue the following command:

```
SAS ZAPREPT6
```

Using the Support Application

A full-screen application has been created to allow you to browse the Usage Notes and compatibility issues, generate an index listing of sample programs available for various products, and edit and execute these sample programs. The SUPPORT exec has been provided to invoke this application.

Customizing the SUPPORT EXEC

An autoexec file that invokes the SUPPORT application is included. You should move this exec file to the appropriate location on your system for global access by users at your site. Some modifications to the exec may be required in order for the SUPPORT application to execute properly. Default names are supplied for the following:

- the command used to invoke the SAS System at your site
- the name of the autoexec file that includes the SAS statements used to invoke the application.

The autoexec file includes references to the SAS data library containing the SAS Notes database, the catalog containing the entries for the SUPPORT application, and the library containing the formats for the SAS Notes database (these two libraries may be the same). You may need to modify the libraries, based on naming conventions at your site.

Using SAS Notes

If you do not have SAS/FSP software, use the PRINT6 program to obtain a printed copy of the notes. If you have SAS/FSP software, you can browse the notes in full-screen form with PROC FSBROWSE. For example, to read the Usage Notes data set, enter the following PROC statement from a SAS session:

```
PROC FSBROWSE DATA=USAGE6.USAGE SCREEN=USAGE6.SCREEN; RUN;
```

To read the copyright data set, enter the following PROC statement:

```
PROC FSBROWSE DATA=USAGE6.COPYRGHT SCREEN=USAGE6.SCREEN; RUN;
```

When using the FSBROWSE and FSEDIT procedures, you need to issue a LIBNAME statement, `LIBNAME LIBRARY 'USAGE6 filemode'`. This gives access to the formats associated with the variables in the data set.

If your terminal does not support lowercase characters, use the following DATA step to translate fields to all uppercase before using PROC FSBROWSE.

```
DATA USAGE6.USAGE;
          SET USAGE6.USAGE;
          ARRAY TEXT TEXT1-TEXT20;
          DO OVER TEXT;
             TEXT=UPCASE(TEXT);
          END;
          TITLE=UPCASE(TITLE);
RUN;
```

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 SAS Notes section of the tape is a maclib called ZAP609 MACLIB. This MACLIB contains ZAPs that fix specific problems. The members included in this maclib have names that begin with Z609 and end with a 4-digit number that corresponds to a Usage Note number and the last four digits of MODULEN in the USAGE6.USAGE SAS data set. For example, `Z6090003`, is a zap for a problem referred to in Usage Note 3 for Release 6.09 Enhanced 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 has been provided on the tape to remove all members at once, or individual members as needed. See Appendix J, "MACLIBs on the Installation Tape" for more 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:

`libname#` is the filename of the loadlibs to be zapped

`filename` is the filename of the file that contains the zap to be applied (the filetype should be `ZAP`)

`NOPRINT` tells the `ZAP` command not to send information on the `ZAP` 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" 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. Your ADDON tape contains your new product, and current SAS Notes. To request an additional SAS System product, contact SAS Institute Customer Service at (919) 677-8003 between 9 a.m. and 8 p.m. Eastern Standard 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*. 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 in Chapter 2 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 Release 6.09 Enhanced SAS disks and install your add-on over the copy.

   Some SAS software products require other SAS products in order to run. For example, SAS/EIS software requires SAS/AF, SAS/FSP, and SAS/GRAPH software. If you attempt to install a product without the other required products, GETSAS displays an error message similar to the following:

   ```
   SAS/EIS requires SAS/AF to function.
   ```

   In this case, 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, see the section "Enabling the SAS Autocall Macro Libraries," in Chapter 2.

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

For more information about updating product licenses, see Chapter 4, Part I, "Updating Product Licenses Using SETINIT."

3. If you have a test program for the new product, run it to verify the installation. For more information about testing your installation, see the section, *Installation Verification Testing* in Chapter 2.

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."

5. Read the appropriate appendices for information about tailoring your new product. Note that not all products require post-installation modification.

Appendices

Appendix A, Tailoring SAS/ACCESS® Interface to SQL/DS Software

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

1. Link-edit step.

   SAS/ACCESS Interface to SQL/DS software uses the IBM SQL/DS resource manager stub routine, ARIRVSTC, to communicate with IBM SQL/DS. 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 SQL/DS is installed at your site.

   a. Access the IBM SQL/DS 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 SQL/DS data base machine. Refer to *IBM SQL/DS System Administration for IBM VM Systems* (GH09-8084-04) for additional information on the production minidisk for a data base machine.

   b. Using the `CMS LKED` command, link-edit the text file ARIRVSTC. 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 SYSPROF SAS file with the `SASLOAD=` option.

2. Execution step.

   The remaining steps of the installation process may require the cooperation of an IBM SQL/DS 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 SQL/DS. Refer to *IBM SQL/DS System Administration for IBM VM Systems* for additional information on defining IBM SQL/DS 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 SQL/DS Interactive SQL Guide and Reference for IBM VM Systems* (SH09-8085-03) for additional information on the production minidisk for a database machine.

c. You must create access packages by precompiling the SASACC, SASSQD, and SASSQDL source unloaded from the tape. The SQL preprocessor is invoked with the following commands:

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

where `fm` indicates the filemode of the SAS disk that contains SAS/ACCESS Interface to SQL/DS 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 SQL/DS data base administrator or refer to *IBM SQL/DS Application Programming for IBM VM Systems* (SH09-8086-03) for additional details on the IBM SQL/DS preprocessor.

d. UNLOAD the packages from a native SQL/DS 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 SASACC and is located on an SQL/DS system called SQLV3R4. It is to be loaded to a DRDA resource called DB2V2R3.

The following steps should be repeated for all three packages: SASACC, SASSQD, and SASSQDL.

1. Invoke SQLINIT to the SQL/DS system using the SQLDS PROTOCOL. This is necessary to perform the UNLOAD operation.

```
SQLINIT DB (SQLV3R4)   PROTOCOL (SQLDS)
```

2. Issue a filedef to describe the file to which the package will be unloaded.

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

3. Invoke the database utility SQLDBSU and issue the following command:

```
UNLOAD PACKAGE (SASACC) 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 (SQLV3R4) PROTOCOL (AUTO)
```

5. Invoke the database utility SQLDBSU again and issue the following commands:

```
CONNECT TO DB2V2R3;
RELOAD PACKAGE (SQLDBA.SASACC) REPLACE KEEP INFILE (ACC);
COMMIT WORK;
```

For more information concerning the SQLDBSU utility or DRDA, consult with your IBM SQL/DS data base administrator, or refer to the following documents:

IBM SQL/DS Database 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 SASACC TO userid
GRANT RUN ON SASSQD TO userid
GRANT RUN ON SASSQDL TO userid
```

or

```
GRANT RUN ON SASACC TO PUBLIC
GRANT RUN ON SASSQD TO PUBLIC
GRANT RUN ON SASSQDL 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 SQL/DS authority, you should ask your IBM SQL/DS data base administrator before attempting to issue the `GRANT` command.

Appendix B, SAS/ACCESS® Interface to SYSTEM 2000® Software

Perform the following installation steps only if you are licensed for the SAS Institute 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.

   Important 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

```
MAKEBUF
QUEUE      "data base1 filemode"
QUEUE      "data base2 filemode"
           ...
QUEUE      "data basen filemode"
QUEUE      ""
EXEC SASS2K
DROPBUF
EXIT
```

Multi-User Mode

```
MAKEBUF
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 C, Post-Installation Setup for SAS/ASSIST® Software

An exec called ASSIST EXEC is on the tape for 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.

Converting User Profiles From a Previous Release

There are many ways to convert your user profile. One is to simply note any changes you have made to your user profile, exit SAS/ASSIST software, delete the profile SASUSER.PROFILE.PASSIST.SLIST, and reinvoke SAS/ASSIST software. A profile with the new format will be created. You can now update your new profile with the option values you noted in your original profile.

Another way you can convert your user profile is to use a conversion program that is provided with this release. In the Program Editor window type and submit the following to convert to the new structure:

```
proc display cat=sashelp.qassist.profconv.scl; run;
```

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

The first section in this appendix describes the use of the sample script files shipped with the product. The remaining sections list supported software for access methods available on CMS and outline configuration procedures for those access methods that require additional configuration.

The access methods supported for CMS are TELNET, TCP/IP, and APPC. They are described in the order listed. Be sure to read the sections for the access methods that you use at your site for requirement information. Remote capability for the ASYNC, PC3270, PCLINK, and RSAS7171 access methods supported in previous releases is also available in this release. There is no change in the requirements for these access methods, and, therefore, there is no discussion of them in this appendix. Refer to *SAS/CONNECT Software: Usage and Reference, Version 6, Second Edition* for complete details on the access methods supported by other systems.

Storing and Locating SAS/CONNECT Script Files

Several sample script files are shipped with SAS/CONNECT software. SAS/CONNECT software uses these script files to establish a connection to a remote SAS session.

The new SAS System option, `SASSCRIPT`, points to the location of the SAS/CONNECT script files. The `SASSCRIPT` option is used by SAS/ASSIST and could be used by user-written SCL applications. The value for this option is a maclib name. When the system is installed, the script files are placed in RLINKSCR MACLIB. The following line appears in the SYSPROF SAS file:

```
* SASSCRIPT='RLINKSCR MACLIB'
```

Remove the asterisk to uncomment this option for use at your site. If you want to move the script files to another maclib, you must edit the SYSPROF SAS file and update the `SASSCRIPT` option with the new location.

System Configuration for the TELNET and TCP Access Methods

Note: The information in this section is required if you are using TELNET or TCP/IP.

Using the TELNET or TCP access method, a user on CMS can connect to any supported platform that is on the TCP/IP network, is running a SAS System release that has the corresponding access method support, and has SAS/CONNECT software licensed. With the TCP access method, one of the supported TCP/IP products must be installed on any node, local or remote, that you want to use with SAS/CONNECT software. For the TELNET access method, a supported TCP/IP package must be installed on the local node. The remote node does not have to run a supported TCP/IP product, but must run some TCP/IP product to make the node accessible using TELNET. Both the TCP/IP and TELNET access methods require line-mode TELNET on the remote host in order for a connection to be established through SAS/CONNECT software.

IBM's VM TCP/IP Release 2 or later is supported for VM/CMS in Release 6.09 Enhanced.

System Configuration for the APPC Access Method

Note: The information in this section is required if you are using APPC.

To use the APPC access method, you must have a release of VM that has APPC support:

- ❑ VM/SP Release 6
- ❑ VM/ESA.

This is all you need to communicate between userids on the same VM system.

You can also communicate with other systems that support APPC. For example:

- ❑ You can communicate between VM/ESA systems that are in either the same Transparent Services Access Facility (TSAF) collection, or in the same Communication Services (CS) collection.

- ❑ You can communicate with systems in an SNA network if you have installed Advanced Communication Facility for the Virtual Telecommunications Access Method (ACF/VTAM), Group Control System (GCS), and APPC/VM VTAM Support (AVS). To use this configuration, you will need to manage SNA session limits. For more information, see *Managing SNA Session Limits* later in this section.

For more information on available connections, refer to the IBM publications IBM VM/SP Release 6 Connectivity and Programming Guide and Reference (SC24-5377), or IBM VM/ESA Connectivity Planning, Administration, and Operation (SC24-5448).

Configuring CMS userids for APPC support

To set up a CMS userid to be the local host for SAS/CONNECT software, you need to set up a communications directory file. In addition, you may need to provide userid and password security information in an `APPCPASS` statement in the `CP` directory, or with the `APPCSEC` SAS System option.

To set up a CMS userid to be the remote host for SAS/CONNECT software, you need to set up a `$SERVER$ NAMES` file. In addition, you may need to edit the `PROFILE EXEC` file of any CMS userids used as a remote host. The following sections describe how to do this.

Communications Directory Files for CMS Userids Used as Local Hosts

To establish a SAS/CONNECT conversation from a CMS userid running as the local host, you must set up a communications directory file available on that userid. The file must contain an entry that will be used for the value of the SAS System option `REMOTE`. This communications directory file can reside at the system level, the user level, or both. By default, the system-level communications directory file is named `SCOMDIR NAMES`, and the default user communications directory file is `UCOMDIR NAMES`. After editing a communications directory file, you will need to issue the CMS command `SET COMDIR`. The format of an entry in a CMS communications directory is as follows:

```
:NICK.luname      :LUNAME.gateway targetlu
                  :TPN.SASRMT
                  :MODENAME.modename
                  :SECURITY.level
                  :USERID.user1
                  :PASSWORD.user1pw
```

where:

`NICK.luname`

specifies the eight-character symbolic destination name of the resource. When running SAS, specify this luname for the value of the SAS System option `REMOTE`.

Note: Due to restrictions on non-VM systems, a CMS user cannot take advantage of the aliasing support implicit in the CMS communication directory structure when accessing a non-VM system. Instead, the `NICK` value in the communications directory must be identical to the `targetlu` value in the `:LUNAME` definition.

`LUNAME.gateway.targetlu`

specifies a combination of two eight-character names. The first is a defined gateway name for connections outside of the TSAF

collection, and the second is the name of the target partner logical unit.

`TPN.SASRMT`

indicates the transaction program name as it is known to the target LU. For SAS/CONNECT software, this must always be `SASRMT`.

`MODENAME.modename`

specifies the mode name of the SNA session connecting the gateway to the target LU.

`SECURITY.level`

specifies the security type of the conversation. If the target LU is configured to validate security, set level to `PGM` and provide userid and password information. Otherwise, set level to `NONE`. See the following section, *Providing Userid and Password Security Information*, for more information.

`USERID.user1`

indicates the access security userid that is presented to the target logical unit for verification.

`PASSWORD.user1pw`

indicates the access security password presented to the target LU.

The following example enables a VM-to-VM connection. In this example, the `NICK` value (`CONNVM`) is different from the `targetlu value (N01SASPG)`.

```
:NICK.CONNVM       :LUNAME.N01SASOG N01SASPG
                   :TPN.SASRMT
                   :MODENAME.MSASIND
                   :SECURITY.PGM
                   :USERID.USER1
                   :PASSWORD.XXXX
```

The following example enables a VM-to-MVS connection. In this example, the `NICK` value (`N01TGT62`) is identical to the `targetlu value (N01TGT62)`.

```
:NICK.N01TGT62     :LUNAME.N01SASOG N01TGT62
                   :TPN.SASRMT
                   :MODENAME.MSASIND
                   :SECURITY.PGM
                   :USERID.USER2
                   :PASSWORD.XXXX
```

Providing Userid and Password Security Information for Local CMS Userids

The following alternatives are available for providing userid and password security information:

- Keep userid and password security information in an `SCOMDIR` or `UCOMDIR NAMES` file as shown in the examples in the previous section.

- Keep userid and password security information in the local user's `CP` directory in an `APPCPASS` statement. Refer to the IBM publication *VM/ESA Connectivity Planning, Administration and Operation* (SC24-5448) for more information on APPCPASS.

- Specify userid and password security information when running the SAS System by using the following SAS System option:

  ```
  APPCSEC=userid.password|password|_PROMPT_|_NONE_
  ```

  where:

  `userid.password`

  specifies both the userid and password.

  `password`

  specifies just the password and uses the local userid as the value for userid. It does not get userid from `UCOMDIR NAMES`, `SCOMDIR NAMES`, or an `APPCPASS CP` directory statement (this is different than `_NONE_`).

  `_PROMPT_`

  causes the SAS System to prompt the user for userid and password information when necessary. If the communications directory file entry contains `SECURITY.NONE`, no prompting is necessary.

  When prompted for a userid, if you press Enter without supplying one, then the local userid is used. It does not get userid from `UCOMDIR NAMES`, `SCOMDIR NAMES`, or an `APPCPASS CP` directory statement (this is different than `_NONE_`).

  When prompted for a password, the input field is non-display. If you press Enter without supplying a password, one is obtained from `UCOMDIR NAMES`, `SCOMDIR NAMES`, or an `APPCPASS CP` directory statement.

  `_NONE_`

causes the userid and password information to be obtained from `UCOMDIR NAMES`, `SCOMDIR NAMES`, or an `APPCPASS CP` directory statement. This is the default.

The APPCSEC option can be used as the primary method of providing userid and/or password information, or to override the userid and/or password information in UCOMDIR NAMES, SCOMDIR NAMES, or APPCPASS CP directory entries.

The maintenance and security features of these methods vary. For example:

UCOMDIR NAMES

- Maintained by user (including LUNAME, MODENAME, and so on).
- File needs to be made secure (for example, filemode 0, read password on the disk).

APPCPASS

- Maintained by system administrator for each userid.
- Very secure since accessing CP directories of other users requires privileged authority.

APPCSEC SAS System option

- Maintained by user. Interactive use requires typing in the information, leaving room for user error.
- If userid.password or password is specified noninteractively, then the file containing the option needs to be secure (for example, filemode 0, read password on the disk).
- When using _PROMPT_, the information does not show up in a SASLOG, nor in a console spool file.

It is possible to mix and match these alternatives. For example, you can keep the information in an APPCPASS CP directory entry and use the APPCSEC=_PROMPT_ SAS System option to override it when there are changes (at least until the system administrator can update APPCPASS). The handling of userid and password information for APPC at your site can involve both system administrators and users.

$SERVER$ NAMES Directory for CMS Userids Used as a Remote Host

To establish a SAS/CONNECT conversation to a remote CMS system, you must set up the $SERVER$ NAMES file in the remote CMS environment. This file defines the EXEC to be run when the connection is made to the remote CMS system. The format of a CMS $SERVER$ NAMES directory entry is as follows:

```
:NICK.SASRMT      :LIST.userid1...userid|.*
                  :MODULE.execname
```

where:

NICK.SASRMT

specifies the eight-character symbolic destination name of the resource. For SAS/CONNECT software, the program is always SASRMT.

LIST.userid1...useridn

enables you to limit the users that are allowed to connect to this system. Specifying LIST* allows all users to connect.

MODULE.execname

specifies the EXEC to be executed when a connection is made to the remote CMS system. This replaces the script file used to signon and signoff. The main function of this EXEC is to invoke the remote SAS session with the desired SAS options.

The following is a sample $SERVERS NAMES file:

```
:NICK.SASRMT      :LIST.*
                  :MODULE.RMTBOOT
```

The RMTBOOT EXEC specified in the above example might be structured as follows:

```
/*   This is the BOOTSTRAP EXEC */
/*   for the remote CMS host */
say 'Remote Bootstrap in Progress'
say 'Invoking the SAS System     '
'EXEC SAS (COMAMID=APPC DMR REMOTE=N01SASPG)'
queue 'CP logoff'
exit
```

Note: The SAS System options included in this example (COMAMID, DMR, and REMOTE) are required to invoke SAS/CONNECT software on the remote host.

Tailoring the PROFILE EXEC File for CMS Userids Used as a Remote Host

You must also ensure that the appropriate CMS SET commands are specified in the remote CMS virtual machine to allow the virtual machine to function as a remote host. The easiest way to ensure that these commands always get issued is to edit the PROFILE EXEC file on the remote CMS userid and add the following logic:

```
/*   Make sure that we are set up to accept */
/*   connections if we get autologged       */
if substr(diagrc(24,-1),11,1) = '2' then do
'SET SERVER ON'
'SET FULLSCREEN OFF'
'SET AUTOREAD OFF'
end
```

Since this logic is part of the `PROFILE EXEC`, these CMS commands will get issued at LOGON if the CMS userid is being autologged through a SAS/CONNECT signon.

Defining VTAM gateways for APPC support

In order to establish a SAS/CONNECT conversation from a VM/CMS system, (VM/CMS is the local session), local-domain VTAM application minor node identifiers must be defined. This is accomplished (by VTAM systems personnel) through the specification of `APPL` definition statements. A global gateway must be defined to provide SAS/CONNECT software users on VM/CMS with a pathway for reaching the desired remote environment.

The following is an example of a definition for a VTAM gateway for APPC support:

```
N01SASOG APPL  ACBNAME=N01SASOG,

      APPC=YES,
      AUTH=(ACQ),
      AUTHEXIT=YES,
      AUTOSES=0,
      DLOGMOD=modeent,
      DMINWNL=1,
      DMINWNR=0,
      DSESLIM=1,
      EAS=30,
      MODETAB=modetab,
      PARSESS=YES,
      SECACPT=CONV,
      SONSCIP=YES,
      VPACING=n
```

In order to establish a SAS/CONNECT conversation to a VM/CMS system (VM/CMS is the remote session), a non-dedicated private gateway must be defined to provide SAS/CONNECT software users with a pathway for reaching the VM/CMS system.

For example:

```
N01SASPG APPL  ACBNAME=N01SASPG,

      APPC=YES,
      AUTH=(ACQ),
      AUTHEXIT=YES,
      AUTOSES=0,
      DLOGMOD=modeent,
      DMINWNL=0,
      DMINWNR=1,
      DSESLIM=1,
      EAS=30,
      MODETAB=modetab,
      PARSESS=YES,
      SECACPT=CONV,
      SONSCIP=YES,
      VPACING=n
```

A description of the gateway definition is as follows:

ACBNAME	defines the minor node name assigned to this application program.
APPC	allows the application to exploit the APPCCMD interface.
AUTH	enables the application to acquire a session with a particular logical unit.
AUTHEXIT	allows the application's exit routines to run in supervisor state.
AUTOSES	defines the number of contention-winner sessions to activate automatically.
DLOGMOD	defines the default session parameter mode table entry.
DMINWNL	reserves n local contention winner sessions.
DMINWNR	reserves n remote contention winner sessions.
DSESLIM	defines maximum session limits.
EAS	defines concurrent session expectations, based on expected concurrent user count.
MODETAB	defines the session parameter mode table.
PARSESS	allows multiple concurrent sessions with another application program.
SECACPT	enables the acceptance of FMH5 security subfield information.
SONSCIP	allows the application to receive UNBIND RUs in its SCIP exit routine.
VPACING	sets per-site network requirements.

Session mode entries (or modenames) defining protocols and performance parameters are required to support session binding to a secondary logical unit residing within the local VTAM domain. A sample `MODEENT` description for a VM/CMS system is provided below. The reader is referred to the `MODEENT` description in *VTAM Resource Definition Reference* (SC31-6412) for a more detailed description.

```
N01MOD1  MODEENT LOGMODE=N01MOD1,
                 FMPROF=X'13',
                 TSPROF=X'07',
                 PRIPROT=X'B0',
                 SECPROT=X'B0',
                 COMPROT=X'50B1',
                 RUSIZES=X'xxxx',
                 PSERVIC=X'060200000000000000102F00',
                 TYPE=0
```

Managing SNA Session Limits

To use SNA for your APPC communications, the SNA session limits between the AVS outbound gateway and the appropriate remote system logical unit partner must be raised above zero. The VM command, `Change Number of Sessions` (`CNOS`) manages those limits. IBM's APPC/VM does not support a programmatic interface to this command, so the SAS System cannot issue it automatically. Instead, the operator must enter control information at an AVS console, or systems personnel must provide some other way of entering that information.*

Note: `CNOS` is a privileged command.

Once the `CNOS` command has been issued, it remains in effect for as long as the remote system is operative. This spans the life of multiple SAS/CONNECT sessions going to the specified remote LU. However the command must be reissued any time the remote system logical unit experiences an outage and has to be restarted (including any time a remote OS/2 host is rebooted). If the command is not reissued, communication between the two partner logical units cannot proceed.

It is recommended that the `CNOS` command be routinely issued every time the remote system is restarted. This enables VM users to gain access to the remote system at any time without operator intervention.

Since `CNOS` is a VM command, the following information is for example only. The format of the `AGW CNOS` command is as follows:

```
AGW CNOS gateway remotelu  modename seslimit conwin conlose
```

* One way to allow CNOS commands to be entered is through the use of the Programmable Operator Facility (PROP) that is shipped as part of the VM system.

The parameters to the `CNOS` command can be described as follows:

`gateway` specifies the name of the local LU that is the gateway.

`remotelu` specifies the name of the remote LU for which the session limits are set.

`modename` specifies the logon mode name for which the session limit and contention values are changed.

`seslimit` specifies the maximum number of LU-to-LU sessions allowed between the gateway LU *`gateway`* and the remote LU *`remotelu`* for the logon mode name *`modename.`*

`conwin` specifies the number of parallel sessions for which the gateway LU *`gateway`* is guaranteed to be the contention winner.

`conlose` specifies the number of parallel sessions to which the remote LU *`remotelu`* is guaranteed to be the contention winner.

Example `AWG CNOS` command:

```
AWG CNOS N01SASOG N01TGT62 NO1MOD1 100 50 50
```

The VTAM systems personnel at your site will be able to provide the specific values needed for the `CNOS` command. The exact settings for your site will depend on the number of simultaneous users of the gateway. A general rule of thumb for the `CNOS seslimit` value is to allocate three sessions per userid that will simultaneously use the gateway.

References

Establishing communications within an SNA network, especially for a host subarea peripheral node, can be extremely trying until the configuration is properly specified. The APPC communications access method reflects failures as they are reported by the operating system components with which it interfaces. Often this information will be in the form of operation codes, return codes, and sense data. No attempt is made to interpret these failures. Rather, given the numerous possibilities for failure that are a function of your particular environment, it is expected that you will work with network systems and SAS support personnel at your site to attain resolution. You may refer to the following IBM publications for additional information:

- *VM/SP Release 6 Connectivity and Programming Guide and Reference* (SC24-5377)
- *VM/ESA Connectivity Planning, Administration, and Operation* (SC24-5448)
- *SNA Technical Overview* (GC30-3073)
- *SNA Formats* (GA27-3136)
- *VTAM Programming for LU6.2* (SC30-3400)

- ❑ *Extended Services for OS/2 Communications Manager User's Guide* (S04G-1015)
- ❑ *Extended Services for OS/2 Communications Manager Configuration Guide* (S04G-1002)
- ❑ *Extended Services for OS/2 Problem Determination Guide for the Service Coordinator* (S04G-1006)
- ❑ *Extended Services for OS/2 Programming Services and Advanced Problem Determination for Communications* (S04G-1007)
- ❑ *Extended Services for OS/2 Communications Manager System Management Programming Reference* (S04G-1116)
- ❑ *Extended Services for OS/2 APPC Programming Reference* (S04G-1025).

In the event that you cannot resolve your connectivity problem, call SAS Technical Support for assistance. You will likely be requested to generate traces to assist in documenting the problem, so ensure that you or your site support personnel are familiar with the tracing services available through VTAM and OS/2 Extended Services.

Appendix E, Tailoring 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:

Device Help Screens and Managing Device Catalogs

Using SAS/GRAPH with ASCII Devices

Creating a Linkable Device Driver

Using SAS/GRAPH with IBM 3270-type Terminals

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

Using SAS/GRAPH with GDDM

Device Help Screens and Managing Device Catalogs

Device HELP Screens

These instructions contain information on setting up system parameters required to use certain drivers, and instructions on how to build drivers where necessary. Refer to *SAS/GRAPH Software: Using Graphics Devices in the CMS Environment, Version 6, First Edition* for additional information on configuring and using graphics devices.

If you are using the SAS System interactively on a full-screen terminal, you can also obtain details on using graphics devices by entering HELP on any Display Manager command line. From the main `HELP` menu, select `GRAPHICS`, then select `GRAPHICS DEVICES AND DRIVERS` from the `GRAPHICS HELP` menu. These HELP screens contain information on configuring specific devices and `GOPTIONS` required to send output to them.

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 Chapter 25, "The GDEVICE Procedure", in *SAS/GRAPH Software: Reference, Version 6, First Edition,* and pages 36-52 in SAS Technical Report P-215, *SAS/GRAPH Software: Changes and Enhancements, Release 6.07.*

How Device Catalogs are Used

In Version 6 of the 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.

An Institute-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 Institute-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 FILENAME 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
o 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 Institute-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

Note: You need to complete this step only if you are using ASCII terminals or attached plotters interactively on ASCII lines. If all of your ASCII devices are attached using 3270-type or 3287-type protocol converters, you can skip this section.

In order for SAS/GRAPH software to communicate with ASCII devices, graphics commands must be translated from ASCII to EBCDIC, and back to ASCII again. This requires both system and SAS/GRAPH translate tables. `SASGTAB0` is a translate table in the catalog `SASHELP.HOST` that is designed for use with CMS. In most cases, you should be able to use this table to produce correct translation. However, if `SASGTAB0` is not able to perform all of the translation correctly, it may be necessary to create your own SAS translate table. Refer to SAS Technical Report P-197, *The TRANTAB Procedure,* for details on creating your own translate table.

In Version 5 of the SAS/GRAPH software, SAS/GRAPH translate tables were stored as LOADLIB modules. In Version 6, the Institute-supplied tables are stored as members of the catalog SASHELP.HOST. If you were using the Institute-supplied table (GTABCMS) in Version 5, then you can use the SASGTAB0 table in Version 6 and no special steps need be taken. (The SASGTAB0 table that is the default in Version 6 replaces the GTABCMS table used in Version 5.) If you created your own table in Release 6.06, 6.07, or 6.08, you can copy the table from the SASHELP.HOST catalog in that release to your 6.09 Enhanced SASHELP.HOST catalog. If you created your own translate table in Version 5, then you must convert the Version 5 module to a Version 6 catalog entry. If you need to convert your translate table, contact SAS Technical Support for details.

The TRANTAB= graphics option and the TRANTAB device parameter select a translate table for your system. The TRANTAB value can be specified either in a device entry or in a GOPTIONS statement. You can set a new default in either of two ways. One method is to modify device catalog entries for all ASCII terminals and interactive plotters that you will be using at your site. To do this, use the following SAS statements:

```
PROC GDEVICE C=SASHELP.DEVICES NOFS;
MODIFY entryname
TRANTAB=tablename;
QUIT;
```

Replace entryname with the name of the entry you want to modify, and tablename with the new TRANTAB value.

A second method (which may be more efficient if you have many types of terminals and plotters at your site) is to create a file containing the following statement:

```
GOPTIONS TRANTAB=tablename;
```

You can then modify your SAS EXEC to include a filedef for the fileref SASEXEC which points to this file. Alternatively, you can point to this autoexec file using the AUTOEXEC= SAS System option at invocation. Note, however, that any TRANTAB value that you supply in an AUTOEXEC file overrides a value specified in a device entry.

Using SAS/GRAPH with ASCII Terminals or ASCII Terminal Emulators

ASCII terminals and PCs running ASCII terminal emulation software are most commonly attached to CMS systems through a 3270-type protocol converter. When this is the case, it is necessary to provide both a device driver and a GPROTOCOL value on your GOPTIONS statement. For terminals and PCs running emulation software, the DEVICE= parameter on the GOPTIONS statement will reflect the type of graphics terminal your emulation software emulates.

The following is a list of ASCII terminal emulation products that should work with SAS/GRAPH software when attached to 3270-type protocol converters. However, SAS Institute assumes no liability regarding the functionality of any particular emulation product. For details on using a particular emulator with SAS/GRAPH software, contact SAS Technical Support.

Emulation product	Device Driver
Reflection 4	VT340
Reflection 7	HP2397
ZSTEM340	VT340
ZSTEM240	VT240
Smarterm340	VT340
Smarterm240	VT240
Kermit Version 2.3 and higher	TEK4014
TGRAF 4200 series	TEK4205
Emutek 4200 Plus	TEK4205
EM4105	TEK4105

Protocol converters can be used on IBM systems to allow asynchronous devices to communicate with the host computer on bisynchronous or SDLC lines. Protocol converters that perform IBM 3270 emulation make asynchronous terminals appear to the host as remote IBM 3278 or 3279 terminals. Some converters in this category appear to the host as 3274 cluster controllers and permit the attachment of both asynchronous terminals (which appear to be 3278/79 terminals) and hard copy devices (which appear to the host to be 3287 printers). When these converters are used, SAS/GRAPH output is supported to both asynchronous graphics terminals and interactive plotters.

Since these converters are used with Institute-supplied interactive drivers and I/O is performed by SAS/GRAPH software, only the following 3270-type protocol converters are supported by SAS/GRAPH software:

- ❑ PCI 1071 and 1076 (and compatible models)
- ❑ KMW 3270-FS
- ❑ DATASTREAM
- ❑ RENEX
- ❑ Commtex C-80
- ❑ Local Data DATALYNX
- ❑ Tektronix 4970
- ❑ IBM 7171
- ❑ Hydra

Because new converter models are continually being introduced by many of the above manufacturers, you should consult the vendor to determine which specific models and configurations will work with SAS/GRAPH software. We strongly recommend that before purchasing a converter, you make arrangements to obtain it on a trial basis to confirm that it will work.

To use SAS/GRAPH software with these converters, asynchronous terminals should be attached directly to the converter. If you are using an interactive plotter, it should be in eavesdrop mode between an asynchronous terminal and the protocol converter.

These converters are designed for interactive use. If you are using an ASCII hard copy device in batch mode with a protocol converter (such as an AGILE 6287 or Local Data INTERLYNX), see the next section.

To obtain graphics output with the interactive converters listed above, invoke SAS/GRAPH software under CMS. If you are using a Local Data DATALYNX, specify:

```
GOPTIONS GPROTOCOL=SASGPSTE;
```

If you are using an IBM 7171 converter, specify the following:

```
GOPTIONS GPROTOCOL=SASGP71;
```

If you are using a HYDRA protocol converter, specify the following:

```
GOPTIONS GPROTOCOL=SASGPHYD;
```

For any of the other supported models, specify the following:

```
GOPTIONS GPROTOCOL=SASGPSTD;
```

You should also specify the name of the graphics device you are using in an `OPTIONS` or `GOPTIONS` statement. Then proceed to use SAS/GRAPH as you normally would with an interactive device.

Note to Version 5 users: `GPROTOCOL` values have changed between Version 5 and Version 6. However, you can still use the Version 5 equivalents `(GSASPSTE, GSAS7171,` and `GSASPSTD)` in Version 6. Refer to table 5.2 in *SAS/GRAPH Software: Reference, Version 6, First Edition* for a complete listing of `GPROTOCOL` values.

Using SAS/GRAPH with ASCII Printers, Plotters, or Cameras

When using ASCII printers or plotters on a CMS system, the device must be connected to a 3287-type converter, which is normally attached to a 3174 or 3274 control unit and appears to the host as an IBM 3287 printer. You can attach various types of ASCII hard copy devices to these converters and spool output to them through RSCS.

To create output that can be sent to the device, specify the following `GOPTIONS:`

```
GOPTIONS DEVICE     = drivername
         GACCESS    = GSASFILE
         GPROTOCOL  = protocol-value
         GSFMODE    = REPLACE
         HANDSHAKE  = XONXOFF
         GSFLEN     = 64;
```

These options write output to the fileref `GSASFILE` that can point to a file or directly to the device. Refer to *SAS/GRAPH Software: Using Graphics Devices in the CMS Environment* for additional details on sending output to your device.

To fully automate this process for the end user, the parameters in the above GOPTIONS statement can be incorporated into an Institute-supplied device driver entry. Refer to the section, *Device Help Screens and Managing Device Catalogs* for details on using PROC GDEVICE to customize a device driver.

When using SAS/GRAPH software with a 3287-type converter, use the driver for the actual graphics device you are using (a QMS printer or Hewlett-Packard plotter, for example). For specific GOPTIONS settings, refer to the SAS System online HELP facility or to *SAS/GRAPH Software: Using Graphics Devices in the CMS Environment.* You must also specify a value for the GPROTOCOL option. If the converter can send all characters in the data stream to the device, or if the data stream contains only printable characters, specify GPROTOCOL=''. This value can also be used for most devices with a direct coaxial attachment, such as the Zeta 887 plotter or Xerox printers with coax interface.

If the data stream for the device contains unprintable characters that the converter cannot process, you must specify an appropriate value for the GPROTOCOL option so that the data stream is modified to a format that passes through the converter. GPROTOCOL values for selected converters are presented in the following table.

GPROTOCOL value	Protocol Converter
SASGPAGL	Agile 5287 and 6287
SASGPLCL	Local Data InterLynx, KMW 3287, Andrew (Newport, Malibu, Laguna)
SASGPISI	ISI 87
SASGPVAT	Avatar
SASGPIDX	IDEX
SASGPNET	NetCommander
SASGPDCA	IrmaPrint, IrmaPrint II, TEK 4512, QMS AFPlink
SASGPIDA	I-Data 3274
SASGPDAT	(EG) VII-Data

Note to Version 5 users: GPROTOCOL values have changed between Version 5 and Version 6. However, you can still use the Version 5 equivalents (GSASAGIL, GSASLDIL, and GSASISI) in Version 6.

If your converter is not listed above, but can process your device's data stream without modification, GPROTOCOL='' may work.

You can send output to the device through RSCS. To spool directly to RSCS, specify the following commands from within the SAS System:

```
CP SPOOL PRINT TO RSCS;
CP TAG DEV PRINT dest;
FILENAME GSASFILE PRINT;
```

The dest parameter on the CP tag command is the RSCS destination for your device.

You can also direct your output to a file and later use the PRINT command to send your output to the device using RSCS. Simply change the FILENAME for GSASFILE so that it points to a file rather than to RSCS PRINT.

Installing the Linkable Driver

This section describes the steps necessary to install the LINKABLE driver. The LINKABLE driver makes calls to user-supplied "CalComp compatible" plotting subroutines. These subroutines are normally provided by the plotter vendor when the plotter is purchased. Other plotting subroutine libraries may be available from third party vendors. The subroutine library should contain routines named `PLOT`, `PLOTS`, `SYMBOL`, and `NEWPEN` (or `TONE`). If your site does not have a device that uses vendor-supplied subroutines (and does have a ready-to-use SAS/GRAPH driver), skip this part and refer to the instructions for the device(s) you are using.

The LINKABLE driver must be linked with your plotting subroutines before it can be used. The output from this driver may be directed to tape for an off-line device, or it may write to RSCS and be directed to an online device (that appears to the system as a punch or a 3287 printer). It can also be written to disk for later spooling to the device.

The installation of the LINKABLE driver involves two steps. In the first step, an executable driver module is created. In the second step a device catalog entry, containing actual characteristics of the device, is created.

Notes: If you are also using Version 5 SAS/GRAPH software, the LINKABLE driver has been redesigned for Version 6 and uses the Metagraphics driver facility. The installation process for the driver has changed somewhat, but to the end user, use of the driver is nearly the same as the Version 5 driver. See "Note to Sites Using Version 5 SAS/GRAPH Software" for an overview of changes.

If you built a driver using the linkable driver in Release 6.06, you must rebuild the driver for Release 6.09 Enhanced. If you built a driver in Release 6.07 or 6.08, you can copy your old device entry to your 6.09 Enhanced device catalog and use your old module.

This section is divided into the following parts:

- Overview of the LINKABLE device driver

  This section describes how the LINKABLE driver operates and provides a description of the calls made to the plotting routines by the driver.

- Generating a driver module

  This section provides instructions for compiling an Institute-supplied FORTRAN program and including a library of plotting routines to create a driver module.

- Customizing device characteristics

  The characteristics of the device (such as size of plotting area, default colors, rows and columns) are stored in a device catalog entry. This section shows how to edit the device catalog entry to supply the specific characteristics of your device.

- Testing the driver

  This section illustrates a sample program that you can use to make sure that the driver and device are working correctly.

- Special information for Versatec RANDOM users

  If you are creating a driver for a Versatec monochrome or color plotter, refer to this section before attempting to build a driver. This section contains special instructions for creating a driver that calls the Versatec TONE routine (instead of `NEWPEN`), and invokes Versatec RANDOM routines.

- Special Information for Xerox XPPI Version 1.0 Users

  If you are creating a driver for a Xerox printer, refer to this section before attempting to build a driver. This section contains special instructions for creating a driver that calls `XPPI` routines.

- Common problems and solutions

  This section describes some of the most common problems encountered when trying to install and/or use the LINKABLE driver. If you have any problems, please consult this section before calling SAS Institute.

Overview of the LINKABLE Device Driver

The LINKABLE device driver is a special driver supplied with SAS/GRAPH software that is incomplete until linkedited with user-supplied plotting routines. These routines are normally provided by the manufacturer when the plotter is purchased or obtained from a third party vendor. Since these plotting routines perform all the driver output, this driver does not depend on ASCII translate tables described in the documentation for ASCII devices.

How the driver works

When built, the driver consists of two parts: a device catalog entry that contains the characteristics of the device, and an executable module. The executable module is created by compiling an Institute-supplied FORTRAN program and linking it with the plotting subroutines that are supplied with your device.

The LINKABLE driver uses the SAS Metagraphics driver facility to produce its output. When a SAS/GRAPH software procedure (that produces a graph) is executed, the following occurs:

1. An "internal" (Institute-supplied) driver module, using information on device characteristics from a device catalog entry, produces a metafile of graphics data.

2. An "external driver" (the executable module that you create by compiling the `LINKAIBM FORTRAN` program and linking it with your vendor-supplied plotting routines) reads the metafile and makes calls to the plotting subroutines. The plotting subroutines actually produce the output.

Plotting routines used by the driver

The LINKAIBM FORTRAN program makes calls to four subroutines (PLOTS, PLOT, SYMBOL, and NEWPEN) that must follow the CalComp standard format as documented below.

```
PLOTS(0,0,14)
```

The PLOTS subroutine is normally used to initialize the device. The third parameter, 14, causes the driver to write output to the FORTRAN unit 14 (FT14F001). To change the value of the first two parameters, you can modify the values of the PLOTS1 and PLOTS2 variables in the LINKAIBM FORTRAN program. To modify the FORTRAN output unit used by the driver, change the value of the variable OUT in the INIDVR subroutine of the LINKAIBM FORTRAN program.

```
PLOT(X,Y,Ipen)
```

Ipen codes the pen up or down as well as coding origin reset and plot termination.

Ipen=-3	move and reset origin
Ipen=2	draw
Ipen=3	move
Ipen=999	move and terminate plot

X and Y are the pen coordinates in inches.

NEWPEN(Ipen#)

The NEWPEN routine is used to select the current pen or color. By default, the LINKABLE device driver assumes there are four pens named BLACK, RED, GREEN, and BLUE associated with Ipen numbers 1, 2, 3, and 4 respectively. To change the number of pens or the default colors, refer to the section *Customizing Device Characteristics*.

SYMBOL(X,Y,Height,String,Angle,Nchar)

Height=	height in inches of characters
String=	string of EBCDIC characters
Angle=	angle in degrees (0.0 or -90.0)
Nchar=	number of characters in string

X and Y are the pen coordinates in inches.

Subroutines written in FORTRAN 77 have a SYMBOL routine that takes seven parameters, as follows:

```
SYMBOL(X,Y,Height,String,INTEQ,Angle,Nchar)
```

where INTEQ is a dummy argument. If your SYMBOL routine expects seven parameters, you will need to make modifications to the LINKABLE program, as described in the next section, *Generating a Driver Module*.

The relevant routines in some plotter libraries may not be named as above, but have the correct arguments. In other cases, the arguments may not agree in type, number, or meaning. You can write interface routines to transform the arguments into what the plotter subroutines expect. In addition to calls to the above routines, calls may be made to a series of additional routines to customize the driver to suit the requirements of your site or nonstandard plotting libraries. A description of these additional routines follows.

Additional routines

In addition to calls to the PLOTS, PLOT, SYMBOL and NEWPEN routines, a number of additional routines are called by the driver. These may be used for pre-initialization, post-termination, and so forth. These routines are:

```
PLOTA1: called before PLOTS(0,0,Iunit) - Beginning of each PROC
PLOTA2: called after PLOTS(0,0,Iunit)  - Beginning of each PROC
PLOTC1: called before PLOT(x,0.0,-3)   - After CALL PLOT(-x,.5,-3)
PLOTC2: called after PLOT(x,0.0,-3)    - After CALL PLOT(-x,.5,-3)
PLOTF1: called before PLOT(x,-.5,-3)   - End of each plot
PLOTF2: called after PLOT(x,-.5,-3)    - End of each plot
PLOTG1: called before PLOT(-x,.5,-3)   - Beginning of each plot
PLOTG2: called after PLOT(-x,.5,-3)    - Beginning of each plot
PLOTZ1: called before PLOT(0.,0.,999)  - End of each PROC
PLOTZ2: called after PLOT(0.,0.,999)   - End of each PROC
```

Empty versions of the above subroutines are included in the LINKEXT program. You can replace the empty subroutines with your own versions by modifying the LINKEXT FORTRAN source.

To avoid the minimal overhead of these empty function calls, you can modify the LINKAIBM source program by commenting out the CALL statements to these routines before compiling the driver.

Generating a Driver Module

To generate an executable driver file, you must compile the LINKAIBM FORTRAN program (which can be found in GSOURCE MACLIB) and use the CMS INCLUDE and GENMOD commands to include vendor-supplied plotting routines and create an executable module. You can rename the LINKAIBM FORTRAN program.

These instructions assume that your plotting subroutines are stored in a txtlib. If your routines are stored in a loadlib, contact SAS Institute Technical Support for details on linking the LINKAIBM object with your plotting routines.

The LINKAIBM FORTRAN program passes six parameters when it makes calls to the SYMBOL routine. If your plotting library contains a SYMBOL routine that uses seven parameters, you must modify the LINKAIBM program so that it passes seven parameters. To do this, edit the program and find the CALL SYMBOL statement (which passes six parameters). Following this statement is a commented statement containing a CALL SYMBOL with seven parameters. Remove the C in column 1 of the commented statement, and add a C to column 1 of the original CALL SYMBOL statement.

If the SYMBOL routine in your plotting library only uses six parameters, you should not have to make any modifications to the LINKAIBM program.

Note: You should compile the program with the same compiler that you use for your plotting routines. If you do not, errors may occur when using the driver.

To complete installation of the LINKABLE driver, the driver must be compiled and linked with your CalComp-compatible routines following the procedure described below.

1. Copy the LINKAIBM FORTRAN program from GSOURCE.MACLIB to your A disk. Rename the file to mydriver FORTRAN. mydriver should match the name of the executable specified in the PROCESS option of your device catalog entry (see the section, *Customizing Device Characteristics*).

   If the additional routines described in the previous section will not be used by your driver, comment out the CALL statements to these routines by placing a C in column one of each of these statements. If the additional routines will be used, the LINKEXT FORTRAN program can be modified to add the desired functions and concatenated to the bottom of the LINKAIBM FORTRAN program.

2. If your plotting subroutines are written in FORTRAN 66, compile the FORTRAN program using the following command:

   ```
   FORTVS mydriver (LANGLVL(66)
   ```

   If the LINKAIBM FORTRAN program is written using FORTRAN 66, and you compile the program without the LANGLVL(66) option, you will receive warning messages similar to the following:

   ```
   AN AMPERSAND IS AN INVALID CHARACTER FOR "LANGLVL 77."  THE
   AMPERSAND WILL BE PROCESSED AS AN ASTERISK.  SPECIFY A
   VALID CHARACTER TO AVOID AN ERROR MESSAGE.
   ```

   The program compiles correctly. If you want to avoid the message, edit the FORTRAN program and change all occurrences of an ampersand to an asterisk.

   If your plotting subroutines are written in FORTRAN 77, compile the FORTRAN program using the following command:

   ```
   FORTVS mydriver
   ```

3. Make your FORTRAN and plotting routine txtlibs available by using the following command:

   ```
   GLOBAL TXTLIB yourfortranlibrary yourplotterlibrary
   ```

4. Create a temporary copy of the executable program in virtual storage using the LOAD command. This loads the plotting subroutines that replace the equivalent subroutines in the LINKAIBM program.

```
LOAD NEWPEN PLOT PLOTS SYMBOL (RLD CLEAR
```

If you do not make the FORTRAN txtlibs `global`, you will receive the following message:

```
DMSLIO210W THE FOLLOWING NAMES ARE UNDEFINED: IFYVIOFM VFFXF#
VFIXF# etc.
```

5. Load the `LINKAIBM mydriver` program into virtual storage:

```
INCLUDE mydriver (SAME)
```

You may receive "duplicate identifier" messages when executing the `LOAD` command. These can be ignored.

6. Create an executable program that is stored as a relocatable module on your CMS disk:

```
GENMOD mydriver
```

Verify that the executable file, `mydriver MODULE A`, was successfully created.

Note: The LINKABLE driver primarily calls plotting routines (`NEWPEN`, `PLOT`, `PLOTS`, and `SYMBOL`). These routines may in turn call other routines in your plotting library.

Customizing Device Characteristics

In Version 6 of SAS/GRAPH software, the device characteristics (such as graph size, default colors, rows, and columns) used by each driver are stored in an entry in a device catalog. When you specify a device driver name in your SAS/GRAPH program (in the `DEVICE=` option or when prompted), the name refers to an entry in a device catalog. For example, if you specify `DEVICE=MYDRIVER`, the software locates an entry named `MYDRIVER` in available device catalogs, and uses the information in that entry to determine such parameters as graph size, orientation, resolution, and so on.

The default device catalog, `SASHELP.DEVICES`, is supplied as part of SAS/GRAPH software. In the `SASHELP.DEVICES` catalog, an entry named `LINKABLE` contains the default characteristics used by the LINKABLE driver. For many devices, you can use the default characteristics and create the driver entry for your device by simply copying the `LINKABLE` entry. For devices that do not use the default characteristics, you can create a new entry and modify the default parameters.

To get a list of the default characteristics used by the LINKABLE driver, you can use the GDEVICE procedure to list the contents of the LINKABLE entry in SASHELP.DEVICES. To do this, submit the following SAS statements:

```
PROC GDEVICE NOFS C=SASHELP.DEVICES;
             LIST LINKABLE;
             END;
```

Important parameters from the LINKABLE entry are below. Refer to *SAS/GRAPH Software: Reference, Version 6, First Edition* for additional details on the parameters.

```
Lrows:          0      Hsize:       7.500 in
Lcols:          0      Vsize:      10.000 in
Prows:         66      Xmax:        7.500 in
Pcols:         75      Ymax        10.000 in
Aspect:     1.500      Rotate:      PORTRAIT
Xpixels:     1500      Paperfeed:   8.500 in
Ypixels:     2000
Background color:  WHITE
Color list:  BLACK,RED,GREEN,BLUE
Process: LINKAIBM
```

If the default parameters are correct for your device, you can create a new entry by making a copy of the LINKABLE entry. Assuming that you want to name the new driver MYDRIVER, you can do this with the following PROC GDEVICE statements:

```
PROC GDEVICE NOFS C=SASHELP.DEVICES;
             COPY LINKABLE NEWNAME=MYDRIVER;
             MOD MYDRIVER DES='COPIED LINKABLE DRIVER'
             PROCESS='processname';
             END;
```

where processname is the name of the module that you created in the section, *Generating a driver module*. For example, if you created a module name MYDRIVER MODULE, you would specify PROCESS=MYDRIVER.

In many cases, you will need to modify some of the parameters in the LINKABLE entry to fit your particular device. For example, suppose you have an 8-pen plotter with the following colors: BLACK, RED, GREEN, BLUE, ORANGE, BROWN, YELLOW, and GOLD. To create a modified device entry, use the following PROC GDEVICE statements:

```
PROC GDEVICE NOFS C=SASHELP.DEVICES;
             COPY LINKABLE NEWNAME=MYDRIVER;
             MOD MYDRIVER DES='MODIFIED LINKAIBM DRIVER'
             MAXCOLORS=9
             COLORS=(BLACK,RED,GREEN,BLUE,ORANGE,BROWN,YELLOW,GOL
             D)
             PROCESS='processname';
             END;
```

Notice that the value of MAXCOLORS includes the background color, so it should be set to one more than the number of foreground colors. The maximum number of foreground colors supported by the LINKABLE driver is 255. This is an increase over the maximum of 16 in previous releases.

Testing the Driver

To test the driver, make your FORTRAN libraries global using the following command:

```
GLOBAL LOADLIB VFLODLIB
```

Then use the program below. This program assumes that the driver writes output to the fileref FT14F001. If your plotting software writes to another location or requires that additional filedefs be provided, add or change filedefs as necessary.

```
X FI FT23F001 DISK SCR X A;
X FI FT03F001 TERM;
X FI FT06F001 TERM;
X FI FT14F001 DISK your output A;
GOPTIONS DEVICE=MYDRIVER ;
PROC GTESTIT; RUN;
```

Special Instructions for Versatec Random Users

If you are creating a driver for Versatec plotters using RANDOM software, the following changes to the instructions in the section *Generating a Driver Module* should be made to account for the differences in the software interface:

1. Instead of compiling and using the LINKAIBM program alone, you can concatenate a FORTRAN program especially designed for use with Versatec plotters using RANDOM and VGS software. The VERSAEXT program (located in the GSOURCE MACLIB data set) contains a subroutine PLOTA1, which makes calls to the VPOPT routine in RANDOM software to initialize parameters. You must modify the CALL VPOPT statement in the PLOTA1 routine and replace MODELNO with the actual model number of the plotter or printer being used. For example, if you are using a Versatec 2756 thermal printer, you should replace MODELNO with 2756 in the CALL VPOPT statement.

   The calls to the VPOPT routines also cause the RANDOM routines to be executed at the end of each graph, eliminating the need to run RANDOM as a separate step. The VERSAEXT program also contains the subroutine PLOTF2, which calls the PLOT routine with a third parameter of -999 at the end of each graph. This eliminates potential problems with RANDOM software in which only one graph is generated if multiple graphs are requested in a procedure.

2. In the linkedit step, replace the statement:

   ```
   LOAD NEWPEN PLOT PLOTS SYMBOL (RLD CLEAR
   ```

   with the following statement:

   ```
   LOAD TONE PLOT PLOTS SYMBOL (RLD CLEAR
   ```

Generating Graphics Using RANDOM Software

If you have created your driver using the LINKAIBM and VERSAEXT programs, the RANDOM routines are called from within your SAS program and you do not have to run them as a separate step.

Note the following filedefs specified in this SAS/GRAPH software job:

- The `VRFOUT` filedef contains output directed to the device.
- The `Cxxxxx` and `Exxxxx` filedefs point to the `DATPAT` files, (such as `CE3200D`), that contain information about the printer that RANDOM software requires. Not all of the `DATPAT` files are required. You need only supply the one that refers to your printer or plotter.
- The `FT06F001` filedef points to where error messages and a log of the `RANDOM` routines will be written.
- The `SASLIB` filedef points to where your driver module is stored. If your driver is a member in one of your SAS software load libraries, this statement is not required.
- The `PLOTPARM` filedef normally points to a file used by RANDOM software to specific parameters to enhance the output of your graphs. The model number should be specified in the `&PLOT` statement. Other parameters, such as `XFACT, YFACT, JWTH,` and `JCLR` can be used to control the size of your graph, the width of the "pens", and the colors used. Note that the first parameters, `JCLR(1)` and `JWTH(1)`, correspond to pen 0, which is not used by the SAS software. The second parameters, `JCLR(2)` and `JWTH(2),` correspond to pen 1, the third parameters correspond to pen 2, and so on. This means that the first and second parameter of `JCLR` and `JWTH` should be set to the same values. A typical `PLOTPARM` data set is listed below:

```
&PLOT IUSER=0,MODEL=2736,XFACT=1.00,YFACT=0.66,
    VXMIN=0.45,WXMIN=0.45,
    JWTH(1)=2,JWTH(2)=2,JWTH(3)=2,JWTH(4)=2,JWTH(5)=2,
    JWTH(6)=2,JWTH(7)=2,JWTH(8)=2,JWTH(9)=2,JWTH(10)=2,
    JWTH(11)=2,JWTH(12)=2,JWTH(13)=2,JWTH(14)=2,
    JWTH(15)=2,JWTH(16)=2,
    JCLR(1)=8,JCLR(2)=8,JCLR(3)=6,JCLR(4)=7,
    JCLR(5)=5,JCLR(6)=2,JCLR(7)=3,JCLR(8)=4,
    JCLR(9)=146,JCLR(10)=142,JCLR(11)=10,
    JCLR(12)=114,JCLR(13)=218,JCLR(14)=164,
    JCLR(15)=139,JCLR(16)=69 &END
```

Some of the parameters that you can use include the following:

`JCLR` This assigns the color that the pen will use. These colors are found in the Versatec manual. The colors used in this example are BLACK, RED, GREEN, BLUE, CYAN, MAGENTA, YELLOW, ORANGE, BROWN, GRAY, LIME, VIOLET, ROSE, TAN, AQUA. The first 7 colors are produced directly, while the remaining colors are produced by dithering techniques.

`JWTH` This controls the width of the "pen" or nib on your device. This is commonly set to 2; however other settings can be used if a thicker line is desired (such as on 400 DPI electrostatic models).

`MODEL` This indicates the model number of your printer or plotter.

`XFACT` These are factors you can use to control the size of the plot.

`YFACT` If your plot is smaller than the `HSIZE` and `VSIZE` specified in your SAS software job, increasing the `XFACT` and `YFACT` values will increase size of the plot. If your plot is larger than `HSIZE` and `VSIZE`, fractional values, (between 0 and 1), can be used to decrease the size of the plot.

A typical SAS/GRAPH EXEC using Versatec RANDOM software is listed below:

```
SETSYS    FORTVS
GLOBAL    LOADLIB    VFLODLIB
FILEDEF   PLOTPARM   DISK COLRPARM  DATA A   (PERM)
FILEDEF   FT23F001   DISK SCR       X A
FILEDEF   FT03F001   DISK FT03F001  DATA A   (PERM LRECL 132 RECFM F
FILEDEF   FT06F001   DISK ERROR     DATA A   (PERM LRECL 132 RECFM F
FILEDEF   CE3200D    DISK CE3200D   DATA *   (PERM LRECL 80 RECFM F
FILEDEF   CE3200P    DISK CE3200P   DATA *   (PERM LRECL 80 RECFM F
FILEDEF   CE3400D    DISK CE3400D   DATA *   (PERM LRECL 80 RECFM F
FILEDEF   CE3400P    DISK CE3400P   DATA *   (PERM LRECL 80 RECFM F
FILEDEF   C2502D     DISK C2502D    DATA *   (PERM LRECL 80 RECFM F
FILEDEF   C2502P     DISK C2502P    DATA *   (PERM LRECL 80 RECFM F
FILEDEF   C2504D     DISK C2504D    DATA *   (PERM LRECL 80 RECFM F
FILEDEF   C2504P     DISK C2504P    DATA *   (PERM LRECL 80 RECFM F
FILEDEF   C2506D     DISK C2506D    DATA *   (PERM LRECL 80 RECFM F
FILEDEF   C2506P     DISK C2506P    DATA *   (PERM LRECL 80 RECFM F
FILEDEF   C2508D     DISK C2508D    DATA *   (PERM LRECL 80 RECFM F
FILEDEF   C2508P     DISK C2508P    DATA *   (PERM LRECL 80 RECFM F
FILEDEF   C2706D     DISK C2706D    DATA *   (PERM LRECL 80 RECFM F
FILEDEF   C2706P     DISK C2706P    DATA *   (PERM LRECL 80 RECFM F
FILEDEF   C2706AD    DISK C2706AD   DATA *   (PERM LRECL 80 RECFM F
FILEDEF   C2706AP    DISK C2706AP   DATA *   (PERM LRECL 80 RECFM F
FILEDEF   ECP42D     DISK ECP42D    DATA *   (PERM LRECL 80 RECFM F
FILEDEF   ECP42P     DISK ECP42P    DATA *   (PERM LRECL 80 RECFM F
FILEDEF   VRFDATA    DISK VRFDATA   DATA A   (PERM XTENT 65535)
FILEDEF   VRFOUT     DISK VRFOUT    DATA A   (PERM RECFM F LRECL 132
FILEDEF   SASLIB     DISK your module fm
EXEC      SAS
```

Special Instructions for Xerox XPPI Version 1.0 Users

If you are creating a driver for Xerox printers using XPPI 1.0 software, the following changes to the instructions in the section *Generating a Driver Module* should be made to account for the differences in the software interface.

Note: If you are running XPPI version 2 or later, or using a Xerox 4850 or 4700 printer, contact SAS Technical Support for details.

1. Modify the `LINKAIBM` FORTRAN program with the following change to cause the driver to produce the output at 300 dots per inch. Change the statement "`PIXELS = 200.`" to read "`PIXELS = 300.`". This can be found in the section titled "`PREVIOUS USER 'ZAPPABLE' VALUES.`" You should also increase the `XPIXELS` and `YPIXELS` values in the device driver entry using the `GDEVICE` procedure in order to create correctly sized graphs. See the section *Customizing Device Characteristics* for more details. To calculate the number of pixels, use the following formula:

```
XPIXELS = XMAX  x 300
YPIXELS = YMAX  x 300
```

2. Instead of compiling and using the `LINKAIBM` program alone, you can concatenate a FORTRAN program especially designed for use with Xerox printers using XPPI software. The `EPICEXT` program (located in the `GSOURCE MACLIB`) contains a subroutine `PLOTA1` that is used to initialize the `XPPI` namelist parameter file and the `DJDE` file before each graphics procedure. It also contains a subroutine `PLOTC1` which is used to call the `PLOT` routine with a third parameter of `-23` at the end of each graph. This eliminates potential problems with `XPPI` software in which only one graph is generated if multiple graphs are requested in a procedure.

3. After completing the above changes, continue with the instructions in the section *Generating a Driver Module*.

Generating Graphics Using XPPI Software

This section shows some typical EXECs that can be used to set up the necessary file allocations required by XPPI software to generate output for centralized or decentralized printers. The `XPPI` routines are called from within your SAS program and you do not have to run them as a separate step.

The following file names specified in these EXECs are used by the single-phase version of `XPPI` during execution:

- The `FT02F001` file is the transportable output file that is normally sent to the device.

- The `XPPIVECT` file is an intermediate vector file used by single-phase `XPPI` to hold `DJDE` information and printer font text strings when the namelist parameter `IFNTXD` is equal to or greater than `-1`. The disposition of this file should be set to `MOD`.

- The XPPIITMP file is used for the assignment of the intermediate image file type when image files are not produced in a permanent file. The namelist parameter IOILFN(1) should be blank when intermediate image file I/O is done to the file associated with DD name XPPIITMP. This is normally used when producing output for a centralized printer.

- The XPPIILIB file is used for the assignment of the intermediate image file type when images are to be retained after XPPIP2 has finished execution. The namelist parameter IOILFN(1) should be set to a non-null value when intermediate image file I/O is performed to the file created with DDname XPPIILIB. This is normally used to produce output for decentralized printers.

- The FT06F001 file is used for the XPPI output log. The namelist parameter MSGLVD controls the amount of information that is written to the log.

- The FT07F001 file is the merge text file. It is required only if DJDE or linemode data are read and merged between images on the transportable output file (i.e., when the namelist parameter MGTEXT=1).

- The FT05F001 file is the NAMELIST parameter input for XPPI. It is used to control the default system conditions for XPPI by modifying its run-time parameters.

Sample Centralized Printer NAMELIST parameter file
(for use with Xerox 4850, 4090, 4050, 8700, or 9700 printers)

```
&XPPI IMGDSP=3,MSGLVD=7,IFNTXD=-2,IPCC=1,MGTEXT=1,
   DBRUSH=0.006666 &END
```

Sample Decentralized Printer NAMELIST parameter file
(for use with Xerox 2700, 3700, or 4045 printers)

```
&XPPI IMGDSP=6,MSGLVD=6,IFNTXD=-2,IOILFN(1)='TEMP' &END
```

Some namelist parameters that you can use include the following:

DBRUSH	(default 0.003333)	This controls the width of the pen that will be used.
DPRESO	(default 300)	This controls the resolution in dots per inch on your device.
IMGDSP	(default 1, should be set to 3 for centralized printers, 6 for decentralized printers.)	This will set the characteristics of the transportable data set. Setting the value to 3 produces on-line I/O. A value of 6 means that no image files are included in the transportable output. Refer to your XPPI manual for more details on this parameter.

IFNTXD	(default -1, should be set to -2)	This controls the processing of printer fonts. Setting the value to -2 means the printer fonts are not used.
MSGLVD	(default 1)	This controls what is written to your FT06F001 file. The value supplied is the SUM of specific options requested, for example, a value of 5 causes the IMAGE SUMMARY and NAMELIST data to print. This can be useful in debugging problems in your graphics jobs.

0	Print no messages, with the exception of error messages
1	Print image summary information
2	Print detailed information
4	List NAMELIST parameter data
8	List fonts definition file
16	List clipped/undrawable vectors
32	List user level calls
64	List vector/trapezoid generation
132	List TONE debug output.

A Typical EXEC for Centralized Printer Output

```
/*                 ************************************************************************/
/*                                                                                        */
/*   This sample shows a set-up to generate images for use on a Xerox printer             */
/*   operating in the 'batch' mode and online to the host computer.                       */
/*                                                                                        */
/*      File Definition:              Usage:                                              */
/*                                                                                        */
/*      FT23F001                      The SAS/GRAPH intermediate metafile                 */
/*      FT02F001                      The printer output file                             */
/*      FT05F001                      Namelist parameter file                             */
/*      FT06F001                      XPPI's log file                                     */
/*      FT07F001                      The DJDE definition file                            */
/*      XPPIVECT                      Intermediate vector file (temporary)                */
/*      XPPIITMP                      Intermediate image file library                     */
/*                                                                                        */
/*                 ************************************************************************/

'GLOBAL LOADLIB VSF2LOAD'     /* Global FORTRAN libraries */

'FILEDEF * CLEAR'
'FILEDEF FT23F001 DISK SCR X A'
'FILEDEF FT02F001 PRINT (RECFM UA LRECL 133'
'FILEDEF FT05F001 TERMINAL'
```

```
'FILEDEF FT06F001 DISK XPPI LOGFILE * (RECFM LRECL 132)'
```

```
'FILEDEF FT07F001 TERMINAL'
'FILEDEF XPPIITMP DISK XPPIPPIF IMGLIB'
'FILEDEF XPPIVECT DISK XPPIVECT TEMPLIB (DISP MOD'

Queue ' &XPPI IMGDSP=3,MSGLVD=7,IFNTXD=-2,MGTEXT=1,IPCC=1 &END'

Queue '  $DJDE$     JDL=XHS100,JDE=BATCH,END;'
Queue '         '
Queue '  $DJDE$     BATCH=START,END;'
Queue '  $DJDE$     IMAGE=(,.1,.1),END;'
Queue '%%%%'
Queue '  $DJDE$     BATCH=END,END;'
Queue '%%%%'

Exit 0   /* end of program */
```

After running this EXEC, the following `GOPTIONS` should be used in your SAS/GRAPH software job to execute the driver:

```
GOPTIONS DEV=XPPI;
```

Output from the SAS System job will then be written to the file referenced by `FT02F001` (the transportable output file device), which can be sent directly to a centralized printer.

A Typical EXEC for Decentralized Printer Output

```
/*                 ************************************************************************/
/*                                                                                        */
/*   This sample shows a set-up to generate images for use by a Xerox                    */
/*   decentralized laser printer.                                                         */
/*                                                                                        */
/*      File Definition:          Usage:                                                  */
/*                                                                                        */
/*      FT23F001                  The SAS/GRAPH intermediate metafile                   */
/*      FT02F001                  The printer output file                               */
/*      FT05F001                  Namelist parameter file                               */
/*      FT06F001                  XPPI's log file                                       */
/*      XPPIVECT                  Intermediate vector file (temporary)                  */
/*      XPPIILIB                  Intermediate image file library                       */
/*                                                                                        */
/*                 ************************************************************************/

'GLOBAL LOADLIB VSF2LOAD'    /* Global FORTRAN libraries */

'FILEDEF * CLEAR'
'FILEDEF FT23F001 DISK SCR X A'
'FILEDEF FT05F001 TERMINAL'
'FILEDEF FT06F001 TERMINAL'
'FILEDEF XPPIILIB DISK TEMP IMGLIB A (RECFM F LRECL 6144'
'FILEDEF XPPIVECT DISK XPPIVECT TEMPLIB (DISP MOD'

Queue ' &XPPI IMGDSP=6,MSGLVD=6,IFNTXD=-2,IOILFN(1)='TEMP' &END'
```

After running this EXEC, the following `GOPTIONS` should be used in your SAS/GRAPH software job to execute the driver:

```
GOPTIONS DEV=XPPI;
```

Output from the SAS job will then be written to the disk file `TEMP IMGLIB A` referenced on the `XPPIILIB filedef` statement. This is an `XPPI .IMG` image file that must be converted to a file containing `UDK` control codes that the decentralized printer can interpret. The `COPYIMG` utility program from Xerox can be used to perform this conversion. Converted images can either be stored in a sixel image file or printed directly on a decentralized printer. The following is an EXEC to run the `COPYIMG` step. This must be run as a separate step after the SAS system step has completed.

Note: The `COPYIMG` step must only be done if output is being generated for a decentralized printer.

A Typical EXEC for Running the COPYIMG Utility

```
/*                ********************************************************************/
/*                                                                                    */
/*  Run COPYIMG                                                                       */
/*                                                                                    */
/*                ********************************************************************/

'FILEDEF * CLEAR'
'FILEDEF FT02F001 PRINT (RECFM F BLKSIZE 80'
'FILEDEF FT05F001 TERMINAL'
'FILEDEF FT06F001 TERMINAL'
'FILEDEF INPDS DISK TEMP IMGLIB A'

Queue ' &NLIST MSGLVD=7,NAME'*',SAMPLE=1,ROT=0 &END'

'COPYIMG'

'FILEDEF * CLEAR'

Exit 0  /* end of program */
```

Common Problems and Solutions

This section describes some problems you may encounter when installing or trying to use the LINKABLE driver. If you have any problems, check the section below before calling SAS Institute.

PROBLEM:

The driver always abends when executed.

SOLUTION:

Make sure you have provided all the necessary filedefs required by your plotting subroutines prior to execution of the driver. Generally, any file allocations required when you use these routines in a FORTRAN program will also be necessary when the LINKABLE driver is executed. This includes the driver output file (usually `FT14F001`) and a file for FORTRAN error messages (`FT03F001` or `FT06F001` depending on your FORTRAN installation). If the required logical names are not available, the driver will abend.

PROBLEM:

Portions of a graph are missing when the LINKABLE driver is used with a plotter with only one pen.

SOLUTION:

The LINKABLE driver assumes that there are four pens available. If SAS/GRAPH software tries to use multiple colors and there is only one pen available, anything using a color other than the first default color may not be drawn. To circumvent this, use the `GDEVICE` procedure to change the `MAXCOLORS` value in the device entry to `2`.

PROBLEM:

Multiple graphics procedures are invoked, but only output from the first or last one comes out.

SOLUTION:

If output is written to a file, specify disposition of `MOD` on the filedef statement.

Even this may not suffice for some rare plotter control units that refuse to go past the first 999 records they encounter. Each graphics procedure causes another 999 record to be written as the procedure is terminating. If your plotter does not read past the 999 record for additional plots, you may be able to modify the plotter subroutines so that the 999th record is not written as a 999th record, but rather as a 3 record (see `Ipen` values for subroutine `PLOT`). You would then invoke a special procedure at the end of your SAS system step that would add a 999th record to the end of the plot file.

An alternative solution is to use the `NODISPLAY` option on each procedure so that the output from each procedure is stored in a graphics catalog, but is not plotted. Then at the end of the job, use `PROC GREPLAY` to produce the plotting commands in one plot file. The call to `PLOT (0, 0, 999)` is only made when the procedure is terminated.

PROBLEM:

`15A` abend codes received when attempting to run the LINKABLE driver with VS FORTRAN Release 1.4. Message received is:

```
THE FOLLOWING NAMES ARE UNDEFINED: IFYVLBCM

             DMSABN148T SYSTEM ABEND 15A
```

SOLUTION:

Version 1.4 of VS FORTRAN requires that `VFLODLIB` be global. You should global this LOADLIB before it is concatenated to your `SYSLIB` in the SAS EXEC.

PROBLEM:

AFB915 errors received when attempting to run the LINKABLE driver with VS FORTRAN release 2.4. Message received is:

```
AFB915I MODULE AFBVNREN MODULE NOT FOUND A GLOBAL LOADLIB IS
NEEDED
```

SOLUTION:

Version 2.4 of VS FORTRAN requires that VFLODLIB be global. You should make this LOADLIB global before it is concatenated to your SYSLIB in the SAS EXEC.

PROBLEM:

IEAXPSIM EXTENDED PRECISION FLOATING POINT ERRORS received when attempting to use the LINKABLE driver in a SAS System job.

SOLUTION:

This means your routines were loaded into one member instead of many members in the plotter TXTLIB. Instead of loading PLOT, PLOTS, SYMBOL, and NEWPEN, you should load the single member. For example:

```
LOAD NEWPEN PLOT PLOTS SYMBOL PLOTA2 (RLD CLEAR
```

would be replaced with:

```
LOAD CALCOMP (RLD CLEAR
```

Using SAS/GRAPH with IBM Graphics Terminals and 3270 Emulators

This section describes the steps necessary to use SAS/GRAPH software with IBM graphics terminals and 3270 emulators. Note that you do not need IBM's Graphical Data Display Manager (GDDM) product to use SAS/GRAPH software with IBM graphics terminals. GDDM is required, however, when using interactive plotters (either IEEE-attached to an IBM 3979 expansion unit on a 3179 Model G or 3192 Model G or RS-232 attached to an IBM 3472). GDDM is also required when creating GDF and ADMGDF files. For more information on the SAS/GRAPH interface to GDDM, see the section *Using SAS/GRAPH with GDDM*.

The following sections contain information about using SAS/GRAPH software with IBM graphics terminals and 3270 emulators.

- Specifying the Correct Device Driver

  This section describes which IBM terminals support graphics and which SAS/GRAPH device drivers are available.

- Customization of the Control Unit

  This section describes the customization requirements for the IBM 3274 control unit to support "presentation graphics." Most IBM 3174 control units come with graphics support and require no extra customization.

- Terminal Definitions

  This section discusses the hardware necessary to run SAS/GRAPH software on IBM graphics terminals and describes how to define the terminals to VM through the use of `TERMINAL`, `CLUSTER`, and `RDEVICE` macros.

- Requirements for 3270 Emulators

  This section explains the general requirements for displaying host graphics on a PC or PS/2 using 3270 emulation software. Also listed are several emulation packages and corresponding device drivers to use with SAS/GRAPH software.

- Support for Canadian-French Keyboards (optional)

  This section describes the necessary step to add support for the IBM 327x Canadian-French keyboards.

- Common Problems and How to Solve Them

  This section describes some of the most common problems encountered when trying to use the IBMxxxx device drivers. If you have any difficulties, please refer to this section before calling SAS Institute.

Specifying the Correct Device Driver

In Version 6 of the SAS System, SAS/GRAPH device driver names exist as entries in a SAS catalog. You can browse and manage device driver entries using `PROC GDEVICE`. The default device catalog is `SASHELP.DEVICES`. Given below is a list of IBM terminals that support graphics along with the appropriate device driver. The device driver can either be specified in the `DEVICE=` parameter of a `GOPTIONS` statement or when prompted.

Device	Driver
IBM 3179 G	IBM3179
IBM 3192 G	IBM3192
IBM 3472 G	IBM3472
IBM 3270 PC/G	IBMPCG
IBM 3270 PC/GX	IBMPCGX
IBM 3278 (with PS)	IBM3278
IBM 3279 (with PS)	IBM3279
IBM 3270 PC (with PS)	IBM3270
IBM 3290	IBM3290
IBM PC or PS/2 running GDDM-OS/2 LINK	IBMOS2CM

The IBM 3278, 3279, and 3270 PC terminals must have a Programmed Symbols (PS) adapter installed before they are capable of displaying graphics. Version 6 supports the SAS/GRAPH graphics window with the drivers listed above. You can disable the graphics window by specifying:

```
OPTIONS NOGWINDOW;
```

You can use the GRAPH window only when using the SAS Display Manager System. If GWINDOW is in effect and you are using one of the devices listed above, you do not have to specify a device driver. A default device driver is chosen for you. Note that in Version 6, most device drivers for IBM graph terminals are interchangeable. You can use the IBM3179 driver on an IBM 3472 terminal and vice-versa.

Customization of the IBM 3274 Control Unit

The IBM 3274 control unit to which your IBM terminal is connected must be properly configured in order to use the terminal for "presentation graphics. " Most IBM 3174 control units come standard with graphics support. The items below apply only to 3274 control units.

If the 3274 is pre-Configuration Support "D," the necessary customization questions and answers are as follows:

Question	Answer
161	1
162	1
163	same answer as question 112
164	1
165	1 if 3274 is a "C" model (remote), 0 otherwise

If the 3274 has Configuration Support "D" installed (for example, the newer 41 and 61 models), the questions and answers are as follows:

Question	Answer
160	1
161	1
165	1 if 3274 is a "C" model (remote), 0 otherwise

Note that attachment to IBM 3274-1B, 3274-21A, 3274-21B, 3274-21C, 3274-21D, or 3276 control units is not supported.

Terminal Definitions

In order to use SAS/GRAPH software with IBM graphics terminals, they must be defined correctly to VM. Failure to have a correct definition produces the message `Error: Device does not support graphics` or possibly a console write I/O error when you attempt to do graphics. Refer to *Planning and System Generation Guide SC19-6201,* for details on the appropriate `RDEVICE` macro for locally attached terminals or `CLUSTER` and `TERMINAL` macros for bisynchronous terminals. Two sample macros, one each for local and remote definition are presented below.

Note: These are samples only, and may not be correct for your installation. Consult the appropriate IBM publication for details.

```
SAMPLE TERMINAL AND CLUSTER MACROS FOR REMOTE TERMINALS
ATTACHED TO A 3274 CONTROL UNIT.  USER MUST ALSO SPECIFY
RDEVICE MACROS FOR LINE AND 3705 CONTROLLER

CL02F CLUSTER CUTYPE=3274,GPOLL=407F,LINE=02F
TERMINAL TERM=3278,SELECT=6040,MODEL=4
TERMINAL TERM=3279,SELECT=60C1,MODEL=3

SAMPLE RDEVICE MACROS FOR LOCALLY ATTACHED TERMINALS

RDEVICE ADDRESS=(360,01),DEVTYPE=3278,MODEL=4
RDEVICE ADDRESS=(361,01),DEVTYPE=3279,MODEL=3
```

General Requirements for 3270 Emulators

When attempting to display SAS/GRAPH output on personal computers attached to a CMS system and running 3270 emulation software, it is important to remember that not all 3270 emulation packages support mainframe graphics. Some 3270 emulation packages do not support mainframe graphics at all, some support mainframe graphics by default, and some require an additional graphics "module" before they can support mainframe graphics. For those 3270 emulators that support mainframe graphics, most emulators require you to configure your emulator for host graphics support before you can display mainframe graphics back to your screen. In addition to having a 3270 emulator that supports mainframe graphics and is configured to support mainframe graphics, it must also be defined to VM as a device that supports extended data streams.

If your emulation package is configured for mainframe graphics support, the typical SAS/GRAPH device driver to use to display graphics on your screen is the IBM3179 device driver. As an alternative to the IBM3179 device driver, you can use the GDDMPCG device driver if the GDDM base product is installed on your system.

Support for Canadian-French Keyboards (optional)

Canadian-French keyboards have accent keys that result in composite characters when used with certain letters. These accent keys are considered "dead keys" unless used with a second valid character; therefore, the `EBCDIC` code points that represent them may not be used as a programmed symbol code point. To remove these five code points from use with the programmed symbols data stream, specify the `$FSOPT2` system option when invoking the SAS System as follows:

```
SAS ($FSOPT2
```

Common Problems and How to Solve Them

This section describes some of the most common problems encountered when installing or trying to run SAS/GRAPH software on IBM graphics terminals. If you have any problems, please check the section below before calling SAS Institute.

PROBLEM:

`PROG 470 or PROG 471` error on 3278 and 3279 terminals

SOLUTION:

Make sure the terminal has the graphics option installed. If you get this error on a 3278 or 3279 graphics terminal, do the following:

1. Press `RESET` and then `ALT/TEST`.

2. Key in `/6` and press `ENTER`. This causes several lines to appear on the upper left part of the screen. The second of these lines will initially be `00`.

3. Press `ENTER`. This causes the second line to change to `04`. Keep hitting `ENTER` until the second line equals `14`.

4. There will be several lines of hexadecimal under the second line. Get the first 4 half words of the bottom line and have them ready when you call SAS Institute to report the problem.

PROBLEM:

```
Error: Device does not support graphics.
```

SOLUTION:

First, make sure that the device you are using is a graphics device. If you are using a PC with emulation software, be sure that the emulator is designed to support host graphics. Second, check to make sure that your terminal is defined correctly to VM. Third, make sure that the control unit to which your device is attached supports graphics data streams.

PROBLEM:

Incomplete graphs with question marks or solid blocks at the bottom.

SOLUTION:

This occurs when programmed symbols have been exhausted, and is commonplace when the terminal has PS-2. The problem can be avoided by upgrading to PS-4. (However, on very complex graphs, the problem may still occur with PS-4).

PROBLEM:

`PROG 401` or `PROG 474` error.

SOLUTION:

Make sure that items 160-165 on your 3274 control unit have been set correctly. See the section *Customization of the 3274 Control Unit* for details.

PROBLEM:

Graphs on an IBM 3279 Model 2 do not fill the entire screen.

SOLUTION:

This results because the Model 2 has only 24 lines. If SAS/GRAPH software were to try to use the entire screen, there would be gaps in the graph; to avoid this problem, only about 2/3 of the screen is used.

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

This section describes the steps necessary for using SAS/GRAPH software with IBM 3287, 3268, and 4224 printers using stand-alone (non-GDDM) device drivers. As alternatives to the stand-alone device drivers, device drivers for IBM 3287, 3268, and 4224 printers are available which interface with IBM's GDDM product. If you choose to use a GDDM device driver instead of a stand-alone device driver, see the section *Using SAS/GRAPH Software with GDDM*.

As with other SAS/GRAPH device drivers, the IBM3287, IBM3268, and IBM4224 drivers exist as entries in the default device catalog, `SASHELP.DEVICES`. You can browse and manage device driver entries using `PROC GDEVICE`.

The IBM 3287 and 3268 printers are four-color dot matrix printers that are capable of producing graphics. These devices may only be used along with IBM 3274 or 3174 control units capable of supporting graphics devices. 3287 and 3268 printers must be capable of producing graphics, and must have the Extended Character Set Adapter and Programmed Symbols (PS) installed. While SAS/GRAPH software will work with either PS-2, PS-4, or PS-4A, we strongly recommend that PS-4A be installed to provide smooth graphics output.

The IBM 4224 Model 2E2 is a monochrome graphics printer while the IBM 4224 Model 2C2 is a color graphics printer. The 4224 Model 2C2 supports either 4 colors or 8 colors (depending on the type of ribbon installed in the printer). The IBM 4224 printer does not support PS graphics but instead supports Intelligent Printer Data Stream (IPDS).

IBM printers must be attached (either locally or remotely) to an IBM 3274 or 3174 control unit. For 3274 control units, models 41A, 41C, 41D and 61C are pre-configured for graphics hardware support and have fully expanded memory. Configuration Support 'D' is standard in these devices. All other IBM 3274 control units must have Configuration Support 'C' (Extended Function Store). Most IBM 3174 control units come standard with graphics support. The items below apply only to 3274 control units.

If the 3274 is pre-Configuration Support "D", the necessary customization questions and answers are as follows:

Question	Answer
161	1
162	1
163	same answer as question 112
164	1
165	1 if 3274 is a "C" model (remote), 0 otherwise
22	must be either 0 (System) or 2 (Shared) for the 'Printer Mode' column of the Printer Authorization Matrix

If the 3274 has Configuration Support "D" installed (for example, the newer 41 and 61 models), the questions and answers are as follows:

Question	Answer
160	1
161	1
165	1 if 3274 is a "C" model (remote), 0 otherwise
022	must be either 0 (System) or 2 (Shared) for the 'Printer Mode' column of the Printer Authorization Matrix

Note: Attachment to IBM 3274-1B, 3274-21A, 3274-21B, 3274-21C, 3274-21D, or 3276 control units is not supported.

Using SAS/GRAPH Software with IBM 3287 and 3268 Printers

You can use the IBM3287 and IBM3268 device drivers in interactive or noninteractive modes. In most cases, you attach the 3287 or 3268 printer to the user's virtual machine. As an alternative, you can use RSCS Networking Version 2 to spool graphics output to IBM 3287 and 3268 printers.

To use an IBM 3287 or IBM 3268 printer with SAS/GRAPH software when the printer is attached to the user's virtual machine, specify the following:

```
GOPTIONS DEVICE = IBM3287   (or DEVICE=IBM3268)

         DEVADDR = 'cuu';
```

where `cuu` is the virtual device address of the 3268 or 3287 printer that is attached to your machine.

Note: The virtual address should be enclosed in single quotes.

You may omit the `DEVADDR` parameter from the `GOPTIONS` statement, in which case the driver assumes the printer is attached at address 061. An operator (or someone else with the proper authority) must issue the `CP ATTACH` command for local 3268 or 3287 printers or the `NETWORK ATTACH` and `NETWORK ENABLE` commands for remote 3268 or 3287 printers.

Users with remote printers should make sure that when the `NETWORK ENABLE` command is issued, the printer is ready for communications so that when VM/SP sends the read-partitioned query data stream to the printer, it can respond with the query reply instead of `INTERVENTION REQUIRED.`

Spooling Graphics Using RSCS

When RSCS Networking Version 2 (or later) is installed, IBM3287 or IBM3268 device driver output may be spooled to an RSCS printer that supports graphics. To use the IBM 3268 or 3287 printer with RSCS, specify the following:

```
X  CP SPOOL PUN TO RSCS NOH CONT;
X  CP TAG DEV PUN prtid 50 PRT=GRAF;
FILENAME GSASFILE PUNCH;
GOPTIONS DEVICE = drivername
         GSFNAME = GSASFILE
         GSFMODE = REPLACE
         GSFLEN = 80
         DEVADDR = '00D';
```

where `prtid` is the RSCS network name for the printer and `drivername` is one of the entries in the following table:

Driver	Description	
IBM3287C	IBM 3287 with PS-4A and 11 inch	(vertical) paper
IBM3287B	IBM 3287 with PS-4 and 11 inch	(vertical) paper
IBM3287A	IBM 3287 with PS-2 and 11 inch	(vertical) paper
IBM328SC	IBM 3287 with PS-4A and 8.5 inch	(vertical) paper
IBM328SB	IBM 3287 with PS-4 and 8.5 inch	(vertical) paper
IBM328SA	IBM 3287 with PS-2 and 8.5 inch	(vertical) paper
IBM3268C	IBM 3268 with PS-4A and 11 inch	(vertical) paper
IBM3268B	IBM 3268 with PS-4 and 11 inch	(vertical) paper
IBM326SC	IBM 3268 with PS-4A and 8.5 inch	(vertical) paper
IBM326SB	IBM 3268 with PS-4 and 8.5 inch	(vertical) paper

When you have finished creating the graphics spool file, you can send it to RSCS for processing by specifying the following:

```
X   CP SPOOL PUN CLOSE;
```

Using SAS/GRAPH Software with an IBM 4224 Printer

To use SAS/GRAPH software with an IBM 4224 graphics printer, specify the following:

```
GOPTIONS DEVICE  = IBM4224;   (OR IBM42241, OR IBM42244)
```

If you are using paper size that is 8.5 inches vertically, you can use the IBM424S, IBM424S1, or IBM424S4 device drivers. The IBM4224 and IBM424S drivers support eight colors (BLUE, RED, MAGENTA, GREEN, CYAN, YELLOW, BLACK, and BROWN). The IBM42241 and IBM424S1 drivers support one color (BLACK), and the IBM42244 and IBM424S4 drivers support four colors (BLUE, RED, GREEN, and BLACK).

Spooling Graphics Using RSCS

When RSCS Networking Version 2 (or later) is installed, IBM4224 device driver output may be spooled to an IBM 4224 printer defined to RSCS. To use an IBM 4224 printer with RSCS, specify the following:

```
X CP SPOOL PUN TO RSCS NOH CONT;
X CP TAG DEV PUN prtid 50 PRT=GRAF;
FILENAME GSASFILE PUNCH;
GOPTIONS  DEVICE  = IBM424R
               GSFNAME = GSASFILE
               GSFMODE = REPLACE;
```

The `PRTID` parameter on the `CP TAG` command is the RSCS network name for the printer. If you are using paper size that is 8.5 inches vertically, you can use the IBM424RS device driver instead of the IBM424R driver.

When you have finished creating the graphics spool file, you can send it to RSCS for processing by specifying the following:

```
X CP SPOOL PUN CLOSE;
```

Common Problems and How to Solve Them

This section describes some of the most common problems encountered when installing or trying to use SAS/GRAPH software with IBM graphics printers. If you have any problems, see the section below before calling SAS Institute Technical Support.

PROBLEM:

Rough boundaries between filled areas on graphs. For example, diagonal lines on pie charts look like stair steps.

SOLUTION:

This is a hardware limitation, resulting from having either PS-2 or PS-4, rather than PS-4A on your 3268 or 3287. To determine which of these features you have installed, run Off-line Diagnostic Test 5 described in the IBM 3268 or 3287 Maintenance Information Manual.

PROBLEM:

Distorted output on 3268 or 3287 when hard copy key is hit on IBM3279.

SOLUTION:

The hard copy key on the 3279 cannot be used to produce graphics on the 3268 or 3287. You must use the IBM3268 or IBM3287 device driver.

PROBLEM:

I/O error with `CSWSTAT=0600 (or 0E00)` and `SENSE=01.`

SOLUTION:

Run Off-line Test 5 described in the 3268 or 3287 Maintenance Information Manual (located in the back of the printer) to verify the existence of programmable symbols on the machine. Also, check the Printer Authorization Matrix in the 3274 configuration; `SYSTEM (0)` or `SHARED (2)` mode must be specified for the printer to allow the host to load the symbol sets.

PROBLEM:

I/O error with `CSWSTAT=0E00` and `SENSE=80`.

SOLUTION:

Check the Printer Authorization Matrix as described above. Also, remote 3268 or 3287 printers must be powered on before issuing the `NETWORK ENABLE` command in order for the system to recognize the "advanced features" of the device.

Using SAS/GRAPH Software with GDDM

SAS Institute provides a set of drivers which 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 and requires no special action 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 drivers, the GDDM base product, Version 1.4 or 2.x, must be installed on your system. Support for the IBM 4224 graphics printer requires GDDM Version 2.x.

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
```

GDDM Nicknames

GDDM nicknames are device definitions which tell GDDM what type of device you are using, what type of graphics output to generate, and where to send the graphics output. You enter GDDM nickname definitions in a CMS file with the filename of PROFILE and a filetype of ADMDEFS. The PROFILE ADMDEFS file should have a record format of fixed block and a record length of 80.

The information in the PROFILE ADMDEFS file must be in uppercase and begin in column 2. Column 1 should be blank (or include an asterisk to indicate a comment). Nickname definitions are not required for all GDDM drivers. The drivers that require nickname definitions are those drivers that support IEEE-attached plotters, IBM 3800-type printers, and GDDM/PCLK. See the individual device sections for sample nicknames.

GDDM Tokens

GDDM tokens are default device definitions stored in load module format when GDDM is installed. For graphics terminals, IEEE-attached plotters, IBM 3287, 3268, and 4224 printers, and GDDM/PCLK, the default tokens are located in the file ADMLSYS1 ASSEMBLE on your GDDM system disk. Tokens for IBM 3800-type printers are located in the file ADMLSYS4 ASSEMBLE on your GDDM system disk.

Please note that not all GDDM device drivers require the use of a device token. The only device driver that requires the use of a token is the GDDMFAM4 driver (which you use for IBM 3800-type printers). The token is passed to GDDM using either the GDDMT = (or GDDMTOKEN =) parameter in the SAS GOPTIONS statement or the DEVTOK= parameter in the device nickname. See the section on IBM 3800-type printers for a further discussion on device tokens.

GDDM Drivers for Graphics Terminals

If your graphics device supports All-Points-Addressable (APA) graphics and 7 colors (such as the IBM 3179 Model G, 3192 Model G, or the 3270 PC/G), specify the following:

```
GOPTIONS DEVICE = GDDMPCG;
```

For the IBM 3270 PC/GX (which supports 15 colors), specify the following:

```
GOPTIONS DEVICE = GDDMPCGX;
```

For graphics devices that support PS graphics and 7 colors (such as the IBM 3279 Model S3G and 3270 PC), specify the following:

```
GOPTIONS DEVICE = GDDM79;
```

For monochrome graphics devices that support PS graphics (such as the IBM 3278 Models 2 and 4 with Programmed Symbols), specify the following:

```
GOPTIONS DEVICE = GDDM78;
```

To terminate a graph that is displayed on the screen of a graphics terminal, press Enter, PF3, or PF15. Please note that the GDDM drivers do not support the Version 6 SAS/GRAPH graphics window.

GDDM Drivers for IEEE-Attached Plotters

The GDDMPLT2, GDDMPLT6, and GDDMPLT8 drivers can be used to drive 2, 6, and 8-pen plotters attached to an IBM 3179 Model G, 3192 Model G, 3270 PC/G, or 3270 PC/GX using an IEEE interface. There are two methods for sending graphics output to an IEEE-attached plotter. The first method sends the output directly to the plotter. With the second method, the graph is displayed on the screen of the terminal and then sent to the plotter by pressing PF4 or PF16.

To send output directly to the plotter in an interactive session, use the following `GOPTIONS` statement:

```
GOPTIONS DEVICE = GDDMPLTx GDDMN = SASPLOT;
```

The driver suffix should match the number of pens in the plotter. The name you specify on the `GDDMN=` option should match a nickname (`SASPLOT`, in this case) that you define within your GDDM Admdefs nickname file (described earlier in this section). Define the nickname of `SASPLOT` in the `PROFILE ADMDEFS` nickname file as follows:

```
NICKNAME        NAME=SASPLOT,
                TOFAM=1,
                TONAME=(*,ADMPLOT)
```

The keyword `NICKNAME` should begin in column 2 of the `ADMDEFS` file and the value for the `NAME=` parameter should match the name specified in the `GDDMN=` graphics option within the SAS System job.

The second method for producing SAS/GRAPH software output on an IEEE-attached plotter is to display a graph initially on the screen of the terminal and then to send the graph to the plotter by pressing PF4 or PF16. For this method, specify the following:

```
GOPTIONS        DEVICE = GDDMPCG
                GDDMN = *
                DEVADDR = (.,.,SASPLOT);
```

The nickname for the plotter is specified as the last parameter in the `DEVADDR=` option. Define the nickname of `SASPLOT` in the `PROFILE ADMDEFS` nickname file as follows (this is the same nickname definition as given above when sending graphics output directly to the plotter):

```
NICKNAME        NAME=SASPLOT,
                TOFAM=1,
                TONAME=(*,ADMPLOT)
```

GDDM Drivers for the IBM 3287, 3268, and 4224 Printers

As alternatives to the IBM3287, IBM3268, and IBM4224 drivers, GDDM drivers are available which produce graphics output on IBM 3287, 3268, and 4224 graphics printers. To use these drivers, specify the following:

```
GOPTIONS DEVICE = GDDM87 (or GDDM68, or GDDM4224)

         GDDMN = CMS filename or device nickname;
```

If your printer has paper size that is 8.5 inches vertically, you can use the GDDM87S, GDDM68S, or GDDM424S device driver. The GDDM87 and GDDM68 drivers support 4 colors while the GDDM4224 driver supports 8 colors. The `GDDMN=` parameter is equivalent to specifying `DEVADDR=` (CMS filename or device nickname). The `GDDMN=` parameter can be used in one of two methods. With the first method, specify the CMS filename that you want GDDM to use when creating the Admprint graphics file. The default filetype is `ADMPRINT` and the default filemode is A. To send an Admprint file to a printer directly attached to the user's virtual machine, specify the following:

```
ADMOPUV filename (ON cuu)
```

The `filename` you specify is the filename of the `ADMPRINT` file that the driver creates. The `cuu` parameter is the virtual address of the printer attached to your machine and is not needed if the printer is attached to the default address of 061. As an alternative, you can specify `GDDMN=` device-nickname. This method is most commonly used when using RSCS to spool graphics output to a printer as follows:

```
GOPTIONS DEVICE = GDDM87 (or GDDM68, or GDDM4224) GDDMN =
PRTRSCS;
```

Define the nickname of `PRTRSCS` in the `PROFILE ADMDEFS` nickname file as follows:

```
NICKNAME        NAME=PRTRSCS,
                FAM=1,
                TONAME=PUNCH,
                DEVTOK=device token,
                PROCOPT=((CPSPOOL,TO,RSCS),
                           (CPTAG,node,prtid,50,PRT=GRAF))

NICKNAME        NAME=PRTRSCS,
                FAM=2,
                DEVTOK=device token,
                PROCOPT=((INVKOPUV,YES))
```

Check the `ASMLSYS1 ASSEMBLE` file on your GDDM system disk for valid device tokens. The `node` parameter specified after the `CPTAG` parameter is the name of the system to which the printer is attached. The `node` parameter is optional. The `prtid` parameter is the RSCS network name for the printer.

GDDM Drivers for the IBM 3812 Model 2 Printer (with IPDS)

The IBM 3812 Model 2 printer is available with the Intelligent Printer Data Stream (IPDS) feature. The IPDS feature allows the 3812 Model 2 printer to support the same type of graphics that the IBM 4224 printer supports. To send SAS/GRAPH output to the IBM 3812 Model 2 printer with the IPDS feature, specify the following:

```
GOPTIONS DEVICE = GDDM381P (or GDDM381L) GDDMN = IBM3812;
```

The GDDM381P driver produces graphics output in portrait format while the GDDM381L driver produces graphics output in landscape format.

Define the nickname of IBM3812 in the `GDDM ADMDEFS` nickname file as follows:

```
NICKNAME        NAME=IBM3812,
                TOFAM=2,
                DEVTOK=X3812Q,
                TONAME=(CMS filename)
```

The token of X3812Q is for a 3812 printer attached to a non-SNA control unit. Check the file `ADMLSYS1 ASSEMBLE` on your GDDM system disk for additional 3812 device tokens.

GDDM Drivers for IBM 3800-type Printers

Use the GDDMFAM4 device driver to produce SAS/GRAPH output on high-resolution graphics printers such as the IBM 3800 and 3820 laser printers. The output produced by the GDDMFAM4 driver is a bit-image, device-dependent file known as an `ADMIMAGE` file. This file can be sent directly to a high resolution printer using a print utility such as IBM's Print Services Facility/VM (PSF/VM) or the `PRT3812 EXEC`.

By default, the IBM 3800 printer produces graphics output in landscape mode on paper that is 8.5 inches vertically and 11 inches horizontally. To produce SAS/GRAPH output on an IBM 3800 printer, specify the following:

```
GOPTIONS DEVICE = GDDMFAM4 GDDMN = IBM3800 GDDMT = IMG240X;
```

The token of IMG240X is a default token supplied by IBM for a 3800 laser printer. Other default tokens for 3800-type printers can be found in the file `ADMLSYS4 ASSEMBLE` on the GDDM system disk.

Define the nickname of IBM3800 in the `PROFILE ADMDEFS` nickname file as follows:

```
NICKNAME NAME=IBM3800,
         TONAME=(filename,filetype,filemode),
         PROCOPT=((CDPFTYPE,PRIM),
                  (HRIFORMT,CDPF),
                  (HRISPILL,NO),
                  (HRISWATH,10),
                  (HRIPSIZE,110,75,TENTHS))
```

Most other IBM 3800-type printers (such as the 3820) produce graphics output in portrait format on paper that is 11 inches vertically and 8.5 inches horizontally. To produce SAS/GRAPH output on a 3820-type printer, specify the following:

```
GOPTIONS DEVICE = GDDMFAM4 GDDMN = IBM3820 GDDMT = IMG240;
```

The token of `IMG240` is a default token supplied by IBM for a 3820 laser printer. Define the nickname of IBM3820 in the `PROFILE ADMDEFS` nickname file as follows:

```
NICKNAME        NAME=IBM3820,
                TONAME=(filename,filetype,filemode),
                PROCOPT=((CDPFTYPE,PRIM),
                          (HRIFORMT,CDPF),
                          (HRISPILL,NO),
                          (HRISWATH,10),
                          (HRIPSIZE,80,105,TENTHS))
```

The `TONAME=` parameter in the nickname tells GDDM what filename, filetype, and filemode to use when writing the `ADMIMAGE` file to disk.

Setting the `CDPFTYPE` parameter in the nickname to `PRIM` (which is the default) tells GDDM to write out an `ADMIMAGE` file as a primary page document. Primary page documents are normally printed by themselves, one to a page. To create a secondary page segment (which can be included in a DCF or Script document), set the `CDPFTYPE` parameter in the nickname to `SEC`.

Setting the `HRIFORMT` parameter in the nickname to `CDPF` tells GDDM to create the `ADMIMAGE` file as a structured bit-map file (which most printers use). If you need an unformatted bit-map file, set the `HRIFORMT` parameter in the nickname to `BITMAP.`

Setting the `HRISPILL` parameter in the nickname to `NO` tells GDDM not to use an external spill file when building the `ADMIMAGE` bit-map in memory. If the `HRISPILL` parameter is set to `YES` (which is the default), SAS/GRAPH software procedures that normally produce multiple graphs produce only one graph.

The `HRISWATH` parameter in the nickname tells GDDM how many horizontal 'slices' to divide the picture into when building the bit-map in memory. Increasing this number causes the job to use less memory but more CPU time. A good overall value is 10.

The `HRIPSIZE` parameter in the nickname determines the actual graph size. The first value is the width of the page (in tenths of inches). The second value is the depth of the page (in tenths of inches). Use the `HRIPSIZE` parameter to reduce the size of the graphics image (especially when creating secondary page segments). For example, use an `HRIPSIZE` setting of `50,40` for a graph size of 5 inches horizontally and 4 inches vertically.

Creating GDF and ADMGDF Files

Graphics Data Format (GDF) and ADMGDF files are device-independent files containing general graphics orders and commands. You can create GDF and ADMGDF files with SAS/GRAPH software and GDDM but you cannot currently display or import GDF and ADMGDF files into the SAS System. To create a GDF file, specify the following:

```
X FILEDEF ADMPIF DISK SASGRAF GDF A (LRECL 400 RECFM F ;

GOPTIONS        DEVICE = GDDMPCG
                GSFNAME = ADMPIF
                GSFMODE = REPLACE
                GSFLEN = 400
                CHARTYPE = 2
                NODISPLAY;
```

Setting GSFNAME=ADMPIF causes the GDF file to be written to the file with a libref of ADMPIF.

ADMGDF files are similar to GDF files except that ADMGDF files contain header information in the first 16 bytes of each record. The first eight bytes of each record contain the filename and the second eight bytes contain the filetype (which is always ADMGDF). To create an ADMGDF file, specify the following:

```
GOPTIONS        DEVICE = GDDMPCG
                GSFNAME = GRAF
                GSFMODE = REPLACE
                NODISPLAY
                ADMGDF;
```

The filename is automatically created by appending a sequence number to the end of the GSFNAME= option above. For example, the GOPTIONS statement above followed by a graphics procedure creates an ADMGDF file with the filename GRAF0000 and a filetype of ADMGDF.

Using SAS/GRAPH Software with GDDM/PCLK

GDDM/PCLK is terminal emulation software for a PC or PS/2 that provides mainframe graphics support. GDDM/PCLK requires the following three parts:

- ❑ The GDDM base product (installed on the host)
- ❑ The GDDM PCLKF feature (installed on the host)
- ❑ The GDDM/PCLK product (installed on the PC or PS/2).

To display graphics on the monitor of the PC or PS/2 running GDDM/PCLK, specify the following:

```
GOPTIONS DEVICE = GDDMPCG; (or GDDMPCGX for 15 color support)
```

In the PROFILE ADMDEFS nickname file, specify the following:

```
NICKNAME FAM=1,PROCOPT=((PCLK,YES))
```

To send SAS/GRAPH output to attached printers using GDDM/PCLK, specify the following:

```
GOPTIONS DEVICE = GDDMPCG GDDMN = PCPRINT;
```

Define the nickname of PCPRINT in the PROFILE ADMDEFS nickname file as follows:

```
NICKNAME        NAME=PCPRINT,

                FAM=0,
                TOFAM=1,
                DEVTOK=device token,
                TONAME=(*,ADMPCPRT),
                PROCOPT=((PCLK,YES))
```

To send SAS/GRAPH output to attached plotters using GDDM/PCLK, specify the following:

```
GOPTIONS DEVICE = GDDMPLT8 GDDMN = PCPLOT;
```

Define the nickname of PCPLOT in the PROFILE ADMDEFS nickname file as follows:

```
NICKNAME        NAME=PCPLOT,
                FAM=0,
                TOFAM=1,
                DEVTOK=device token,
                TONAME=(*,ADMPLOT),
                PROCOPT=((PCLK,YES))
```

Consult the GDDM/PCLK documentation for a list of valid device tokens to specify in the DEVTOK= nickname parameter.

Consult GDDM Base Programming Reference manuals (SC33-0332) or the GDDM Installation and System Management (SC33-0321) or the GDDM Application Programming Guide (SC33-0337) for further information about GDDM calls.

Appendix F, Implementing SAS/ACCESS® Interface to ORACLE

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

Linking with the ORACLE RDBMS Interface Subroutines (required)

The modules, SASORA6 and SASORA7 shipped with the SAS/ACCESS Interface to ORACLE software, provide ORACLE Call Interface (OCI) support. To prepare the software for execution, it must be linked with the ORACLE (OCI) support subroutines. You have the option of linking the ORACLE Version 6 subroutines, the ORACLE Version 7 subroutines, or both.

To perform this step you can use the SASORA exec that is supplied with the SAS/ACCESS software.

Note: If you install additional ORACLE products, or if you install a new version of the ORACLE RDBMS at your site, you should perform this step again.

- ❑ Access the ORACLE disk (read-only) that contains the `V6ORALNK, TXTLIB` and/or the `V7ORALNK TXTLIB`. These textlibs contain the ORACLE (OCI) routines that will be linked with the SAS/ACCESS software.

- ❑ Access the SAS System disk (write) that contains the SAS/ACCESS loadlib `SASDBIOR`, the SASORA EXEC, and the `SASORAT` and `SASORAT7 TEXT` files.

- ❑ Execute the SASORA EXEC as follows to create the appropriate load module:

  - ❑ `SASORA`

    includes the `ORAV6` load module and the ORAV7 load module in the `SASDBIOR LOADLIB`

  - ❑ `SASORAV6`

    includes only the `ORAV6` load module in the `SASDBIOR LOADLIB`

  - ❑ `SASORAV7`

    includes only the `ORAV7` load module in the `SASDBIOR LOADLIB`.

Note: There may be warnings or errors generated from the linkage editor. This is expected and will not prevent the image from linking.

Appendix G, Implementing SAS/SHARE® Software

Creating a VM Directory Entry for the Server's Virtual Machine

For each SAS server, you must create a VM directory entry for the virtual machine in which the server will run. A typical VM directory entry for a SAS server virtual machine is shown below:

```
USER SASSHARE XXXXXXX 20M 20M G 100
MACHINE XA
IPL CMSXA PARM AUTOCR
OPTION MAXCONN 1024 REALTIMER
IUCV ALLOW PRIORITY MSGLIMIT 255
CONSOLE 009 3215
SPOOL   00C 2540 READER *
SPOOL   00D 2540 PUNCH  A
SPOOL   00E 1403 A
LINK MAINT 19E 19E RR
LINK MAINT 19D 19D RR
LINK MAINT 190 190 RR
MDISK 191 3380 707 5 VM0800 MR XXXXXXX
MDISK 192 3380 501 3 VM0450 MR XXXXXXX
```

Descriptions of some of the directory records of the VM directory entry for a SAS server virtual machine are given below.

USER directory record

The default virtual storage size and the maximum virtual storage size of the server's virtual machine should be the same because it is impractical to stop the server, define its virtual machine's storage to a larger value, and then start the server again.

The amount of virtual storage required by a server depends on many factors, which include, but are not limited to, the following:

- ❏ whether the SAS supervisor and base procedures are installed in saved segments
- ❏ the number of users who access the server simultaneously
- ❏ the internal data set pagesize of the data sets shared through the server (as reported by the `CONTENTS` procedure)
- ❏ the number of SAS files each user accesses at one time through the server.

In the example presented at the beginning of this appendix, the server executes in a 20-megabyte virtual machine. Use this value initially, then increase (or decrease) the virtual storage if warranted by your use of the SAS server. SAS Institute has executed SAS servers in XA-mode virtual machines and in 370-mode virtual machines. If you are running a VM/XA operating system and would like to give your SAS server(s) plenty of virtual storage, you have the option of using one or more XA-mode virtual machines for your SAS servers. A SAS server requires no CP command privilege class higher than G. The dispatching priority of the server's virtual machine should initially be the default value for interactive VM users. Experience will help you to determine whether a higher or lower dispatching priority is appropriate for your use of the SAS server.

IPL directory record

The server's virtual machine should IPL CMS (or CMSXA). It is not necessary to specify `PARM AUTOCR` if the class A or B CP command `AUTOLOG` is used to log on the SAS server because the `AUTOLOG` command passes a line of data to satisfy the server virtual machine's first console read. However, `PARM AUTOCR` is convenient when executing the SAS server interactively. No negative effects have been observed when `PARM AUTOCR` is specified and the SAS server virtual machine is logged on by an `AUTOLOG` command.

OPTION directory record

SAS/SHARE software does not currently require the `REALTIMER` option, but it might in the future. Therefore, it might be more convenient for you to specify this option now while defining the server's virtual machine. The effect of the `REALTIMER` directory option can be duplicated by including the CP command `SET TIMER REAL` in the PROFILE EXEC of the server's virtual machine.

MDISK directory records

Since the modules that make up SAS/SHARE software under CMS reside on your SAS System or SAS/SHARE minidisk, the server requires no minidisk space of its own for program storage. When you configure your first SAS server, you may want to use the single read- write minidisk at virtual address 191, file-mode A. The minidisk needs to be large enough to hold all the SAS data sets that are shared through the server. If you already have SAS data sets that you copy to the server's minidisk, the minidisk must be large enough to contain all the data sets; additionally, you should allow some extra room for growth. Alternatively, you can allocate several minidisks to a SAS server. This enables you to divide space according to department and/or application system. If a DATA step or procedure is used to replace server data sets, you must allow space for two copies of each data set that a DATA step or procedure replaces. You rarely need to allow space for all server data sets on a minidisk to be replaced at the same time. Instead, find the DATA step or procedure that replaces the largest server data sets and allow room for two simultaneous copies of those data sets.

Selecting Communications Access Methods to Use

Note: This task is required.

STEP 1: Determine the access method(s) to use.

Communication between a SAS/SHARE server and user is handled by a part of SAS software called a communications access method. The following communications access methods are available for use with this release of SAS/SHARE software under CMS:

- ❑ IUCV
- ❑ APPC
- ❑ TCP/IP.

To use the IUCV access method, a server and user must be running on the same CMS system.

You can choose to use one access method exclusively, or you can choose one as your primary access method and others as secondary access methods. If you choose to define primary and secondary access methods, SAS/SHARE software will attempt to establish a user-to-server connection using the primary access method first. If that attempt fails, SAS/SHARE software will then attempt to establish the connection using each of the secondary access methods in turn.

You should choose the access methods you will use based on your site's requirements and restrictions.

STEP 2: Set SAS System options to specify selected access methods.

The SAS System option `COMAMID=` specifies which access method SAS/SHARE software should use as the primary or only access method. The SAS System options `COMAUX1=` and `COMAUX2=` specify secondary access methods. These options are specified, typically in a SAS System configuration file, by the SAS/SHARE Software Consultant.

The following table shows the value of these options for each access method:

Access Method	`COMAMID=/COMAUX1=/COMAUX2= Value`
IUCV	`IUCV`
APPC	`APPC`
TCP/IP	`TCP`

For a server, these three options have essentially the same meaning; each access method specified by these options will be initialized when the server is started, making the server accessible to users via any of those access methods.

For example, for a server that is to be accessible only to users who use the IUCV services access method, specify

```
COMAMID=IUCV
COMAUX1=
COMAUX2=
```

For a server that is to be accessible to users who use either the IUCV access method or the APPC access method, specify

```
COMAMID=IUCV
COMAUX1=APPC
COMAUX2=
```

or

```
COMAMID=APPC
COMAUX1=IUCV
COMAUX2=
```

For a user session, the access method specified by the `COMAMID=` option is the first one used to attempt to connect to a server. If the server is not found, the access method specified by the `COMAUX1=` option is used. If the server still is not found, the access method specified by the `COMAUX2=` option is used.

For example, to cause a user session to use only the IUCV access method, specify

```
COMAMID=IUCV
COMAUX1=
COMAUX2=
```

Note that it is not necessary to specify `COMAUX1=` or `COMAUX2=` if you do not want to specify a secondary access method.

To cause a user session to first try to locate a server using the IUCV access method and then to use the APPC access method if the server is not found, specify

```
COMAMID=IUCV
COMAUX1=APPC
COMAUX2=
```

To cause a user session to try the IUCV, APPC, and TCP/IP access methods, in that order, specify

```
COMAMID=IUCV
COMAUX1=APPC
COMAUX2=TCP
```

System Configuration for the IUCV Access Method

Modifying the VM Directory Entry for the Server

You must include the following statements in the VM directory entry for the server's virtual machine:

```
OPTION MAXCONN 1024
```

specifies the maximum number of simultaneous connections to the server. As a rule of thumb, you should allow for 4-5 connections for each user who accesses data through the server. The number shown here is only a guideline; the default `MAXCONN` value is `4` and the maximum `MAXCONN` value is `65535`. SAS/SHARE software does not specifically limit the number of simultaneous connections to a SAS server.

```
IUCV ALLOW PRIORITY MSGLIMIT 255
```

The `ALLOW` parameter enables users to establish IUCV connections to the server's virtual machine. The `PRIORITY` parameter enables the SAS server to send IUCV priority messages to users. The `MSGLIMIT` parameter increases the number of outstanding messages allowed on each path from the default value of `10` to `255`.

System Configuration for the APPC Access Method

Software Requirements

To use the APPC access method, you must have a release of VM that has APPC support:

- ❑ VM/SP Release 6
- ❑ VM/ESA.

This is all you need for a user to communicate with a server on the same VM system.

You can also communicate with other systems that support APPC. For example:

- ❑ You can communicate between VM/ESA systems that are in either the same Transparent Services Access Facility (TSAF) collection, or in the same Communication Services (CS) collection.

- ❑ You can communicate with systems in an SNA network if you have installed Advanced Communication Facility for the Virtual Telecommunications Access Method (ACF/VTAM), Group Control System (GCS), and APPC/VM VTAM Support (AVS).

For more information on available connections, refer to the IBM publications IBM VM/SP Release 6 Connectivity and Programming Guide and Reference (SC24-5377), or VM/ESA Connectivity Planning, Administration, and Operation (SC24-5448).

Configuration for a Server

Step 1: Specify the SAS System option **APPCSEC=_SECURE_**.

Note: This task is required.

Specify the SAS System option `APPCSEC=_SECURE_` to cause the APPC access method to require users to supply a valid userid and password (for the system where the server is running) in order to connect to the server.

Step 2: Modify the VM Directory Entry

Note: This task is required.

You must include the following statements in the VM directory entry for the server's virtual machine:

```
IUCV *IDENT serverid GLOBAL
```

specifies the name of the server that will run in the virtual machine. The server name is specified via the `SERVERID=` option of the `SERVER` procedure in the SAS program that creates the server. The serverid is typically the name of the server's virtual machine (the VM userid). If you specify `serverid` as `RESANY`, any valid server name can be specified for the `SERVERID=` option.

```
IUCV ALLOW
```

allows users to establish IUCV connections to the server's virtual machine.

```
OPTION MAXCONN 1024
```

specifies the maximum number of simultaneous connections to the server. As a rule of thumb, you should allow for 4-5 connections for each user who accesses data through the server. The number shown here is only a guideline; the default `MAXCONN` value is `4` and the maximum `MAXCONN` value is `65535`. SAS/SHARE software does not specifically limit the number of simultaneous connections to a SAS server.

Configuration for Users

Step 1: Modify the VM Directory Entry

Note: This task is required.

You must include the following statement in the VM directory entry for each user who will access a server:

```
OPTION MAXCONN 128
```

specifies the maximum number of simultaneous connections to servers. As a rule of thumb, you should allow for 4-5 connections for each server the user will access simultaneously. The number shown here is only a guideline; the default `MAXCONN` value is `4` and the maximum `MAXCONN` value is `65535`.

Step 2: Create a System-Level Communications Directory File

Note: This task is required.

You must create a system communications directory file with an entry for each server that your users will access. The system communications directory file is named `SCOMDIR NAMES` by default and should reside on a system minidisk common to all CMS users. An entry contains information required for communicating with a server. An entry for a server on the same CMS system or within the same TSAF collection is of the form:

```
:NICK.serverid    :LUNAME.*IDENT
                  :TPN.serverid
                  :SECURITY.type
```

where

`serverid`

is the serverid as specified in the `IUCV *IDENT VM` directory entry record for the server's virtual machine.

`security`

is either `PGM` or `NONE`. Specify `PGM` if the server is running secured (requiring a userid and password for the system where it is running from each connecting user). Specify `NONE` if the server is running unsecured (not requiring a userid/password).

An entry for a server outside the TSAF collection is of the form:

```
:NICK.serverid    :LUNAME.gateway serverid
                  :TPN.SASTP62
                  :MODENAME.modename
                  :SECURITY.type
```

where

`serverid`

is the name of the LU for the server.

`gateway`

is the global gateway to the SNA network.

`modename`

is the communications mode for the SNA session connecting the gateway to the LU defined for the server.

`security`

is either `PGM` or `NONE`. Specify `PGM` if the server is running secured (requiring a userid and password for the system where it is running from each connecting user). Specify `NONE` if the server is running unsecured (not requiring a userid/password).

Refer to the IBM publication *VM/ESA Connectivity Planning, Administration, and Operation* (SC24-5448) for more information on the creation and processing of communications directories.

Step 3: Specify Userid and Password for the Server's System

Note: This step may be required.

Each user who will connect to a server that is running secured must specify a userid and password that is valid on the system where the server is running. A server that is running secured requires a userid and password from each user, which it validates on the system where it is running.

You can specify a user's userid and password for the server's system in any of the following ways:

- ❑ an `APPCPASS` statement in the user's VM directory entry
- ❑ a user communications directory file on the user's A-disk
- ❑ the SAS System option
  `APPCSEC=userid.password|password|_PROMPT`

Specifying an APPCPASS VM Directory Statement

You can specify a user's userid and password for the server's system in an APPCPASS statement in the user's VM directory entry. This statement is of the form:

```
APPCPASS luname userid password
```

where

`luname`

specifies the gateway and serverid (as in the :LUNAME specification in a communications directory entry) as two 8-byte, blank-separated names. Each name should be padded on the right with blanks if necessary.

`userid`

specifies a user's userid for the system where the server is running

`password`

specifies the password for the userid

This is a relatively secure method because a user's VM directory entry is only accessible to the user and the system administrators. But this method may also require more maintenance by your system administrators because the APPCPASS statement must be created initially for a user by an administrator and then modified by the administrator if the user changes his password for the server's system subsequently.

Creating a User Communications Directory File

You can instead specify a user's userid and password for the server's system in a user communications directory file on the user's A-disk. The directory is named UCOMDIR NAMES by default and should contain an entry for each secured server the user will connect to. An entry in a user communications directory file has the same format as one in a system communications directory file with the addition of these two fields:

```
:USERID.userid
:PASSWORD.password
```

This method is less secure than an APPCPASS statement because any other user who can read UCOMDIR NAMES can thus obtain the user's userid and password for the server's system. You can limit this exposure by limiting access to the file and/or the minidisk or SFS directory in which it resides (for example, by making the file have a filemode number of 0, or putting a read password on the minidisk where the file resides). This method may require less involvement of system administrators because a user can modify his own communications directory if he changes his password for the server's system subsequently. But it requires users to maintain other fields like serverid, gateway, and modename.

Refer to the IBM publication *VM/ESA Connectivity Planning, Administration, and Operation* (SC24-5448) for more information on the creation and processing of communications directories.

Specifying the APPCSEC SAS System Option

Yet another way to specify a user's userid and password for the server's system is for the user to run SAS with the SAS System option,
`APPCSEC=userid.password|password|_PROMPT_|_NONE_`

where:

`userid.password`

specifies both the userid and the password.

`password`

specifies just the password, and uses the local userid as the default for `userid` (this is different than `_NONE_`).

`_PROMPT_`

causes the SAS System to prompt the user for userid and password information when necessary. When prompted for userid, if you press Enter without supplying one, then the local userid is used as the default for `userid` (this is different than `_NONE_`). When prompted for a password, if you press Enter without supplying one, then the password is obtained from `UCOMDIR NAMES`, `SCOMDIR NAMES`, or an `APPCPASS CP` directory statement.

`_NONE_`

causes the userid and password information to be obtained from `UCOMDIR NAMES`, `SCOMDIR NAMES`, or an `APPCPASS CP` directory statement. This is the default.

The options `userid.password`, `password`, and `_PROMPT_` are new for this release of the SAS System for CMS. They can be used as the primary method of providing userid and/or password information, or to override the userid and/or password information in `UCOMDIR NAMES`, `SCOMDIR NAMES`, or `APPCPASS CP` directory entries.

How secure this method is depends on how it is used. If userid.password or password is specified, then the information can appear in the SASLOG. If userid.password or password is specified noninteractively, then the file containing the option needs to be secure (for example, filemode 0, or a read password on the disk). If `_PROMPT_` is specified, the information does not show up in a SASLOG.

This method is easy to maintain because the user simply enters different information, or updates the file where they originally entered it. This method, like the user communications directory file method, has the advantage of not requiring the involvement of system administrators. It also shares an advantage of the APPCPASS VM directory statement method in that it allows the system-wide communications directory file (SCOMDIR NAMES) to be more fully utilized. This is because each user doesn't need to have a UCOMDIR NAMES file for the sole purpose of containing secure userid and password information.

Note: One way to mix and match the above alternatives is to keep the information in an APPCPASS CP directory entry, and use the APPCSEC=_PROMPT_ SAS System option to override it when there are changes (at least until the system administrator can update APPCPASS).

The handling of userid and password information for APPC at your site may involve both system administrators and users.

System Configuration for the TCP/IP Access Method

Software Requirements

The SAS TCP/IP access method for CMS relies on an underlying layer of TCP/IP communication software such as IBM TCP/IP Version 2 for CMS. TCP/IP communications software handles the actual communications.

To use the TCP/IP access method for CMS at your site, you must have IBM TCP/IP Release 2.0 or higher.

System Configuration for TCP/IP

Specify the SAS System option **TCPSEC=_SECURE_** for the server execution

Note: This task is required.

Specify the SAS System option TCPSEC=_SECURE_ to cause the TCP/IP access method to require users to supply a valid userid and password for the CMS operating system where the server is running in order to connect to the server.

Install a userid/password verification user exit

Note: This task is required.

When a user attempts to connect via the TCP/IP access method to a server running with the SAS System option TCPSEC=_SECURE_, the access method will verify that the user has provided a valid userid/password pair before allowing the user to connect. The userid/password verification user exit is required by the access method to interface with your site's external security software.

The file SASUSER ASSEMBLE on the installation tape contains two sample routines for this exit (vector slot number 62). One interfaces with Sterling Software's (formerly Systems Center) VMSECURE security product. To activate this routine, change slot 62 from A(0) to A(VMSCPW). The other routine interfaces with IBM's RACF/VM security product. To activate this routine, change slot 62 from A(0) to A(RACFPW). See the Appendix J, "User Exits in the SAS System" for information on installing user exits.

Specify Server Names and Ports

Note: This task is required.

To configure your installation for the TCP/IP access method, specify SAS/SHARE server names and port assignments. The server names and port assignments for the TCP/IP access method are defined in the services file (see the section *Network Configuration Data for TCP/IP* for more information on this file). A server name must be 1 to 8 characters in length. The first character must be a letter or underscore. The remaining seven characters can include letters, digits, underscores, the dollar sign ($) or the at sign (@).

Make the name of the TCP service machine available to the access method for IBM TCP/IP

Note: This task is required.

For IBM TCP/IP under CMS, the access method must locate a TCP/IP virtual machine whose name may vary from site to site. The TCP/IP access method uses the TCPIP_MACH environment variable to determine the value of this name. If the TCPIP_MACH variable does not exist, the access method searches for the virtual machine name in the TCPIP DATA file. If this file is not available, the access method uses a default value of TCPIP to locate the TCP/IP virtual machine.

Ensure TCP/IP configuration data is accessible

Note: This task is required if not previously accomplished.

The TCP/IP access method requires that certain data sets containing configuration data be available to the SAS System. The following section, *Network Configuration Data*, lists each data set and how the SAS System searches for each one. One of the methods below must be used to inform SAS System of the name of each configuration data set.

Most sites already have these data sets available for existing TCP/IP applications. Furthermore, if your site already has TCP/IP applications running under the SAS/C library, this step was probably completed when it was installed.

Method 1: Use the default UNIX-derived naming convention

The easiest way to configure your system is to locate files that have names derived from UNIX file names, such as ETC HOSTS, on a minidisk that is accessible to all TCP/IP users.

Method 2: Use environmental variables for custom naming

If you cannot use the default naming convention, set configuration environmental variables (see the section *Network Configuration Data*) to the names of your TCP/IP configuration files with the `GLOBALV` command. The syntax of the value is `CMS:<filename> <filetype> <filemode>.` For example, to relay the name of the `ETC SERVICES file`, `TCPIP SERVICES T`, set the environmental variable `ETC_SERVICES` to `CMS:TCPIP SERVICES T`.

If utilized, these environmental variables must be set before each SAS session that utilizes the TCP/IP access method completes initialization. Of course the files that the variables reference must be accessible to each TCP/IP user.

Create an ETC RESOLV file for name resolution

Note: This task is optional.

If you want to use the resolver for name resolution, create an `ETC RESOLV` file. See the subsection below, *Finding Host Names and Addresses* for more information about name resolution.

IBM TCP/IP sites do not need this file. They can use the pre-existing `TCPIP DATA` file to supply resolver configuration.

Network Configuration Data for TCP/IP

The operation of the TCP/IP access method depends on its ability to access the configuration information for a site. This section discusses the location of site configuration files and provides a detailed explanation of how the TCP/IP access method finds these files.

You should have already created the relevant files this section discusses when you installed your communications software, such as IBM TCP/IP.

Configuration data sets

On a UNIX operating system, the following five data sets usually contain site-dependent configuration information for TCP/IP.

- /etc/hosts
- /etc/networks
- /etc/services
- /etc/protocols
- /etc/resolv.conf.

The TCP/IP access method uses equivalent data sets under CMS. However this access method does not utilize the /etc/networks or /etc/protocols files, they are listed here for sake of completeness.

The CMS file system differs from the UNIX file system, and local security or organization considerations can affect how data sets are named. Although the name of the data set may be different, the data set that contains site configuration information is in the same format as the equivalent data set in a UNIX operating system.

Search logic

Under CMS, the data set that contains configuration information usually has a name that is derived from the equivalent UNIX filename. For example, the data set name `ETC HOSTS` is derived from the UNIX filename /etc/hosts.

The access method uses the following methods to look for each of the configuration data sets.

/etc/services

The access method looks for the following data set names while searching for the CMS data set that is equivalent to /etc/services:

1. value of `ETC_SERVICES` environment variable, if defined.
2. `ETC SERVICES`.

/etc/hosts

The access method looks for the following data set names while searching for the CMS data set that is equivalent to /etc/hosts:

1. value of `ETC_HOSTS` environment variable, if defined.
2. `ETC HOSTS`.

/etc/resolv.conf

The access method looks for the following data set names while searching for the CMS data set that is equivalent to /etc/resolv.conf:

1. value of `ETC_RESOLV_CONF` environment variable, if defined
2. `ETC RESOLV`

When the access method finds a data set with one of the above names, the name is retained for the duration of the SAS execution. You may need to restart SAS to cause it to find a different filename.

Finding host names and addresses

In the UNIX environment, programs that need host names and addresses may use the /etc/hosts file, or they may call the resolver to contact the name server for the host name information.

If you are running a name server, the name server addresses are determined from the /etc/resolv.conf file. If this file does not exist, the `TCPIP DATA` file is used instead to determine resolver configuration and IBM defaults. The TCP/IP access method only recognizes the first three name servers specified in this file. Both the UNIX operating system and the access method have a limit of three name servers.

Note that the access method does not support the IBM TCPIP files, `HOSTS LOCAL` and `HOSTS SITEINFO`. Sites that use these files must either switch to using name servers or create an equivalent '/etc/hosts' file.

The TCP/IP access method uses the following logic when looking up host names and addresses:

The default logic uses the following algorithm:

1. It looks for the /etc/resolv.conf file, using the rules listed above. If it does not find this file, it goes to step 2. Otherwise, if the "nameserver" statement in this data set specifies the use of the resolver and name server, it performs the requested queries and returns any answer it receives. If attempts to connect to the nameservers are refused, it goes to step 3.

2. It looks for a data set in the format of the IBM TCP/IP file with name `TCPIP DATA`. If the `NSINTERADDR` statement in this data set specifies the use of the resolver and name server, the access method performs the specified queries. It returns any answer it receives. If attempts to connect to the nameservers are refused, it goes to step 3.

3. It looks for an /etc/hosts file, using the rules listed above. If it finds an /etc/hosts file, it returns the result, including failure.

The determination of the domain name in name server queries follows the same logic as the UNIX operating system in using the "domain" statement of the /etc/resolv.conf file or the `DOMAINORIGIN` statement of `TCPIP DATA`, the file specified by the `HOSTALIASES` environment variable, and the value of the `LOCALDOMAIN` environment variable.

Formatting the Server's Minidisks

At some sites, minidisks are formatted by the VM system administrator when a new virtual machine is defined. However, if you must format the server's minidisk(s), you should specify a `BLKSIZE` option in the CMS FORMAT command of 4K or 4096. This allows the maximum amount of data to be transferred in each I/O operation, which is most efficient for the SAS server.

Appendix H, Implementing SAS/TOOLKIT® Software

The SAS/TOOLKIT software product for the VM/CMS system is divided into the following parts:

- ❏ Global material
- ❏ SAS/C material
- ❏ PL/I material
- ❏ IBM Assembly Language material.

Regardless of the language of interest, you should install the global material. You can then install one or more of the sections pertaining to the specific languages.

Global Material

The global material consists of the following:

- ❏ Grammars
- ❏ Test SAS programs
- ❏ Global TEXT
- ❏ Special EXECs.

The *grammars* are sample grammar files that are associated with the sample procedures supplied with various language materials. This maclib should remain intact, but you will also want to have copies of each individual member of the maclib. To copy the maclibs, issue the following:

```
FILEDEF INMOVE DISK GRMMAC MACLIB *
FILEDEF OUTMOVE DISK X GRM
MOVEFILE (PDS
```

This creates separate files, all with the filetype of `GRM`, for the members of `GRMMAC`. Do not delete the `GRMMAC MACLIB` file, since it is accessed by the same EXECs shipped with the separate language material.

The *test SAS programs* are also shipped in a maclib, `SAMPTOOL MACLIB`. It is not necessary to break out the individual members of this maclib, since the SAS System will read directly from a maclib member.

The *global TEXT* is located in the file `TOOLKIT TXTLIB`. This should not be altered. This is used to resolve external references for SAS/TOOLKIT software applications when linking.

The *special EXECs* do not pertain to a specific example, but may pertain to a specific language:

`LKEDPV6 EXEC`	This EXEC is responsible for linking the SAS/TOOLKIT modules, handling the language environments differently. You should examine this EXEC to see if your site has special needs in this area.
`FTNPREP EXEC`	This EXEC is responsible for calling the SAS program `FTNPREP SAS` to preprocess a SAS/TOOLKIT FORTRAN source file. This EXEC does not need to be modified by your site unless your FORTRAN command is different than `FORTVS2`.
`UWC EXEC` `UWCF EXEC`	These EXECs invoke the SAS/C, FORTRAN, IBM Assembler, and PL/I compilers, respectively. Examine them to see if you need to make changes pertinent to your local site.

SAS/C Material

The SAS/C material consists of the following:

- ❑ Sample programs
- ❑ TEXT files
- ❑ #include files
- ❑ executable modules
- ❑ EXEC files.

The *sample programs* are C programs that are sample procedures, formats, informats, functions, and engines. These are found in the `SASCSMAC MACLIB` file. If you want to test the compiler with the source code, you will need to break out the members into individual files:

```
FILEDEF INMOVE DISK SASCSMAC MACLIB *
FILEDEF OUTMOVE DISK X C
MOVEFILE (PDS
```

This creates separate files, all with the filetype of C, for the members of `SASCSMAC MACLIB`.

The *TEXT files* are located in the `SASCTXT TXTLIB` file. If you want to test the `LKED` command against the members, they should be broken out into individual TEXT files:

```
FILEDEF INMOVE DISK SASCTXT TXTLIB *
FILEDEF OUTMOVE DISK X TEXT
MOVEFILE (PDS
```

The *#include files* are located in `SASCMAC MACLIB`. This maclib should remain as is, since it is used by the compiler as a maclib.

The *executable modules* reside in the SASCLOAD LOADLIB file. They should remain in the LOADLIB to be accessed by the SAS System. To test the sample executables, issue the following command:

```
SAS (SASLOAD='SASCLOAD LOADLIB *')
```

The *EXEC files* are loose EXEC files. These EXECs describe how to compile and link sample SAS/TOOLKIT software applications using the SAS/C Compiler.

PL/I Material

The PL/I material consists of the following:

- ❑ sample programs
- ❑ TEXT files
- ❑ #include files
- ❑ executable modules
- ❑ EXEC files.

The *sample programs* are PL/I programs that are sample procedures, formats, informats, functions, and engines. These are found in the PLISMAC MACLIB file. If you want to test the compiler with the source code, you will need to break out the members into individual files:

```
FILEDEF INMOVE DISK PLISMAC MACLIB *
FILEDEF OUTMOVE DISK X PLI
MOVEFILE (PDS
```

This creates separate files, all with the filetype of PLI, for the members of PLISMAC MACLIB.

The *TEXT files* are located in the PLITXT TXTLIB file. If you want to test the LKED command against the members, they should be broken out into individual TEXT files:

```
FILEDEF INMOVE DISK PLITXT TXTLIB *
FILEDEF OUTMOVE DISK X TEXT
MOVEFILE (PDS
```

The *#include files* are located in PLIMAC MACLIB. This maclib should remain as is, since it is used by the compiler as a maclib.

The *executable modules* reside in the PLILOAD LOADLIB file. They should remain in the LOADLIB to be accessed by the SAS System. To test the sample executables, issue the following command:

```
GLOBAL TXTLIB PLILIB
SAS (SASLOAD='PLILOAD LOADLIB *')
```

The *EXEC files* are loose EXEC files. These EXECs describe how to compile and link sample SAS/TOOLKIT software applications using the PL/I Compiler.

FORTRAN Material

The FORTRAN material consists of the following:

- ❑ sample programs
- ❑ TEXT files
- ❑ #include files
- ❑ executable modules
- ❑ EXEC files.

The *sample programs* are FORTRAN programs that are sample procedures, formats, informats, functions, and engines. These are found in the `FTNSMAC MACLIB` file. If you want to test the compiler with the source code, you will need to break out the members into individual files:

```
FILEDEF INMOVE DISK FTNSMAC MACLIB *
FILEDEF OUTMOVE DISK X FORTRAN
MOVEFILE (PDS
```

This creates separate files, all with the filetype of FORTRAN, for the members of `FTNSMAC MACLIB`.

The *TEXT files* are located in the `FTNTXT TXTLIB` file. If you want to test the `LKED` command against the members, they should be broken out into individual TEXT files:

```
FILEDEF INMOVE DISK FTNTXT TXTLIB *
FILEDEF OUTMOVE DISK X TEXT
MOVEFILE (PDS
```

The *#include files* are located in `FTNMAC MACLIB`. This maclib should remain as is, since it is used by the compiler as a maclib.

The *executable modules* reside in the `FTNLOAD LOADLIB` file. They should remain in the `LOADLIB` to be accessed by the SAS System. To test the sample executables, issue the following commands:

```
GLOBAL LOADLIB VSF2LOAD
SAS (SASLOAD='FTNLOAD LOADLIB *')
```

The *EXEC files* are loose EXEC files. These EXECs describe how to compile and link sample SAS/TOOLKIT software applications using the FORTRAN compiler.

IBM ASSEMBLY Language Material

The IBM Assembly language material consists of the following:

- ❑ sample programs
- ❑ TEXT files
- ❑ #include files
- ❑ executable modules
- ❑ EXEC files.

The *sample programs* are assembler programs that are sample procedures, formats, informats, functions, and engines. These are found in the `ASMSMAC MACLIB` file. If you want to test the assembler with the source code, you will need to break out the members into individual files:

```
FILEDEF INMOVE DISK ASMSMAC MACLIB *
FILEDEF OUTMOVE DISK X ASSEMBLE
MOVEFILE (PDS
```

This creates separate files, all with the filetype of `ASSEMBLE`, for the members of `ASMSMAC MACLIB`.

The *TEXT files* are located in the `ASMTXT TXTLIB` file. If you want to test the `LKED` command against the members, they should be broken out into individual TEXT files:

```
FILEDEF INMOVE DISK ASMTXT TXTLIB *
FILEDEF OUTMOVE DISK X TEXT
MOVEFILE (PDS
```

The *#include files* are located in `ASMMAC MACLIB`. This maclib should remain as is, since it is used by the assembler as a maclib.

The *executable modules* reside in the `ASMLOAD LOADLIB` file. They should remain in the `LOADLIB` to be accessed by the SAS System. To test the sample executables, issue the following command:

```
SAS (SASLOAD='ASMLOAD LOADLIB *')
```

The *EXEC files* are loose EXEC files. These EXECs describe how to compile and link sample SAS/TOOLKIT software applications using the IBM Assembler.

Appendix I, Installing the SAS/TUTOR® Courses

Verifying Access for Users

1. It is recommended that you install SAS/TUTOR courses on a minidisk, not a shared file system directory, separate from the disk where the rest of the SAS System is installed.

2. Assign both a read and write password to a second minidisk to be used as a SAS/TUTOR control library. This must be a separate minidisk from the one where the SAS/TUTOR courses are installed.

3. If you employ password suppression, it is recommended that you assign identical passwords for both disks to simplify initialization.

Initializing the SAS/TUTOR System

1. Edit the SASTUTOR EXEC to specify:

   - ❏ the minidisk containing the installed SAS/TUTOR courses (`0MAIN SASCBTA`)

   - ❏ the SAS EXEC used to invoke SAS System Release 6.09 Enhanced.

2. Release and detach both the SAS/TUTOR and the control minidisks.

3. Issue the `SASTUTOR` command at the system prompt.

4. On the Course Management System initialization screen, enter the password:

   ```
   SAS$INIT
   ```

5. When prompted, enter all requested information required to link and access the SAS/TUTOR control library referenced in Step 2 of the section *Verifying Access for Users*.

   If your site enables Password Suppression on CMS (which requires that passwords be entered as a separate argument from the `LINK` command) then enter `ALL` when prompted for all passwords during the initialization.

6. When prompted, enter requested information to temporarily access the minidisk containing `0MAIN SASCBTA`. During this step, your userid is stored, allowing you future access to the Course Management System as a course administrator.

7. From the Course Management System Main Menu, select the option `Initialize a Course`.

8. Enter the `ID` of each course you installed (for example, `CBT 101`).

9. When prompted, enter all requested information required to locate the course files.

10. Once all courses have been initialized, exit from the Course Management System Main Menu.

Verifying Installation and Initialization of SAS/TUTOR Courses

1. At the operating system prompt, issue the command:

   ```
   SASTUTOR
   ```

   If the SAS/TUTOR system is properly installed and initialized, you are taken to the Course Management System Main Menu because you are listed as a course administrator.

2. Select the option `Take a Course`.

3. If courses are properly initialized, a menu of installed courses is displayed.

Adding a Course Administrator to the SAS/TUTOR System

1. At the operating system prompt, issue the `SASTUTOR` command.

2. From the Course Management System Main Menu, select `Update the Course Administrator List`.

3. Enter the userid of the additional course administrator(s). Up to eight userids can be entered.

   Note: Once a userid is listed on the course administrator list, that individual is automatically taken to the Course Management System Main Menu each time he or she issues the `SASTUTOR` command.

   Each course administrator has authority to do the following:

   - ❑ add or remove a course from the SAS/TUTOR Course Selection Menu
   - ❑ activate, modify, and use the student tracking system
   - ❑ add or remove any user id from the course administrator list.

Restoring Tracking data to Release 6.09 Enhanced (optional)

If tracking was turned on for your site for SAS System Release 6.07 or 6.08, you can restore your tracking data by following these steps at any time after installing and initializing the SAS/TUTOR system.

1. Link and access the SAS/TUTOR control minidisk for your prior release.

2. Link and access the new SAS/TUTOR control minidisk for Release 6.09 Enhanced.

3. Invoke Release 6.09 Enhanced of the SAS System.

4. Assign librefs to the SAS/TUTOR control libraries for both your prior release and Release 6.09 Enhanced as follows:

```
libname OLD 'SASCBT0 fm1';   /* filemode for old control minidisk  */

libname NEW 'SASCBT0 fm2';   /* filemode for 6.09E control minidisk  */
```

5. Submit the following SAS program to append Release 6.07 or 6.08 tracking data to the Release 6.09 Enhanced tracking data set:

```
proc append base=NEW.TRACK data=OLD.TRACK;
proc sort data=NEW.TRACK nodups;
       by TRKVAR;
run;
```

Deleting SAS/TUTOR Courses from Disk

If you have received a SAS/TUTOR course on a trial basis and want to delete the course after examination, use the `DELTUTOR EXEC`.

For example:

```
DELTUTOR
```

displays a list of all SAS/TUTOR courses installed at your site. You will then be prompted to select the course to delete.

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

Using User Exits

If you need to monitor or control execution of the SAS System, you should review this information. You might, for example, want to write an exit to send accounting records at the beginning and end of each SAS session program. Information in this section will describe how to do this.

The `SASUSER` assembler module can be modified to activate any of the exits provided. Calls from `SASUSER` to other language environments is possible, but it is the responsibility of the user to establish and terminate the environment, as well as provide the necessary linkage to transfer control to and from the SAS System environment.

A limited number of exits are activated. The `WEHEADER DSECT`, found in the `UXDSECT` member of the `SAMPEXIT MACLIB`, provides a glimpse of longer range possibilities. We welcome input on functionality and control information that you feel is necessary. Many of the SAS/IO and external file exits have not been implemented pending alternative user-written `ENGINEs` and `ACCESS` methods.

Note: Not all exits available in Release 5.18 are available in Version 6 of the SAS software. The form of the exits may change in subsequent releases of the SAS System.

User Exits Available under CMS

The following user exits 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 considered tasks. There are also other tasks for functionality.

- Task Termination

This exit is called at termination of each task in the SAS System.

- ❑ External file assignment

  This exit is called when a fileref is created for a physical file. The exit can be used to modify parameters associated with the file or to control additional printing information.

- ❑ External file close

  This exit is called when a physical file is closed. The exit can be used to post-process a file or alter the routing information of a printer file.

- ❑ SAS/SHARE Library authorization

  This exit is called only from `SAS/SHARE PROC SERVER` for user authorization to access a SAS Library.

- ❑ SAS/SHARE Member authorization

  This exit is called only from `SAS/SHARE PROC SERVER` for user authorization to access a SAS Library member.

- ❑ SAS/SHARE TCP/IP userid/password verification.

Restrictions and Caveats

1. The contents of `SASUSER ASSEMBLE` will change in future releases of the SAS System.

2. For equivalent functionality of the 5.18 step initialization and termination exits, you must test for a task type of `PROC`.

3. Every module in the SAS System saved segments must be reentrant and reusable. For this reason, any enhancements that you make to `SASUSER ASSEMBLE` must, when assembled, produce a module that is reentrant and reusable if you want to save that module in segments.

Overview of the SASUSER Module

The user exits are defined and implemented in the `SASUSER` module. The `SASUSER` module is made available early in the initialization of the SAS System and remains resident during execution of SAS programs.

When you modify the `SASUSER` module, the SASUSER CSECT must remain at the entry point. When the `SASUSER` module is invoked, it must return the address of the Exits Table in register 15.

To activate an exit, place an entry in the appropriate slot in the Exits Table in `SASUSER`. When `SASUSER` is assembled, the entry must resolve to a function address. A function is a section of executable instructions and data in an object module. At predetermined points during execution, the SAS System inspects the

entries in the Exits Table. Control is passed to the function when the entry is non-zero.

Although you can activate all exits individually by providing a different address in each slot in the Exits Table, an alternative approach is available. You can consolidate functions by using the same address in each slot. A function that services multiple exits can use the `CODE` value defined in the WEHEADER DSECT to determine the invoking exit. Control blocks for all available exits are contained in the UXDSECT copy file in SAMPEXIT MACLIB.

Summary

The `SASUSER` module is loaded during initialization of the SAS System and remains resident during execution of the SAS System. At predetermined points during execution, the SAS System indexes to appropriate entries in the Exits Table. If the entry is non-zero, control is passed to that address. By editing the SASUSER ASSEMBLE file, you can define and implement functions that are called by the exits.

The following sections explain how to make the user exits available and how to develop functions that are invoked by the user exits. The installation tape contains a SASUSER ASSEMBLE file and the necessary SAMPEXIT MACLIB in the samples section.

How to Create the **`SASUSER LOADLIB`**

To make the user exits available, modify the assembler file to implement the desired exits, global the necessary MACLIBs, assemble `SASUSER` using Assembler H or High Level Assembler and link the resulting text into a `LOADLIB` member.

The MACLIBS that you must global include SAMPEXIT, DMSGPI, and OSMACRO.

For example, under VM/ESA, issue the following commands:

```
GLOBAL MACLIB SAMPEXIT DMSGPI OSMACRO
HLASM SASUSER
LKED SASUSER ( MAP
```

Now that you have created the SASUSER LOADLIB, you are ready to test the user exits.

How to Test the User Exits

To test the user exits, you must make the SASUSER LOADLIB available to the SAS System. This is accomplished by invoking the SAS System with the SAS System options, `SASLOAD=` and `$NOSEGL`.

For example, issue the following command:

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

Note: The SAS System option `$NOSEGL ($ NO SEGment Load)` is an option that should only be used while testing your implementation of the user exits. This option causes the SAS System to use the specified member from a loadlib rather than from segments.

At this point, the user exits are only available when you invoke the SAS System with `SASLOAD='SASUSER LOADLIB *'` and `$NOSEGL=SASUSER`. If you want to make the user exits a permanent part of the SAS System available to others at your site, follow the instructions in the next section.

How to Make the User Exits a Permanent Part of the SAS System

In order to make the user exits a permanent part of the SAS System, you will have to modify the SASBASE LOADLIB and resave the SAS System segments.

Copy the `SASUSER` module from the SASUSER LOADLIB into your production `SASBASE LOADLIB` by linking to the SAS System minidisk with write access and issuing the following commands:

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

After typing in `SASUSER`, 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 a second copy of the SASBASE LOADLIB. To compress the loadlib, issue the following command:

```
LOADLIB COMPRESS SASBASE LOADLIB fm
```

Resaving the SAS System Saved Segments

Every module in the SAS System saved segments must be reentrant and reusable. Any enhancements that you make to SASUSER must produce a module that is reentrant and reusable if the module is saved in segments.

If you have saved the SAS System supervisor in the SAS System segments, then you must choose to either save SASUSER in, or remove SASUSER from the SAS System segments.

To implement your choice, edit either the SASSUPER or ALLSAS LLIST file, as appropriate. Locate the line:

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

If you want to remove SASUSER from the segments, place an asterisk in column 1 of that line. To save SASUSER in the segments, column 1 must be blank.

If you have the SAS supervisor saved in segments, you must rerun the CMSSEGS exec and resave the segments.

Now that you have created SASUSER LOADLIB, tested the user exits, and understand the steps needed to make the user exits a permanent part of the SAS System, you are prepared to develop and test functions that meet the particular needs of your site. In order to develop functions, you need to know how to use SAS System Dynamic Save Areas, SAS System data areas, and the SAMPEXIT MACLIB. The remainder of this appendix covers these topics.

Writing an EXIT

Any function that you write must adhere to one simple rule. When a function is called, it must save the state of the calling program and, when a function is finished, it must restore the state of the calling program prior to returning control to that program. In this context, the state of a program is the contents of the general purpose registers. A dynamic save area (DSA) is an area in memory where the values of these registers can be saved. The following section describes the steps you must take in order to use the DSAs provided by the SAS System.

Information About SAS System Dynamic Save Areas

Every function that is invoked with an exit is responsible for saving and restoring the state of the calling program. Although you can create your own dynamic save areas, the UXPRLOG and UXEPLOG macros that are in SAMPEXIT MACLIB already provide this service. The UXPRLOG macro is used at the beginning of your routine and it saves the state of the calling program. The UXEPLOG macro is used at the end of your routine. It restores the state of the calling program.

The following parameters apply to the use of `UXPRLOG` and `UXPRLOG`:

- Valid parameters for `UXPRLOG` are:

  ```
  BASE=reg, DSA=YES, or DSA=length
  ```

- Valid parameters for `UXEPLOG` are:

  ```
  RC=reg, DSA=YES, or DSA=length
  ```

- Every function must use a `UXPRLOG` and `UXEPLOG`.

- There is no requirement to specify parameters on either the `PROLOG` or `EPILOG`.

- The default `BASE` register is `5`.

- All registers are saved and restored.

- Register `13` points to the Dynamic Save Area (DSA).

- On entry to the function, register `1` points to the data described by the `WEHEADER DSECT`.

Variables may be added to the DSA by adding storage definitions at the end of the DSA DSECT defined by the UXDSA macro. By using an equate such as `DSALEN EQU *-DSA,` the symbol can be specified on `UXPRLOG` and `UXEPLOG` as `DSA=DSALEN` to allocate additional dynamic storage addressable off of register `13` with `USING DSA,R13.`

Information About SAS System Data Areas that are Available

You may want your function to extract information about the state of the virtual machine and currently executing program. The SAS System provides a global Data Area for all exits and additional exit-specific Data Areas where necessary. These DSECTs are found in UXDSECT in SAMPEXIT MACLIB.

Gathering Segment Accounting Information

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 exits allow you to invoke programs that you have written to record the names of each SAS System procedure used. However, this information alone may be insufficient to enable you to easily identify which modules in the saved segments are being used. Since it would be difficult for you to write your own set of programs that produce this segment accounting information, there is an alternative to using the exits for this purpose.

The SAS System has a segment accounting mechanism that is available using the SAS System option, `$SGSTAT`. When the SAS System is invoked with the `$SGSTAT` option, information concerning the use of modules in saved segments is sent to the SAS6ACT (SAS Version 6 Accounting) userid.

Note: Be forewarned, this technique will produce a great volume of information in a short period of time. Consequently, this method is intended to be used only long enough to gather sufficient information about the use of the modules that are in the SAS System's saved segments.

Appendix K, 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 in the sample section. To use this exec, enter:

```
UNMACLIB filename filetype filemode
```

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 `filemode` is the filemode of the minidisk on which the new files are to be written.

To use this exec to remove a single member of a MACLIB, enter the following:

```
UNMACLIB filename filetype filemode (MEMBER name)
```

where filename is the `filename` of the MACLIB, `filetype` is the filetype you want the loose file to be, filemode is the filemode of the minidisk on which the new file is to be written, and name is the single member name to be moved.

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

Appendix L, CMS Storage Dumps

When a CMS SAS System task abends, it generates a message of the form:

```
ERROR: Task terminated due to an illegal instruction.
```

or

```
ERROR: Task terminated due to a memory violation.
```

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. This line

```
>PROG .........
```

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

To send the storage dump on tape, issue the following command:

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

This puts an IPCS-format storage dump in the virtual reader. Your operations programmers can then put the storage dump on tape with the CP command `SPTAPE.`

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

Appendix M, Main/FRAMEs: Base SAS® Sample Graphical User Interface

Available with this release of the SAS System for CMS, Base SAS software now contains a sample application called Main/FRAMEs, which provides a graphical user interface to the VM/CMS operating system. Main/FRAMEs also enables user control of CMS sessions from any SAS supported platform without extensive knowledge of CMS commands.

To enable the underlying functionality of the sample GUI, two post installation options are available.

The first option uses the install procedure provided as part of the CMSGUI Catalog. Invoke a SAS session and issue the following command at the SAS Program Editor command line:

```
COPY SAMPSIO.CMSGUI.INSTALL.SOURCE
```

Read the source code thoroughly and make the recommended changes before issuing the SAS submit command.

The second option uses the following command with write access to the SAS System disk:

```
UNMACLIB SAMPGUI SASMACRO fm
```

where `fm` is the filemode of the SAS System disk. Either option will copy several `REXX SASMACRO` files from the `SAMPGUI MACLIB` to the SAS System disk.

The following naming convention is used to allow system, group, and personal versions of the Main/FRAMEs SASMACROs:

```
USRxxxxx
US1xxxxx
SASxxxxx
```

where `xxxxx` is the function passed as the first parameter (for example, FILE, DISK, PRINT). The supplied samples all use the `SASxxxxx` name.

If SAS Institute ships maintenance for any of these samples, the new version will overwrite the previous version.

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 Institute Inc.
CMS Systems Development
SAS Campus Drive
Cary, NC 27513-2414