Data General

Software Documentation

How to Generate and Run AOS

How to Generate and Run AOS

093-000217-06

For the latest enhancements, cautions, documentation changes, and other information on this product, please see the Release Notice (085-series) supplied with the software.

Ordering No. 093-000217

Printed in the United States of America
Revision 06, March 1985

How to
Generate and Run
AOS
093-000217

A vertical bar or asterisk in the margin of a page indicates substantive change or deletion, respectively, from the previous revision, except for Chapter 13, which is all new material. Chapter 8 is completely reorganized. Chapter 14 consists of the information previously contained in Chapter 13.

Revision History:	Effective with:
Original Release - April 1977	
First Revision - June 1978	
Second Revision - June 1979	
Third Revision - March 1980	
Fourth Revision - May 1982	
Addendum 086-000043-00 - July 1982	
Fifth Revision - October 1983	
Sixth Revision - March 1985	AOS Rev. 7.00

Preface

AOS, Data General's Advanced Operating System, can run on the following computers: ECLIPSE® Models C/150, C/330, C/350, S/20, S/120, S/130, S/140, S/230, S/250, S/280, M/600, AP/130, CS/100 Series, CS/200 Series A, B and C, MV/8000 ECLIPSE Model 9300 Series, and DG DESKTOP GENERATION™ Computer Models 10/SP, 20 and 30.

This manual tells you how to generate and run an AOS system on your computer.

Generating includes formatting one or more blank disks, installing a starter system, bringing it up, generating a tailored system, then creating the AOS multiuser environment. The whole generation procedure is usually done only once (often by a DG engineer); then parts of it may be repeated for new devices and for new revisions of AOS. Whatever your role, Chapters 1 through 5 lead you through the generation procedure.

Running includes day-to-day operation: bringing up the AOS system, shutting it down, making sure it runs smoothly for users, dumping files for safekeeping, and making decisions that help the system run most efficiently. Whatever your role, Chapters 6 through 14 and the appendixes describe the tools and concepts needed to run AOS.

Chapters 1 through 5 take a "cookbook" approach: they tell you what to do without detail. Chapters 6 through 14 are designed for reference; they describe the tools available, with details, and let you choose what you want.

The book is organized as follows:

Chapter 1	is an overview of system software, hardware, and the steps involved in generating and running an AOS system. It also describes keyboard control keys.
Chapter 2	tells how to bring up your first AOS system from magnetic tape. It assumes you are starting with blank disks. Read it when you are starting from scratch.
Chapter 3	tells how to bring up your first AOS system from diskettes. It assumes you are starting with blank disks. Read it when you are starting from scratch.
Chapter 4	tells how to run AOSGEN to generate a tailored AOS system. Read it whenever you want to generate or modify an AOS system.
Chapter 5	tells how to bring up the multiuser environment, which supports many timesharing and batch users concurrently. It leads you through creating user profiles, initializing the EXEC program, editing some macros to make things easier, and creating a tailored error message file. Read this chapter after you generate your first tailored system — later, you may want to read selected parts of it. This chapter ends the generating part of the book.
Chapter 6	details system startup, normal shutdown, and abnormal shutdown. You'll follow the steps described in this chapter quite often.
Chapter 7	explains the user profile editor, PREDITOR. You may want this information when you create, edit, or delete a user profile.

Chapter 8	details the EXEC multiuser management program: user logon, batch, spooling, user mount requests and labeled tapes, and the related EXEC commands. Because EXEC manages so many important multiuser functions, this is the biggest chapter in the book.
Chapter 9	explains runtime tools other than EXEC: the Command Line Interpreter (CLI), display utilities, confidence testing, system logging, and file backup. It includes the tab divider PED-SYSLOG/REPORT.
Chapter 10	tells how to handle unusual system conditions.
Chapter 11	details the Disk Formatter program, which formats physical disks into logical disk units for use with AOS.
Chapter 12	explains the Installer program, which installs an AOS system on a logical disk unit.
Chapter 13	offers some basic cautions and pointers: things to avoid and things to remember.
Chapter 14	outlines system management considerations: issues and decisions that can help make the system do what you bought it to do.
Appendix A	lists important error messages and recovery steps — from errors in all programs.
Appendix B	summarizes device names and codes.
Appendix C	is a fast-reference alphabetical summary of all EXEC commands.
Glossary	defines pertinent terms, like *process*. When you see a term you don't know, check the glossary.

For fast reference, insert and use the *tab dividers*.

What About Peripherals?

Before you can begin, at least one appropriate disk unit, tape unit, or diskette unit, and a system console (DASHER® terminal) must be connected to your computer, and all must have adequate power.

A DG engineer usually installs the hardware, so you need no information on this. In fact, a DG engineer often brings up the first system. But we include this material because someday you may want to do it.

Peripheral Operation and Manuals

Peripherals include disk units, tape units, diskette units, terminals (called consoles in this book), and line printers.

Operating disk units is not difficult; but if you need details, they are in *Disk Drives*, number 014-000099, and in the 015-series manual supplied with each disk unit.

Operating a tape unit is sketched in this book, and detailed in the Operator Reference Series book *Magnetic Tape Transports*, number 014-000095.

You will be using a DASHER terminal (called the system console) to communicate with the computer. This is easy; but if you need more information on the system console hardware, read the appropriate DASHER Operator Reference Series book.

A line printer isn't absolutely *required*, but if you have one, you will want to know how to turn it on, change the paper, and so on. The operation of line printers is covered in *Line Printers*, 014-000089.

For device status errors and other information, you may want the *Programmer's Reference Manual, Peripherals*, 014-000632.

Other Software Manuals

To create the multiuser environment, you will need to use a text editor. This book gives a brief sketch of DG's SED text editor, enough to start editing. SED is further described in *Learning to Use Your AOS System,* 069-000018, and in the *SED User's Manual,* 093-000249. If you like SED, you may want to look at these books later on.

The main program for communicating with an AOS system is its Command Line Interpreter (CLI). The CLI is AOS' command language. This book tells you about all the CLI commands you need; later, you will want to read about the CLI and its commands in the CLI manual, 093-000122 (full name *Command Line Interpreter (CLI) (AOS and AOS/VS) User's Manual.*)

For details on AOS memory management, you may want to read sections of the *AOS Programmer's Manual,* 093-000120.

Before you get you system up and running, if your computer is connected in a network with other computers, you should see *XODIAC™ Network Management System Guide for Managers and Operators*, 093-000260. And if you run the CEO® Comprehensive Electronic Office software on your system, you should see *Managing the CEO® System*, 093-000286.

Two reference cards should help you run the system. These are *AOS Startup and Shutdown Summary*, 069-000096, and *AOS Commands, Macros, and Programs*, 069-000097.

After you get your system and multiuser environment up and running, you will want to run other software, like compilers and data management products. These are described in books shipped with the software.

The Release Notice

The AOS Release Notice has the latest details on all AOS software; enhancements, new features, and improvements. The Release Notice is supplied both as a printed listing and as a disk file that you can print. The filename in directory :UTIL is RELEASE.n.nn, where n.nn is the revision number (for example, :UTIL:RELEASE.7.00).

You may want to read the Release Notice, or selected parts of it. If you want to know the features of an AOS release, or have problems with a release, check the notice for solutions. The Release Notice assumes that you know the operating system well — so parts of it may be difficult to understand until you *do* know the system.

Document-change files, also in :UTIL, are part of each release, but you must print these yourself after installing the new software. The document-change filenames have the form 0ss_nnnnnn_rr, where ss is the series, nnnnnn is the part number, and rr is the revision. For example, 093_000122_05 is the document-change file for the fifth revision of the *CLI User's Manual.* We suggest that — as you receive new software revisions from DG — you print the document-change file(s) and update the manual(s) as needed.

The Newsletter

Later on, you will find the *AOS Monthly Newsletter* a useful source of information on the latest enhancements to the AOS operating system.

Reader, Please Note:

We call terminals consoles, and extensions suffixes, in this manual.

We use these conventions for command formats in this manual:

COMMAND required *[optional]* ...

Where	Means
COMMAND	You must enter the command (or its accepted abbreviation) as shown.
required	You must enter some argument (such as a filename). Sometimes, we use: $\left\{ \begin{matrix} \text{required}_1 \\ \text{required}_2 \end{matrix} \right\}$ which means you must enter *one* of the arguments. Don't enter the braces; they only set off the choice.
[optional]	You have the option of entering this argument. Don't enter the brackets; they only set off what's optional.
...	You may repeat the preceding entry or entries. The explanation will tell you exactly what you may repeat.

Additionally, we use certain symbols in special ways:

Symbol	Means
↵	Press the NEW LINE on your console's keyboard. If there is no NEW LINE key, press the carriage return (RETURN) key.
□	Be sure to put a space here. (We use this only when we must; normally, you can see where to put spaces.)

All numbers are decimal, except for device codes and numbers marked octal. For example

...27 buffers...	means 27 decimal.
...device code 27...	means 27 octal.
...say 27 octal...	means 27 octal.

We show commands in UPPERCASE; but you can type them in lowercase, UPPERCASE, or any combination. Finally, in examples we use

THIS TYPEFACE TO SHOW YOUR ENTRY↵
THIS TYPEFACE FOR SYSTEM QUERIES AND RESPONSES.

) is the AOS operating system CLI prompt.

Contacting Data General

- If you have comments on this manual, please use the prepaid Remarks Form that appears after the Index. We want to know what you like and dislike about this manual.
- If you need additional manuals, please use the enclosed TIPS order form (USA only) or contact your Data General sales representative.

End of Preface

Contents

Chapter 4 - Generating a Tailored AOS System

Chapter 5 - Creating the Multiuser Environment

Chapter 6 - Startup and Shutdown

Chapter 7 - About the PREDITOR Profile Editor

Chapter 8 - EXEC and User Processes

Chapter 9 - Other Runtime Tools: CLI, Lock, PED, Display, Logging, and Backup

Appendix A - Important Errors and Error Messages

Appendix B - AOS Peripheral Devices

Appendix C - EXEC Command Summary

Appendix A - Important Errors and Error Messages

Appendix B - AOS Peripheral Devices

AOS Device and Unit Names B-1
AOS Magnetic Peripherals B-2
AOS Device Codes B-5

Appendix C - EXEC Command Summary

Glossary

Tables

Table

Illustrations

Figure

Chapter 1
About Generating and Running AOS

Read this chapter

- When you want to learn how to generate, and secondarily to run, an AOS system.

Generating your first system (Chapters 1 - 5) is one part of this. *Running* a system (Chapters 6 - 14) is another part. This chapter sketches both the *generating* and *running* parts. It assumes that AOS is new to you. The major sections proceed

- What is AOS?
- AOS File Structure
- What's Involved in System Generation?
- What's Involved in Running the System?
- Cautions and Control Characters
- If You Make a Mistake
- What Next?

What is AOS?

AOS is a general-purpose, multiprogramming, multitasking, operating system. You can use it to support users on a time-sharing basis, to run batch jobs, or both. You communicate with AOS by typing Command Line Interpreter (CLI) commands on your computer's system console.

Figure 1-1 shows the hardware in a typical small ECLIPSE installation. The names MTC0, and DPF0 are AOS device names. There can be, and often are, additional tape and disk units.

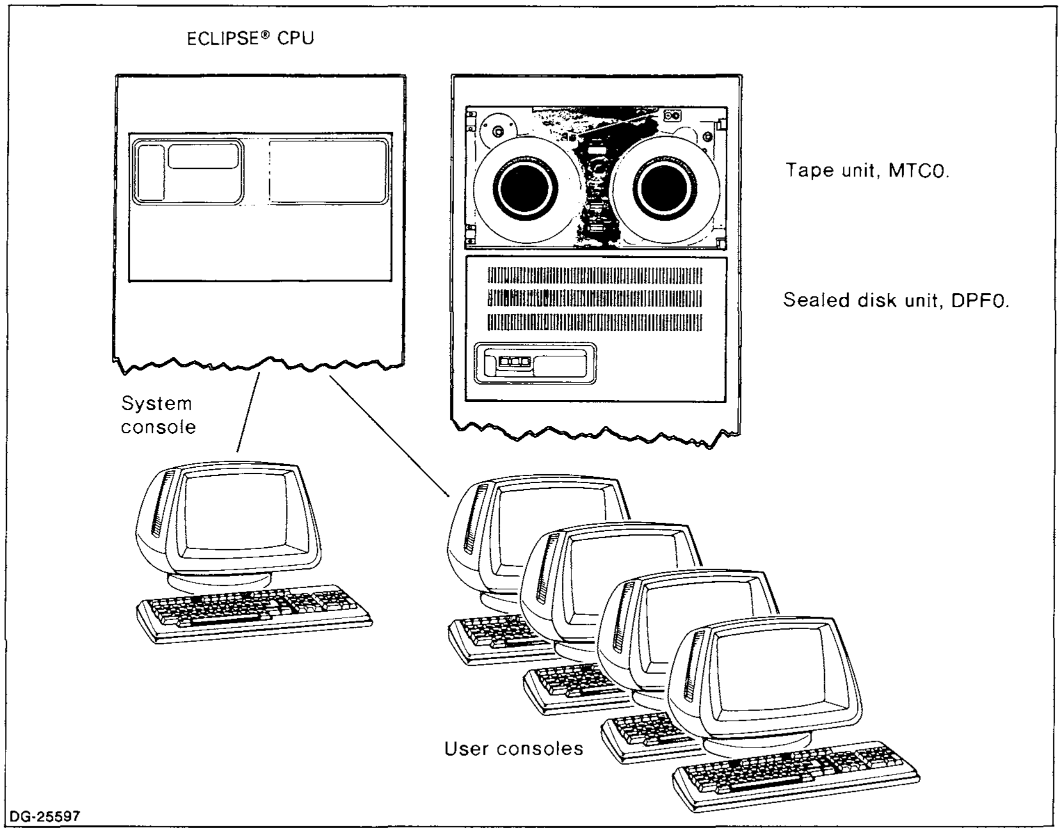


Figure 1-1. Sample AOS ECLIPSE® System Hardware Configuration

AOS File Structure

AOS manages many parts of its file structure, but you need to understand the options that are under your control. A typical AOS file system looks something like Figure 1-2.

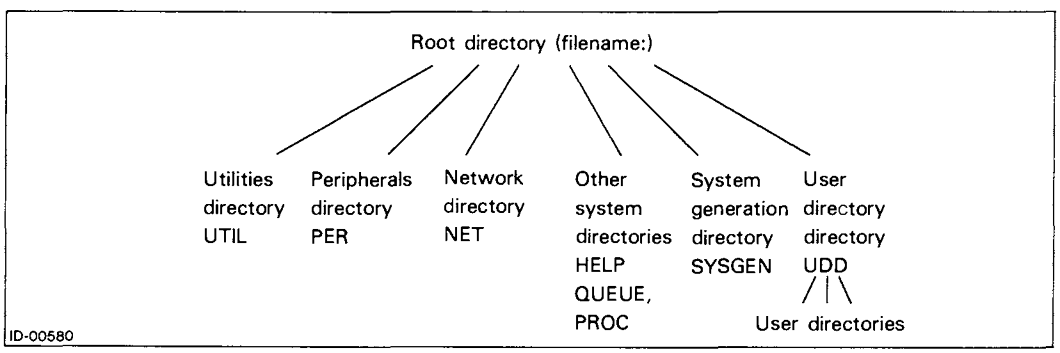


Figure 1-2. AOS System File Structure

The root directory (:) and other system directories are created and managed by AOS or its utility programs. The AOS system program file is usually in directory :SYSGEN. It can be copied (installed) onto an "invisible" part of the disk, dissociated from the directory structure.

Directory :UDD has an entry for each timesharing *user* directory. A user is an authorized person who can execute other programs — often application programs. So UDD often has many subordinate directories and uses a lot of storage space.

All these directories, and their subordinate directories, can reside on one logical disk unit *(LDU)*. Or one or more directories can reside on *its own* LDU. An LDU is one or more physical disks, processed by the Disk Formatter.

Generally, for the most versatile system, you will make each physical disk a single-disk LDU. The system disk is nearly always a single-disk LDU. (If you set up a multiple-disk LDU, all disks must be on line before you can even access the LDU.)

With the Disk Formatter, you can set up a structure in which everything will be on one LDU. Or, you can set up a system LDU with a separate database LDU, UDD LDU, or any combination — depending on your needs and number of disk units.

You might set up a multiple-disk LDU if you need to handle a very large file — perhaps a database file — that won't fit on a single disk. The system will then access the multidisk LDU as one directory file, providing enough space for the large database file.

For an example of a single-disk LDU system, imagine the structure in Figure 1-2 on a single disk. For an example of a large multiple-LDU system, see Figure 1-3.

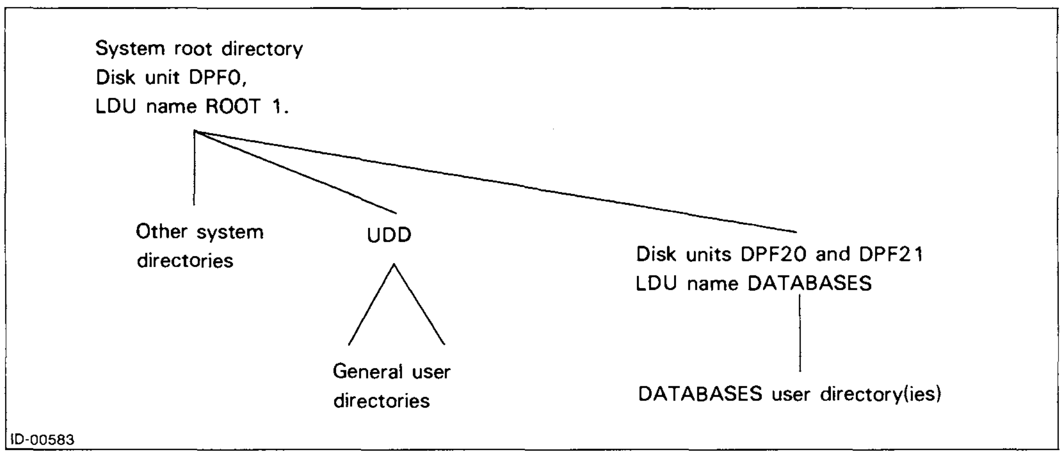


Figure 1-3. A Multiple-LDU System

For your first system, you will create a single-disk LDU in disk unit 0 on the first disk controller. This will be the system disk. But after you have formatted this LDU, you may need to format other physical disks (if you have other units) — and make decisions about their names. We'll give more detail on the specifics later — but mention them here so that you'll know about your options.

What's Involved in System Generation?

The entire system generation procedure assumes that you are starting with blank disks. It takes you all the way through power up and Disk Formatting; installing the starter system; AOS system generation with the AOSGEN program; and creating the multiuser environment, in which AOS can serve many interactive and batch users simultaneously.

Figure 1-4 shows each step that you take in the system generation procedure — beginning to end.

What's Involved in Running the System?

After your tailored, multiuser system has been created, you'll need to determine how to run it.

Running a system is not as ordered a procedure as generating it. Each organization has its own definition of users, job, roles, development, and production. But generally, running the system includes tasks like

- starting up the system, shutting it down, and coping with abnormal shutdown;
- making sure the system serves users: creating and changing profiles for users; starting up the EXEC program and ensuring that batch and device requests run smoothly; perhaps mounting tapes or card decks for users; operating the line printer(s); perhaps doing preventative maintenance;
- executing the installation's main application programs and bringing up network/communications software (if any) on schedule;
- dumping user and system directories and files for backup;
- generating new AOS systems to handle new devices or to make use of enhancements; and formatting new LDUs;
- deciding how to run the system efficiently: how to run program processes, system security, handle updates, and when to get in touch with DG;
- coping with system problems, perhaps running diagnostics;
- making management decisions: how the system can best serve your organization by making users productive. A lot of this involves other software, like data management systems. But there are a number of AOS options as well.

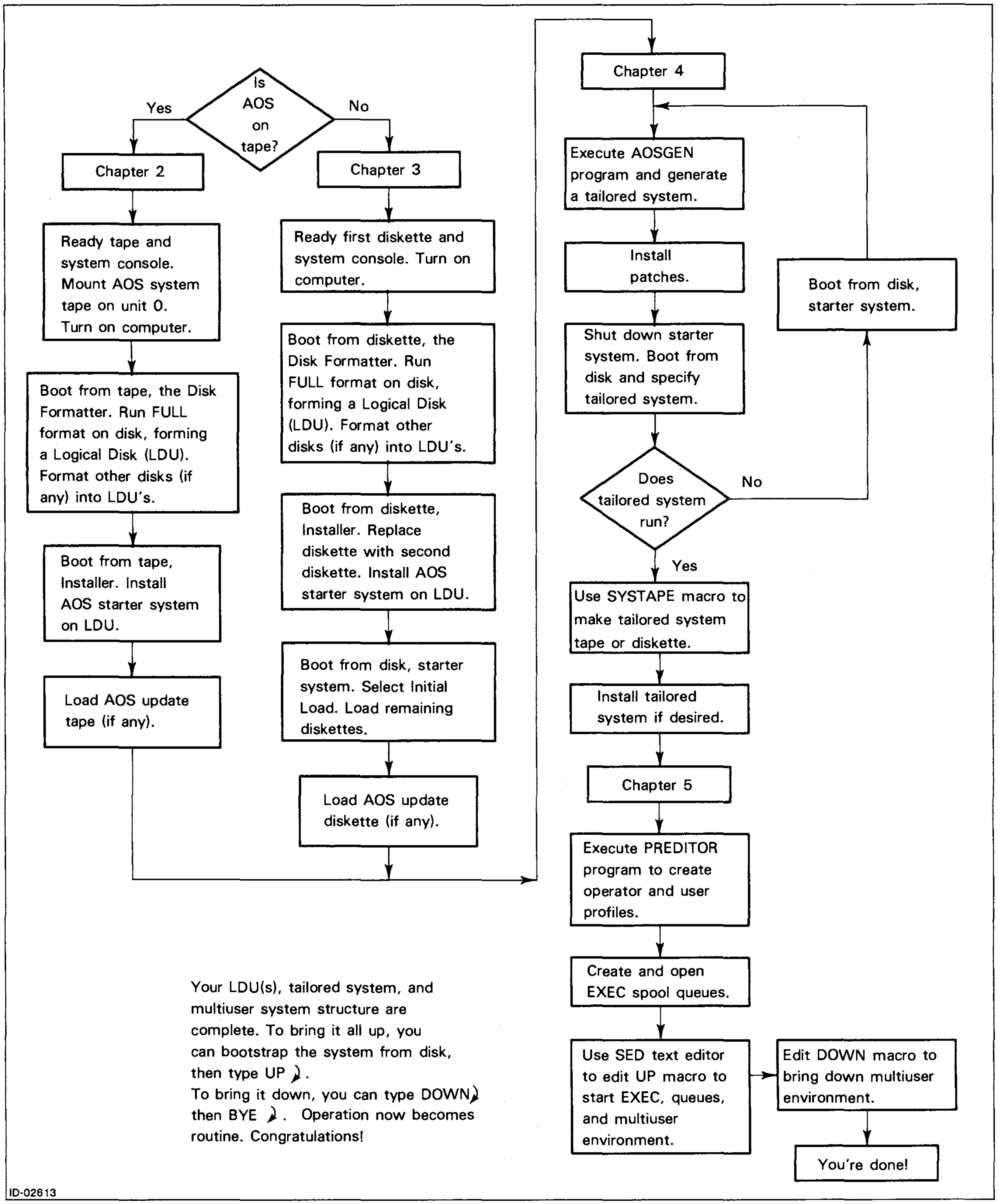


Figure 1-4. How to Bring Up Your First Tailored AOS System

Table 1-1 shows some of these tasks and programs, and the primary place in this book that has information on them. The topics are alphabetical, but this is *not* the main index; it's simply a general directory.

Table 1-1. Finding Material to Help Run the System

Topic	Where It's Described	Topic	Where It's Described
Abnormal shutdown	Chapter 6	Log file, system	Tabbed section PED / SYSLOG REPORT / Chapter 9
Backup procedures	Chapter 9		
Batch	Chapter 8	Management decisions	Chapter 14
Bootstrapping (programs)	Tabbed section STARTUP-SHUTDOWN / Chapter 6	Mounting tapes for users	Chapter 8
CLI commands (operator)	Chapter 9	Multiuser environment, creating	Chapter 5
Dumping files for backup	Chapter 9	Operator tools for run-time	Chapter 7, 8, 9
Executing user application programs	Chapter 14	Power failure	Chapter 6
Errors, serious	Chapter 6, Tabbed section ERRORS / Appendix A	Queues	Chapter 8
EXEC program	Tabbed section EXEC COMMANDS/ Chapter 8		
First system (from tape)	Chapter 2	Shutdown, normal and abnormal	Tabbed section STARTUP-SHUTDOWN / Chapter 6
First system (from diskettes)	Chapter 3		
Fixing LDUs	Tabbed section MEMORY DUMP-ESD-FIXUP / Chapter 6	Shutdown, persistent, abnormal	Chapter 10
Formatting LDUs	Tabbed section DISK FORMATTER / Chapter 11	Startup	Tabbed section STARTUP - SHUTDOWN / Chapter 6
Generating a tailored system	Tabbed section AOSGEN / Chapter 4	User profiles (PREDITOR)	Tabbed section PREDITOR / Chapter 7
Installing a system	Chapter 12	User log on	Chapter 8
Labeled diskettes	Chapter 9		
Labeled tapes	Chapters 8 and 9		

Machine Operation

For the system generation procedure, you'll need to mount and thread magnetic tape or insert diskettes into a diskette unit. If your primary disk unit uses a removable pack, you'll also need to insert a pack in the unit. If you don't know how to do these things, check the Preface for pertinent manuals.

Cautions and Control Characters

This section gives some hints and cautions that will help you during the system generation process. Simply read it; don't do anything yet.

CPU and Disk Switches

During the system generation procedure, power will be on to the CPU and the CPU POWER lamp will be lit. The primary disk will be ready and write-enabled. While you're working with the system (and whenever a system is running), *don't press these switches.*

If power stops to the CPU, the current program may be lost to memory; then it must be reloaded. If power stops to the disk(s) when an AOS program is running, the program will usually abort. In either case, you will need to start the program that was running again — a time-consuming nuisance that you can avoid by leaving the switches alone after the CPU and disks are ready.

When AOS is shut down, you *can* shut off power to the disk(s) if you wish — but you may want to keep CPU power on permanently. Cutting power to the CPU saves energy, but adds a few steps and a few minutes when you bring the system up again.

System Console and Keyboard Control (CTRL) Characters

During the system generation procedure (and afterwards) you'll use the system console (DASHER® display or printing terminal) extensively.

There are several keyboard control sequences and keys that govern console display, interrupt program execution, and the like. You may need one or more of these for system generation; and it will help to know about them afterward — or if you accidentally enter one at the system console.

To type a control sequence, first press the CTRL key; while you hold the CTRL key down, press the other character.

Table 1-2 lists the control characters/keys and their functions.

Table 1-2. Control Characters and Special Keys

Key(s)	What it Does
CTRL-O	Discards console display for the portion of the command that remains to be executed, or until you type CTRL-O, whichever happens first. CTRL-O is a toggle that turns off display, then turns it on again. It does not halt the program. During system generation, you will not want to use CTRL-O. Later, especially on a hardcopy console, you can use it to speed up programs that do a lot of writing to the console.
CTRL-S	Suspends console display. Display resumes where it stopped when you type CTRL-Q. CTRL-S and CTRL-Q are useful when you want to read long files on a CRT screen, or anytime display is too fast to read.
CTRL-Q	Resumes console display. If you suspended display with CTRL-S, then type CTRL-Q. If you stopped display with CTRL-O, CTRL-Q has no effect.
CTRL-U	Erases the current input line. This is handy when you have typed a long, erroneous command line and don't want to press the DEL key many times to erase it. CTRL-U is most useful on hardcopy consoles.
CTRL-C CTRL-A	Interrupts and restarts dialog in stand-alone programs like the Disk Formatter and the Installer. It also interrupts execution of an AOS CLI command. You'll use all these programs, and will find this sequence useful.
CTRL-C CTRL-B	In AOS, aborts the current program process (like the CLI or a text editor). You will probably use this occasionally.
CTRL-D CTRL-D	In AOS, signals an end of file — which usually aborts the process you are running. Generally, avoid this sequence.
CTRL-C CTRL-E	In AOS, creates a memory-image break file (useful for program debugging), and aborts the current process. Generally, avoid this sequence.
DEL key	Erases the last character typed. In AOS stand-alone programs, DEL echoes as _ for each character erased. In the SCP-CLI, DEL echoes as /x (a slash, then character) for each character erased.
BRK key; CMD and BREAK key; BREAK key	This is the break sequence (BRK key on hardcopy consoles, CMD and BREAK/ESC on DASHER® D200- or D400-series CRTs, or BREAK on DASHER D2 CRTs). The break sequence affects only computers with programmed consoles (S/20, S/120, S/140, S/280, DESKTOP GENERATION™, ECLIPSE MV/8000®). (On computers with data switches, the break sequence does nothing.) You can disable it by locking the computer (LOCK switch if any). If the computer is not locked, a break sequence on the system console freezes AOS and gives control to a console loader program (*!* or *SCP-CLI>* prompt.) To have AOS continue, type P ↵. On computers with data switches, nothing happens.

If You Make a Mistake

The programs in the AOS package have good error messages and error recovery. But if you make what appears to be a fatal mistake, you can usually restart the program from the beginning without problems. At worst, you'll need to run a quick disk fixing program.

If, at the system console, everything seems to have stopped, type CTRL-Q. If CTRL-Q has no effect, type CTRL-O. If CTRL-O has no effect, type CTRL-O to undo the first CTRL-O. Finally, if the !prompt has appeared, unexpectedly, type P↵.

What Next?

If you received AOS on magnetic tape, proceed to Chapter 2. Otherwise, skip to Chapter 3.

End of Chapter

Chapter 2
Bringing Up the Starter System on Blank Disks — from Magnetic Tape

Read this chapter

- when your ECLIPSE® or DESKTOP GENERATION Computer has just been installed and you want to bring up AOS on it from magnetic tape;
- whenever you want to format a new disk, install an AOS system on it, load system files onto it, and bring up the installed system.

This chapter tells you how to execute all steps needed before you run the AOSGEN program to generate your first tailored system. The major sections are

- Powering Up
- Running the Disk Formatter
- Installing the AOS Starter System
- Bringing up the AOS Starter System
- What Next?

Powering Up

The following steps assume that CPU power is off.

1. Ready the system console. If this is a hard copy (printing) console, turn it on using the power switch under the keyboard. Press the ON LINE switch and the READY lamp will light.

   If the system console is a CRT display, turn it on (the switch is a rocker switch behind the cabinet or a push-pull knob on the front lower right corner.) Check the ON LINE lamp. If this lamp is lit, fine. If it is not lit, depress the CMD key, hold it down, and press the ON LINE key. The ON LINE lamp should light. (If there is no ON LINE key, check the ON LINE switch behind the console.)

2. If your primary disk unit uses a removable disk pack, make sure a pack is inserted in the unit. Inserting a pack is covered in the illustrated disk booklet. Press DC POWER ON; press WRITE ENABLE; press DRIVE START — and soon, the READY light will glow.

   If your primary disk unit is a sealed unit, press READY. Soon, the READY light will glow. For either disk, don't touch the switches until you have generated a tailored system, tested it, and shut it down (if then).

   If you want to format multiple disks, make sure each is ready as above.

3. If your computer is *not* an ECLIPSE MV/8000, skip to step 4. If your computer *is* an MV/8000, continue with this step.

3a. If the supplied diskette is not in its slot, insert it. Remove the diskette from its protective envelope and make sure it is write-enabled (with tape over the write-protect slot on the edge). Open the diskette unit door, and slide the diskette into its slot with the paper label facing right. The diskette should slide in smoothly and come to a firm stop. Close the diskette door.

3b. Press the computer POWER switch to ON. When you turn power on, EPROM in the CPU runs some power-up tests; then it types

CONSOLE READY

MV/8000 SYSTEM CONTROL PROGRAM REV n
TYPE HELP<CR> FOR HELP
COPYRIGHT (C) DATA GENERAL ...

STARTING POWER UP SEQUENCE

ENTER DATE (MO DAY YR)

If you see an incomplete CONSOLE READY message, or nothing, consult the MV/8000 power up error table in Appendix A. If the *MV/8000 SYSTEM...* message doesn't appear, make sure the diskette is in its slot, and turn power OFF and ON again. If this fails, try another diskette.

3c. Enter the date, separating month, day, and year by a space, followed by a NEW LINE character (↵). For example, for April 15, 1985, you'd type

4 15 85 ↵

The system console says

ENTER TIME (HR MIN)

3d. Enter the time, based on a 24-hour clock. For example, for 1:30 p.m., type

13 30 ↵

MICROCODE (1= STD, 2= C350/MMPU[1])?

3e. You want to load the C/350 microcode, so type

2 ↵

LOADING FROM FILE MVC350 REV. n

.
. (microcode status messages appear here)
.

BEGIN SYSTEM INITIALIZATION
 # OF 256 KB MEMORY MODULES - n
END SYSTEM INITIALIZATION

SCP-CLI>

The whole load takes a little over a minute. The SCP operating system, which allows you to run other programs, now has control. (If the system console types an error message, check the diskette again and try another if needed.)

Go to step 5.

4. If the computer has a LOCK switch or key, place it in the unlocked position. Then press the computer POWER switch to ON.

5. Select a tape unit and turn power on to it.

If the tape unit has one reel above the other, and its control panel has a DENSITY switch, it is a type MTB unit. Set the DENSITY switch to LOW for 800-bpi tape, or HIGH for 1600-bpi tape. If the tape unit has one reel above the other, but no DENSITY switch, it is a type MTA unit. The tape unit that you use must be unit 0 and its controller must be device code 22. Dial 0 on the unit thumbwheel and make sure that no other tape unit that is ON LINE has the same number. Then, set the tape panel switch to LOW DENSITY.

If the tape unit has no SELECT thumbwheel and its reels are side by side, it is a type MTC unit. If the tape unit has no reels and accepts a tape *cartridge,* it is also a type MTC unit. The tape unit you use must be unit 0 and its controller must be device code 22.

6. Get the DG-supplied AOS system tape. Mount and thread this tape on unit 0. (Remove the plastic write-enable ring first, if the ring is present.)

7. Press the tape UNLOAD/BOT switch to BOT; then press the ON LINE/OFF LINE switch to ON LINE. The tape will move forward and stop.

You've finished powering up and can run the Disk Formatter.

Running the Disk Formatter

The Disk Formatter makes physical disks into LDUs; it can also change LDU specifications. The Disk Formatter does this by writing identifiers so that AOS will know what disk and LDU it is accessing. The Formatter can also check the disk surface for *bad blocks* (flawed areas that won't hold information).

Running the Disk Formatter is relatively easy, but — because the Formatter checks each bit on the disk — takes more time than other system generation procedures.

Mistakes and Errors

If you type an incorrect answer to a Disk Formatter question, and have not yet pressed ⤶ to enter the answer, press the DEL key or CTRL-U to erase the wrong characters.

If you have pressed ⤶ and want to abort formatting, enter CTRL-C CTRL-A and go to step 10. If you have an S/20, S/120, S/140, S/280, or an MV/8000 and CTRL-C CTRL-A doesn't work, type the break sequence (Table 1-2) and return to step 8.

If you abort formatting by either method during surface analysis, be sure to run the entire FULL format again.

If the Disk Formatter reports a disk or other error, *make sure the disk unit is write-enabled;* if it is write-enabled, check the error message in the table near the end of the Disk Formatter chapter.

Disk Formatter Dialog

8. The next step is to program load. How you do this depends on the kind of computer you have, as follows:

8a. With an S/20, S/120, or S/280, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type 22H. For example,

```
! 22H      (S/20, S/120, S280)
```

and go to step 9.

8b. With an S/140, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type 11A. When you do, the system displays a number xxxxxx, which you can ignore. Following this number, type 100022⤶. Next to the second prompt, type 100022L⤶. For example,

```
! 11A    xxxxxx     100022 ⤶    (S/140)
! 100022L
```

and go to step 9.

8c. With an MV/8000, the system console displays an *SCP-CLI>* prompt. Next to the prompt, type RESET↲. Then, next to the second prompt, type BOOT 22↲. For example,

SCP-CLI> RESET ↲ (MV/8000)
SCP-CLI> BOOT 22 ↲

and go to step 9.

8d. If your computer has hardware data switches (numbered 0 (or X4/0) though 15), make sure they are set to 100022 (switches 0, 11, and 14 up, the others down). Lift and release the RESET switch; lift and release the PR LOAD (PROG LOAD) switch.

9. The step(s) above program loads the tape bootstrap, TBOOT. TBOOT says

FROM MT-0:

TBOOT is asking for a file number. Type 2↲, the number of the Disk Formatter file.

2 ↲

TBOOT moves the tape forward to file 2, then loads and executes the Disk Formatter. The Formatter says

AOS DISK FORMATTER REV n

FULL FORMAT DESTROYS ANY AOS DISK STRUCTURE, PARTIAL RETAINS IT.

FULL (F) OR PARTIAL (P OR <NL>)?

10. Type

F ↲

FULL FORMAT

ENTER UNIT NAMES FOR EACH UNIT IN THE LDU (<NL> WHEN DONE):

DISK UNIT NAME?

11. The *DISK UNIT NAME* question starts a sequence of questions to identify this LDU.

Each AOS disk unit name has the form DPx0, as shown in Table 2-1. Find the unit name of the first disk on the controller and type it. For example,

DPF0 ↲ (or DPN0 ↲ or DPJ0 ↲)

DEVICE CODE?

11a. For any system, unless you *know* that this disk unit is connected to a nonstandard device code, press ↲ for the default:

↲

DISK UNIT NAME?

11b. The Disk Formatter will cycle the *UNIT NAME?/DEVICE CODE?* questions until you answer ↲ to *UNIT NAME?*. This allows you to create an LDU that includes more than one physical disk. An LDU can include up to eight physical disks.

Generally, for your first system, you want a single-disk LDU; and in most cases, you will want every LDU to be a single-disk LDU. The pros and cons of multiple-disk LDUs are detailed in the Disk Formatter chapter. For now, answer the question with

↲

DO YOU WANT TO ALLOCATE A DIAGNOSTIC AREA? [N]

11c. This question lets you reserve an area on disk for later installation of DG's Advanced Diagnostic Executive System (ADES). ADES can run from a medium other than disk, but it runs much faster from disk; also, diagnostics are easier to run remotely if ADES is on disk. To use ADES, you must purchase it and have it installed on the disk by a DG field engineer. Also, ADES requires a minimum of 8000 disk blocks (3.6 Mbytes). Consequently, your disk must be larger than 25 Mbytes. Space reserved for ADES is lost for AOS file storage.

Unless you really want to install ADES and run it from disk, answer No by pressing ↵. ADES runs only from the system disk — which means you should always say No if the disk isn't the system disk. If you say no, skip to step 11e.

If you really want to reserve an area for ADES, type Y ↵. Then, the Disk Formatter asks

ENTER THE NUMBER OF BLOCKS (11610 TO 35230) THAT ARE REQUIRED. [23420]

11d. The displayed figures are octal. ADES needs at least 8000 blocks (17500 octal). The default, 23420, is 10,000 blocks. Decide on the number of disk blocks needed for the diagnostics you want installed; then type this number (octal!). The Formatter will now assume that this disk is a system disk.

DISK NUMBER 1: 00000000000 THRU n

LOGICAL DISK UNIQUE ID (1 TO 6 CHARS) [ ]?

The numbers 0 through n are the first and last logical addresses on the disk, in octal.

11e. The Disk Formatter wants a unique ID for the disk. Later, AOS will use this to keep track of the physical disks in this LDU. The ID must be 1 to 6 characters long. Any AOS filename character is legal: A through Z (uppercase and lowercase are treated the same), 0 through 9, period (.), dollar sign ($), question mark (?), and underscore (_).

Generally, use an ID that is as close as possible to the name (filename) you will give the LDU. For example, you could use an ID of ROOT1:

ROOT1 ↵

LOGICAL DISK NAME (1 TO 31 CHARS) [ ]?

11f. Later, when you boot this LDU or initialize it from the CLI, the name you type now will be displayed.

This is the master LDU; it will be the system root directory (:). So the name you type is not important in terms of file access.

But for any disk that is not the system disk, the name you type here becomes the filename of the LDU. People can use this name just as any other directory filename. For example, if you have a lot of users and want to put some of them on a nonsystem LDU, you might name the LDU something like UDD1. Or, for a big multidisk LDU, you might choose the name DATABASE. You can always change the LDU name later with the Disk Formatter Partial format (covered in the Disk Formatter chapter).

For your first LDU, come up with a name with from 1 to 31 filename characters and type it. For simplicity, you can make this the same as the unique ID. For example,

ROOT1 ↵

Whatever name (and unique ID) you chose, note them for later reference.

ACCESS CONTROL LIST

USER NAME OR TEMPLATE (1 TO 15 CHARS)?

11g. A user, identified by a user name, or a group of users, identified by a template, can have different kinds of access to a logical disk. A good general-purpose username template is + , which specifies all users:

+)

PRIVILEGES (O,W,A,R,E, NEW-LINE)?

Table 2-1. Common AOS Disk Models and Names

Disk Model Number and Description	Default Device Code of Controller	Disk Number on Controller	Disk Unit Name(s)
4231A. A moving-head disk unit with a top-loading 92-Mbyte disk.	33	first (0) second (1) third (2) fourth (3)	DPE0 DPE1 DPE2 DPE3
	73	first (0) second (1) third (2) fourth (3)	DPE10 DPE11 DPE12 DPE13
6045. A dual moving-head disk unit that includes one 5-Mbyte fixed disk and one 5-Mbyte removable disk cartridge. (Note that the same controller that runs 6030 diskette units can run a 6045 disk unit.	33	first pair second pair third pair fourth pair	(Removable / Fixed) DPD0 / DPD4 DPD1 / DPD5 DPD2 / DPD6 DPD3 / DPD7
	73	first pair second pair third pair fourth pair	(Removable / Fixed) DPD10 / DPD14 DPD11 / DPD15 DPD12 / DPD16 DPD13 / DPD17
6060, 6061, 6067, and 6122; 6160 and 6161. All are moving-head disk units. The 6060, 6061, 6067, and 6122 are free-standing units with removable disks; a controller can run four units. The 6160 and 6161 are sealed units with bay-mounted fixed disks; a controller can run two units.	27	first (0) second (1) third (2) fourth (3)	DPF0 DPF1 DPF2 DPF3
	67	first (0) second (1) third (2) fourth (3)	DPF10 DPF11 DPF12 DPF13

(continues)

Table 2-1. Common AOS Disk Models and Names

<table>
<tr><th>Disk Model Number and Description</th><th>Default Device Code of Controller</th><th>Disk Number on Controller</th><th>Disk Unit Name(s)</th></tr>
<tr><td rowspan="2">A 6060 holds a 96-Mbyte disk; a 6061 holds a 190-Mbyte disk; a 6067 holds a 50-Mbyte disk; a 6122 holds a 277-Mbyte disk; a 6160 holds a 73-Mbyte disk; a 6161 holds a 147-Mbyte disk.</td><td>none; chosen at installation.</td><td>first (0)
second (1)
third (2)
fourth (3)</td><td>DPF20
DPF21
DPF22
DPF23</td></tr>
<tr><td>none; chosen at installation.</td><td>first (0)
second (1)
third (2)
fourth (3)</td><td>DPF30
DPF31
DPF32
DPF33</td></tr>
<tr><td rowspan="2">6063, 6064, 6066. A bay-mounted, fixed-head disk. The 6063 holds a 1-Mbyte disk; a 6064 holds a 2-Mbyte disk; a 6066 has two 6064 units and one controller.</td><td>26</td><td>first (0)
second (1)
third (2)
fourth (3)</td><td>DKB0
DKB1
DKB2
DKB3</td></tr>
<tr><td>66</td><td>first (0)
second (1)
third (2)
fourth (3)</td><td>DKB10
DKB11
DKB12
DKB13</td></tr>
<tr><td rowspan="2">6070. A dual moving-head disk unit that includes one 10-Mbyte fixed disk cartridge and one 10-Mbyte removable disk cartridge.</td><td>33</td><td>first pair
second pair
third pair
fourth pair</td><td>(Removable / Fixed)
DPG0 / DPG4
DPG1 / DPG5
DPG2 / DPG6
DPG3 / DPG7</td></tr>
<tr><td>73</td><td>first pair
second pair
third pair
fourth pair</td><td>(Removable/ Fixed)
DPG10 / DPG14
DPG11 / DPG15
DPG12 / DPG16
DPG13 / DPG17</td></tr>
<tr><td>6098, 6099, 6100, 6103. A bay-mounted moving-head unit. May have a 1.25-Mbyte diskette on the same controller. Disk holds 12.5 or 25 Mbytes.</td><td>33
73</td><td>first(0)
first (0)</td><td>DPI0
DPI10</td></tr>
<tr><td rowspan="2">6101, 6104, 6220-D, 6222-D. A bay-mounted, sealed unit, with a 1.26-Mbyte diskette. Disk capacity is 12.5, 25, 5, or 15 Mbytes respectively. They are designed for S/20s and other microECLIPSE™ systems.</td><td>26</td><td>first (0)</td><td>DPK0 (diskette is DPK1)</td></tr>
<tr><td>66</td><td>first (0)</td><td>DPK10 (diskette is DPK11)</td></tr>
</table>

(continued)

Table 2-1. Common AOS Disk Models and Names

Disk Model Number and Description	Default Device Code of Controller	Disk Number on Controller	Disk Unit Name(s)
6102, 6105, 6220, 6222. A bay-mounted, sealed unit. Disk capacity is 12.5, 25, 5, or 15 Mbytes respectively.	26 66	first (0) first (0)	DPK0 DPK10
6224. A 15-Mbyte unit that runs on the burst multiplexor channel (BMC) instead of the data channel. It's designed for S/20 and other microECLIPSE systems.	25 65	first (0) first (0)	DPL0 DPL1
6225, 6227, 6234. A fixed, bay-mounted moving-head disk unit that holds 5, 15, or 50 Mbytes respectively.	33 73	first(0) first (0)	DPI0 DPI10
6225-D, 6227-D, 6234-D. A fixed, bay-mounted moving-head disk unit that holds 5, 15, or 50 Mbytes respectively, with up to three 1.25-Mbyte diskettes on the same controller.	33 73	first(0) first (0)	DPI0 DPI10
6236 and 6237; 6239 and 6240 Each is a rack-mounted, sealed, moving-head unit with the power switch on the upper right. It has a LED display that shows the unit number and can show the current cylinder or disk fault code. A controller can run four units. Up to three units fit in a cabinet.	24	first (0) second (1) third (2) fourth (3)	DPJ0 DPJ1 DPJ2 DPJ3
	64	first (0) second (1) third (2) fourth (3)	DPJ10 DPJ11 DPJ12 DPJ13
A model 6236 unit holds 354 Mbytes; a model 6237 is three 6236 units in one cabinet, on one controller.	none; chosen at installation	first (0) second (1) third (2) fourth (3)	DPJ20 DPJ21 DPJ22 DPJ23
A model 6239 unit holds 592 Mbytes; a model 6240 is three 6239 units in one cabinet, on one controller.	none; chosen at installation	first (0) second (1) third (2) fourth (3)	DPJ30 DPJ31 DPJ32 DPJ33
6271. A compact, 15-Mbyte unit designed for DESKTOP GENERATION systems.	26	first (0) second (1)	DPN0 DPN1
6280. Same as model 6224, but holds 50 Mbytes instead of 15 Mbytes.	25 65	first (0) first (0)	DPL0 DPL10
6290. Two 6239 units in one cabinet on one controller.	see 6236		
6301. Same as model 6271, but holds 38.6 Mbytes instead of 15 Mbytes.	26	first (0) second (1)	DPN0 DPN1
6336. Same as model 6271, but holds 71.2 Mbytes instead of 15 Mbytes.	26	first (0) second (1)	DPN0 DPN1

(concluded)

11h. The Formatter wants to know which privileges to give the user(s) you just specified. There are five types of privileges: Owner, Write, Append, Read, Execute (OWARE). A NEW LINE gives the user no privileges. Execute (E) access will suffice for most LDUs. So type

E ↵

USER NAME OR TEMPLATE (1 TO 15 CHARS)?

11i. The Disk Formatter will cycle the *USER NAME/PRIVILEGES* questions, allowing you to give very specific user/access information, until you answer ↵ to *USER NAME...* You can change the access control list of any LDU when AOS is running — or run the Formatter again to do it — so the answers you give in steps 11e and 11f are not critical. Generally, answer +↵ to question 11e, answer E↵ to question 11f, and answer ↵ to this question. Later, if needed, you can change access to the LDU. So press

↵

SURFACE ANALYSIS?[N]

12. This step starts a series on surface analysis for this LDU. (The value in brackets is the default, which the Formatter will use if you simply answer ↵.)

For each new disk, you should answer Y↵ to this question. During analysis, the Formatter writes a pattern to each 16-bit word on the disk and reads it back. This destroys all existing information on the disk — but a new disk has no information. Answer yes by typing Y↵:

Y ↵

DISK NUMBER?

12a. The Formatter wants the number of the disk to analyze. For a single-disk LDU, press

↵

YOU MAY RUN UP TO FIVE (5) PATTERNS...HOW MANY WOULD YOU LIKE TO RUN?

12b. To ensure the validity of the disk structure, the Formatter writes to the disk a variety of bit patterns, reads from the disk, and compares the results. When the results differ, AOS marks the block as bad. You can specify up to five bit patterns, and we recommend that you do.

Each pattern takes up to 50 minutes, depending on the disk model, as shown in Table 2-2. So five patterns may take a while. To run five patterns, type

5 ↵

ANALYZING DISK #n
--RUNNING PATTERN nnnnnn

The Formatter runs the patterns you specified, one by one. If it finds too many bad blocks, it aborts; this usually means that the disk heads are misaligned or that the disk needs hardware formatting. But in most cases, it's simply a matter of waiting.

When the Formatter has finished the patterns, it describes the bad blocks on the disk.

12c. If the Formatter found no bad blocks, it says *0 BAD DISK BLOCKS*. Skip to step 12d, *ADDITIONAL BAD BLOCK NUMBER?* question.

If it found any bad blocks, it says

n BAD DISK BLOCKS
PRINT BAD BLOCK STATISTICS? [N]

Table 2-2. Surface Analysis Times for Popular Disks

Disk Model	Capacity (megabytes)	Time per Test Pattern
6060	96	13 minutes
6061	190	26 minutes
6063	1	3 minutes
6064, 6066	2	6 minutes per disk unit
6122	277	26 minutes
6160	73	11 minutes (approximately)
6161	147	22 minutes (approximately)
6227, 6271	15	15 minutes (approximately)
6234	50	8 minutes (approximately)
6236	354	50 minutes (approximately)

12d. The Formatter is asking if you want to see the bad block statistics. These statistics may be useful, so answer yes.

Y ↵

The Formatter now displays the bad block statistics on the console. If there are bad blocks, you might want to note them. There should be few, if any, bad blocks on a new disk.

ADDITIONAL BAD BLOCK NUMBER (<NL>) WHEN DONE):

12e. You have no additional bad blocks to enter, so press

↵

BITMAP SIZE: n

The bitmap is a system table that describes which blocks are in use and which are free for data storage.

BITMAP ADDRESS? [default]

13. Select the default by pressing

↵

SYSTEM DISK? [Y]

14. This step starts a series that determines whether and where an AOS system will reside on the disk.

Your first LDU must be a system disk, so answer ↵ for the default.

↵

OVERLAY AREA SIZE? [default]

14a. Choose the default area size by pressing

↵

OVERLAY AREA ADDRESS? [default]

14b. Choose the default address by pressing

)

DISK NUMBER n REMAP AREA SIZE? [default]

15. Choose the default remap area size by pressing

)

DISK NUMBER n REMAP AREA ADDRESS? [default]

15a. Choose the default remap area address by pressing

)

--LOGICAL DISK CREATED

DONE!

Congratulations! You've formatted an LDU as a system disk. It will rarely — if ever — need full formatting again. If this LDU may be run as a nonmaster LDU, we suggest that you note the date, LDU ID and unit name, and any bad block information — and attach the note to the disk unit (or, for a removable disk, the plastic housing).

The Formatter is done. If you have other new disks, someone must format them into LDUs before they can be used. If you know what LDU configuration(s) you want, you might want to create the LDU(s) now — while you're familiar with the procedure. To do it now, type P) (or press the CONT panel switch) and return to step 10. To format a disk *not* described in this chapter, see the Disk Formatter chapter.

If you don't want to format other disks now, proceed and install the AOS starter system.

Installing the AOS Starter System

The Installer program installs any AOS system from a system tape onto an LDU. (AOS must reside on disk before it can run.)

The steps described here work the same way for a tailored system tape that you will create later: the Installer simply installs the tailored system, not the starter system.

Mistakes and Errors

If you make a mistake, handle it the same way as with the Disk Formatter.

If the Installer reports a disk or other error, check the error message in the table near the end of the Installer chapter.

If the Installer stops with an *ABORT* message, return to step 16 and start the Installer dialog again.

Running the Installer

16. Repeat step 8 to program load from tape again

FROM MT-0:

17. The Installer is in tape file 3, so answer

3 ↵

TBOOT moves the tape forward to file 3, loads the Installer into memory, rewinds the tape, then executes the Installer. The Installer says

AOS INSTALLER REV n

ENTER ALL UNITS IN LDU:

DISK UNIT NAME?

18. Type the name of the unit that holds your newly created LDU. This is the same unit name you gave to the Disk Formatter earlier, in step 11. For example,

DPF0 ↵ (or DPN0 ↵ or DPJ0 ↵)

DEVICE CODE?

19. As with the Disk Formatter, press ↵ to select the default code:

↵

--DISK BOOTSTRAP INSTALLED

INSTALL A SYSTEM BOOTSTRAP?

20. You must install a system bootstrap the first time you install a system. The first disk in a system LDU must have a disk bootstrap and a system bootstrap; it usually has a system as well. Answer Y ↵:

Y ↵

FROM MAG TAPE (M) OR DISKETTE (D)?

20a. Type M ↵.

FROM MT-0, FILE #:

21. The system bootstrap, SYSBOOT, is in tape file 4. So type

4 ↵

--SYSTEM BOOTSTRAP INSTALLED

INSTALL A SYSTEM?

22. You want to install a system, so type Y ↵:

Y ↵

FROM MAG TAPE (M) OR DISKETTE (D)?

22a. Type M ↵.

FROM MT-0, FILE #?

23. The AOS system is always on file 5 of a system tape, so type

5 ⤶

The Installer now copies the AOS system from tape onto the LDU. Then it says

--SYSTEM INSTALLED

DONE!

You've installed the needed bootstraps and an AOS system on your LDU. Now you can bring up the AOS system.

Bringing Up the AOS Starter System

24. Now you must program load again, this time from the *disk*. The default device code of a DPF-type disk is 27; for a fixed-head disk it is 26; for a DPJ-type disk, it is 24; for all other disks, see Table 2.1. Follow these steps (as before, but for the disk):

24a. With an S/20, S/120, or S/280, the system console displays a *!* prompt. Next to the prompt, type nnH, where nn is the device code. For example,

! 27H (S/20, S/120, S/280)

and go to step 25.

24b. With an S/140, the system console displays a *!* prompt. Next to the prompt, type 11A. The system will display a 6-digit number, after which you type 1000nn⤶, where nn is the device code. Next to the second prompt, type 1000nnL⤶. For example,

! 11A *xxxxxx* 100033 ⤶ (S/140)
! 100033L

and go to the step 25.

24c. With an MV/8000, the system console displays an *SCP-CLI>* prompt. Next to the prompt, type RESET⤶. Then, next to the second prompt, type BOOT nn⤶, where nn is the device code. For example,

SCP-CLI> RESET ⤶ (MV/8000)
SCP-CLI> BOOT 27 ⤶

and go to step 25.

24d. If your computer has hardware data switches (numbered 0 (or X4/0) though 15), make sure they are set to 1000nn, where nn is the device code. For code 27, set switches 0, 11, 13, 14, and 15 up; for code 24, set switches 0, 11, and 13; for code 26, set switches 0, 11, 13, and 14 up; for code 33, set switches 0, 11, 12, 14, and 15 up. Lift and release the RESET switch; lift and release the PR LOAD (PROG LOAD) switch.

Your program load steps bring the *disk* bootstrap into memory. It reads the system bootstrap into memory. The system bootstrap says

SPECIFY EACH DISK IN THE LDU

DISK UNIT NAME?

25. The system bootstrap has control. Type the name of the unit that holds your LDU. This is the same unit name you gave to the Disk Formatter and Installer. For example,

DPF0 ⤶ (or DPN0 ⤶ or DPJ0 ⤶)

DEVICE CODE?

26. As before, press) for the default device code:

)

SYSTEM PATHNAME?

27. Having installed a system, you can press) to load and execute it.

)

AOS REV n

DATE (MM/DD/YY)?

28. You're almost done. Enter the date as numbers for month, day, and year. Spaces or slashes can separate each number. For example, for April 15, 1985, you'd type

4 15 85)
TIME (HH:MM:SS)?

29. Enter the time, based on a 24-hour clock, in hours, minutes, and seconds. (Minutes and seconds are optional. If you omit them, the system sets each to 0.) Use spaces or colons to separate each number pair. For example, for 2:30 p.m., you'd type

14 30)

OVERRIDE DEFAULT SPECS [N] ?

30. SPECS means the parameters in the system specification file created during AOSGEN. For your first system, you must answer yes, so type

Y)

MASTER LDU: ROOT1 (logical disk name you gave to Formatter)
NUMBER OF BUFFERS IN CACHE [default]?

31. Press

)

SWAP FILE DEFINITION [default]?

32. Press

)

INITIAL LOAD [N]?

33. In an *INITIAL LOAD,* the system loads the CLI and other needed files onto the LDU. These files must be loaded the first time you bring up the starter system; they need not be loaded again unless — later on — you want to load a new revision of AOS. Answer yes by typing

Y)

FILENAME [MTA0:6]?

34. Press) to specify the default, tape file 6:

)

You'll see the tape moving as the starter system copies files from it to the disk, then

AOS CLI REV n date time

)

Congratulations! You've brought up AOS and its CLI. The CLI's prompt,), tells you that it is ready for a command.

(If you get a *FATAL ERROR 25* message, a needed file wasn't loaded. Perhaps you forgot to answer Y⤶ to the *INITIAL LOAD* question. In any case, run Emergency Shutdown (ESD). (See Chapter 6 for information on running ESD.) Then, return to step 24 to try again. (See Appendix A for a description of errors by numeric code.)

The CLI is far more sophisticated than any program you've been using. It has many commands and fine error handling; it has a HELP feature that you can use after you've loaded the next tape file. You can interrupt executing CLI commands by typing CTRL-C CTRL-A. As always, you can delete characters with DEL and delete bad lines with CTRL-U; and you can type CTRL-S to suspend display and CTRL-Q to restore it (useful for reading long files on a display console.)

35. If your system console is a CRT display, change the console characteristics by typing

```
) CHAR/605X/OFF/NAS ⤶
```

36. Now you can load tape file 7 with all its programs. First, turn on SUPERUSER to provide write access.

```
) SUPERUSER   ON ⤶
*)
```

With AOS running, you are subject to file access controls — which means that might get an error message if you tried to load files into the root directory. SUPERUSER allows you to bypass all file access controls. The asterisk before the) prompt means that SUPERUSER is on.

37. Load the files in tape file 7 by typing

```
*) DIR   : ⤶

*) LOAD/V   @MTA0:7 ⤶
.
.
.
```

Even if the tape is mounted on an MTB or MTC unit, you must specify MTA to the starter system.

The CLI verifies (/V switch) the directory and filenames copied from tape, by printing their names on the system console. The whole directory structure on the tape is copied, creating directory :UTIL (with utilities), directory :SYSGEN (for system generation), and directory :HELP (for help).

All these directories (along with files like DFMTR, the Disk Formatter) are copied into the *root* directory. The root directory's name is : (colon), so the pathname to any of these newly-created directories is

```
:directory-name
```

for example, :UTIL.

After all these files have been loaded, the CLI SUPERUSER prompt returns.

```
*)
```

38. Rewind the tape by typing

```
*) REWIND @MTA0 ↲
```

39. Now, if you received an AOS *update* tape, get it. If not, skip this step. Updates have revision numbers with the last two digits not 00; for example, 7.01 is an update number. Mount the tape on unit 0, and type

```
*) DIR :↲

*) LOAD/V/R @MTA0:0 ↲
.
.  (CLI verifies load of update notice and patch files.)
.
```

This puts AOS update and patch files on the LDU, for access later on.

```
*) REWIND @MTA0 ↲
```

40. Turn SUPERUSER off by typing

```
*) SUPERUSER OFF ↲
)
```

The) is the standard CLI prompt.

41. You're done with the tape, so you might want to remove it from the tape unit. Clip the cover ring around all DG-supplied tapes, and store them in a handy place. You may need them again if you need to restore this revision of AOS to an LDU.

Well done! You've powered up, formatted at least one LDU, installed an AOS system on it, brought up AOS, and loaded all files you need to generate your tailored system.

If you're interested in the files on the system tape, see Table 2-3; all of these files are now on your LDU. The LDU also contains directory :PATCH, with current patch files (if you loaded an update tape).

Figure 2-1 is a summary of all the steps you've taken — from turning on the system console to dismounting the last tape.

Table 2-3. AOS System Tape File Format

Tape File Number	Program Filename	Tape File Contents
0	TBOOT	Tape bootstrap; a short program that can load files 1, 2, and 3 from this tape.
1	FIXUP	Disk Fixer utility, which finds and optionally corrects disk file errors if abnormal AOS shutdown occurs. TBOOT loads this program into memory and executes it after you type 1) to the FROM MT-0: query.
2	DFMTR	Disk Formatter utility, which formats physical disks into LDUs. TBOOT loads DFMTR into memory and executes it after you type 2) to the FROM MT-0: query.
3	INSTL	Installer utility, which installs an AOS system from a system tape. TBOOT loads INSTL into memory and executes it after you type 3) to the FROM MT-0: query.
4	SYSBOOT	The system bootstrap program loads an AOS system or other program into memory *from disk,* then executes the program. The Installer installs SYSBOOT onto an LDU.
5	AOS System	On the DG-supplied system tape, this is the AOS starter system. On a system tape you make, it is a tailored AOS system. INSTL installs this file on an LDU.
6	First Dump File	The CLI and other system program files, including the system GHOST, peripheral manager (PMGR), and copies of programs in tape files 0 though 4. AOS copies the contents of this tape file into the LDU root directory when you specify INITIAL LOAD.
7	Second Dump File	You copy the contents of this file onto disk with the CLI LOAD command, as part of the initial load procedure. This tape file contains nearly all AOS support software files, including the - Disk File Editor (DEDIT) - DISPLAY file display - Error message (ERMES) and message object files (.OBs) - EXEC and PREDITOR - HELP directory and files - LABEL tape labeler - Link - Library File Editor - Macroassembler (MASM) - Process Enviroment Display (PED) - Release Notice (latest on software) - SED and SPEED text editors - System macros - Utility program symbol table files (.ST) - SYSGEN directory with AOSGEN system generation program and libraries

Powering Up

1. Have system console ON and ON LINE.
2. Have disk pack(s) inserted (if removable); disk unit(s) write-enabled and READY.
3. Unless computer is MV/8000, go to step 4.

3a. CPU diskette in slot.

3b. Press computer POWER switch to ON.

CONSOLE READY

. . .

3c. *ENTER DATE* 4 15 85 ⤶ (current date)

3d. *ENTER TIME (HR MIN)* 13 30 ⤶ (current time)

3e. *MICROCODE (1 = STD, 2 = C/350/MMPU[1]?* 2 ⤶

SCP-CLI>

Go to step 5.

4. Unlock the computer. Turn power on.
5. Turn POWER ON to tape unit 0.
6. Mount DG-supplied AOS system tape on unit 0.
7. Press BOT and ON LINE switches to start tape.

Running the Disk Formatter

8. Program load from tape (device code 22) as follows:

8a. With an S/20, S/120, or S/280, type

! 22H

8b. With an S/140, type

! 11A *xxxxxx* 100022 ⤶
! 100022L ⤶

8c. With an MV/8000, type

SCP-CLI> RESET ⤶
SCP-CLI> BOOT 22 ⤶

8d. With hardware data switches, make sure they are set to 100022 (switches 0, 11, and 14 up). Lift RESET, then PR LOAD (PROG LOAD).

9. *FROM MT-0:* 2 ⤶

AOS DISK FORMATTER REV n

FULL FORMAT DESTROYS ANY AOS DISK STRUCTURE,
PARTIAL FORMAT RETAINS IT.

DG-26892T

Figure 2-1. Step Summary, Bringing Up the AOS Starter System on Blank Disk(s), from Magnetic Tape (continues)

10. *FULL (F) OR PARTIAL (P OR <NL>)?* F ↵
FULL FORMAT
ENTER UNIT NAMES FOR EACH UNIT IN THE LDU (<NL> WHEN DONE):

11. *DISK UNIT NAME?* DPN0 ↵ (or DPN0 ↵ or other disk unit name)

11a. *DEVICE CODE?* ↵

11b. *DISK UNIT NAME?* ↵

11c. *DO YOU WANT TO ALLOCATE A DIAGNOSTIC AREA? [N]*

Unless you want to allot 8,000 blocks for later installation of the Advanced Diagnostic system, press ↵ and skip to step 11e. To reserve an area, type Y ↵.

11d. *ENTER THE NUMBER OF BLOCKS (11610 TO 35230)...[23420] ?* ↵
DISK NUMBER 1: 00000000000 THRU n

11e. *LOGICAL DISK UNIQUE ID (1 TO 6 CHARS) [ ]?* ROOT1 ↵ (valid disk ID)

11f. *LOGICAL DISK NAME (1 TO 31 CHARS) [ ]?* ROOT1 ↵ (valid LDU name)

11g. *USER NAME OR TEMPLATE (1 TO 15 CHARS)?* + ↵

11h. *PRIVILEGES (O, W, A, R, E, NEW-LINE) ?* E ↵

11i. *USER NAME OR TEMPLATE (1 TO 15 CHARS)?* ↵

12. *SURFACE ANALYSIS? [N]* Y ↵

12a. *DISK NUMBER?* ↵

12b. *YOU MAY RUN UP TO FIVE (5) PATTERNS...HOW MANY WOULD YOU LIKE TO RUN?* 5 ↵ (Choose pattern(s))
ANALYZING DISK #n
--RUNNING PATTERN nnnnnn (takes 8 to 52 minutes per pattern)

12c. If it found no bad blocks, go to 12e.

12d. *nBAD DISK BLOCKS*
PRINT BAD BLOCK STATISTICS [N] Y ↵

12e. *ADDITIONAL BAD BLOCK NUMBER (<NL>) WHEN DONE):* ↵
BITMAP SIZE: nn

13. *BITMAP ADDRESS? [default]* ↵

14. *SYSTEM DISK? [Y]* ↵

14a. *OVERLAY AREA SIZE? [default]* ↵

14b. *OVERLAY AREA ADDRESS? [default]* ↵

15. *DISK NUMBER n REMAP AREA SIZE? [default]* ↵

DG-26892T

Figure 2-1. Step Summary, Bringing Up the AOS Starter System on Blank Disk(s), from Magnetic Tape (continued)

15a. *DISK NUMBER n REMAP AREA ADDRESS? [default]* ↵

-- LOGICAL DISK CREATED

DONE!

To format other disks, type CONTINUE ↵ and return to step 10.

Running the Installer

16. Program load from tape; repeat step 8.

17. *FROM MT-0:* 3 ↵
AOS INSTALLER REV n
ENTER ALL UNITS IN LDU:

18. *DISK UNIT NAME?* DPF0 ↵ (or DPN0 ↵ or other disk unit name)

19. *DEVICE CODE?* ↵
--DISK BOOTSTRAP INSTALLED

20. *INSTALL A SYSTEM BOOTSTRAP?* Y ↵

20a. *FROM MAG TAPE (M) OR DISKETTE (D)?* M ↵

21. *FROM MT-0, FILE #:* 4 ↵

--SYSTEM BOOTSTRAP INSTALLED

22. *INSTALL A SYSTEM?* Y ↵

22a. *FROM MAG TAPE (M) OR DISKETTE (D)?* M ↵

23. *FROM MT-0, FILE #:* 5 ↵
-- SYSTEM INSTALLED
DONE!

Bringing Up the AOS Starter System

24. Now program load again, this time from the *disk*. (DPF-type disk default code is 27; DPJ-type default is code 24; fixed-head disk is code 26; for others, see Table 2-1).

24a. With an S/20, S/120, or S/280, type nnH↵. For example,
! 27H

24b. With an S/140, type 11A. The system will display a 6-digit number, after which you type 1000nn ↵. Then, type 1000nnL. For example:
! 11A *xxxxxx* 100033 ↵
! 100033L

24c. With an MV/8000, type RESET↵ and BOOT nn↵. For example,
SCP-CLI> RESET ↵
SCP-CLI> BOOT 27 ↵

24d. With hardware data switches, set switches to 1000nn. Lift RESET, then PR LOAD (PROG LOAD).
SPECIFY EACH DISK IN THE LDU

DG-26892T

Figure 2-1. Step Summary, Bringing Up the AOS Starter System on Blank Disk(s), from Magnetic Tape (continued)

25. *DISK UNIT NAME?* DPF0 ↵ (Or DPI0 ↵ or other disk unit name)

26. *DEVICE CODE?* ↵

27. *SYSTEM PATHNAME?* ↵
AOS REV n

28. *DATE (MM/DD/YY)?* 4 15 85 ↵ (current date)

29. *TIME (HH:MM:SS)?* 14 30 (current time, 24 hour clock)

30. *OVERRIDE DEFAULT SPECS [N] ?* Y ↵
MASTER LDU: ROOT1 (LDU name you gave via Disk Formatter)

31. *NUMBER OF BUFFERS IN CACHE [default]?* ↵

32. *SWAP FILE DEFINITION [default]?* ↵

33. *INITIAL LOAD [N]?* Y ↵

34. *FILENAME [MTC0:6]?* ↵
(tape file 6 is loaded)
AOS CLI REV n date time
)

Powering Up

35.) CHAR/605X/OFF/NAS ↵

36.) SUPERUSER ON ↵

37. *) DIR : ↵
*) LOAD/V @MTA0:7 ↵
.
. (CLI displays file and directory names loaded.)
.

38. *) REWIND @MTA0 ↵

39. Get and mount AOS update tape (if any); type
*) DIR : ↵
*) LOAD/V/R @MTA0:0 ↵
.
. (CLI verifies patch files loaded.)
.
*) REWIND @MTA0 ↵

40. *) SUPERUSER OFF ↵
)

41. Remove tape from unit 0, and store tapes in handy place.

DG-26892T

Figure 2-1. Step Summary, Bringing Up the AOS Starter System on Blank Disk(s), from Magnetic Tape (concluded)

What Next?

If you want to stop for a while, fine. To shut down AOS you can type BYE↵; then Y↵ when it asks if you really want to shut down. To bring it up again follow the program load procedure shown in step 24; then default each question except DATE and TIME by pressing ↵.

If this is your first system, you should now generate a tailored system. The starter system is not a general purpose timesharing system. Chapter 4 describes how you run AOSGEN to create a tailored AOS system.

If you are rebuilding/restoring a tailored system, the next next steps are to start EXEC (Chapter 5) and load user and application directories and files from backup media.

End of Chapter

Chapter 3
Bringing Up the Starter System on Blank Disks — from Diskettes

Read this chapter

- when your ECLIPSE or DESKTOP GENERATION computer has just been installed and you want to bring up AOS on it from diskettes;
- whenever you want to format a new disk, install an AOS system on it, load system files onto it, and bring up the installed system.

This chapter tells you how to execute all steps needed before you run the AOSGEN program to generate your first tailored system. The major sections are

- Powering Up
- Running the Disk Formatter
- Installing the AOS Starter System
- Bringing up the AOS Starter System
- What Next?

Powering Up

The following steps assume that CPU power is off.

1. Ready the system console. If this is a hard copy (printing) console, turn it on with the power switch under the keyboard. Press the ON LINE switch and the READY lamp will light.

   If the system console is a CRT display, turn it on (the switch is a rocker switch behind the cabinet or a push-pull knob on the front lower right corner). Then check the ON LINE lamp. If this lamp is lit, fine. If it is not lit, depress the CMD key, hold it down, and press the ON LINE key. The ON LINE lamp should light. (If there is no ON LINE key, check the ON LINE switch behind the console.)

2. If the computer has a LOCK switch or key, place it in the unlocked position. Then press the computer power switch to ON.

3. Ready your diskette unit. The full AOS system ships on two types of 8-inch diskette: high density diskettes (capacity 1.26 megabytes) and low density diskettes (capacity 0.3 megabyte). (AOS for DESKTOP GENERATION systems ships on 5-1/4 inch, 368 Kbyte diskettes.) For each type of 8-inch diskette, a diskette and hard disk unit can *share* a controller board. The steps you follow vary, depending on diskette type and whether it shares a controller with the primary hard disk. Identify and set up your unit, device code, and device name as follows:

   a. For disk model 6227-D or 6222-D (15-megabyte disk that shares a controller with a high-density diskette), remove the front panel and press the left toggle switch to the "SWP" position. This makes the diskette number 0 and the disk number 1. For model 6227-D, the device code is 33; the diskette unit name is DPI0 and the disk unit name is DPI1. For model 6222-D, the device code is 26; the diskette unit name is DPK0 and the disk unit name DPK1. Go to step 4.

b. For disk models 6098, 6100, 6101, and 6104 (shared controller, high-density diskette), remove the front panel and move the toggle switch that says DRIVE 0 SELECT to the "DSK 0/FPY 1" position. This makes the diskette number 0 and the disk number 1. For models 6098 and 6100, the device code is 33; the diskette unit name is DPI0 and the disk unit name is DPI1. For models 6101 and 6104, the device code is 26; the diskette unit name is DPK0 and the disk unit name is DPK1. Go to step 4.

c. For disk models 6045-6050 (cartridge disk subsystems) that share a controller with diskette (low density), dial 0 on the diskette unit and 1 on the disk. This makes the diskette number 0 and the disk number 1. The device code is 33. The diskette unit name is DPD0 and the disk unit name is DPD1. Go to step 4.

d. For disk models 6271, 6301, or 6336 (15-Mbyte, 38.6 Mbyte, or 71.2 Mbyte hard disk, and a 5-1/4 inch Winchester unit), the diskette and disk do not share a controller. The diskette unit is model 6267/6268 and uses 368-Kbyte 5-1/4 inch diskettes. The diskette device code is 20, and its unit name is DPM0. The hard disk device code is 26, and its unit name is DPN0 or DPN1. This diskette and disk unit are designed for DESKTOP GENERATION systems.

e. For all other disk models (e.g., 6234, 6060-series, 6160-series) or for all situations where diskette and disk are on different controllers, the diskette is usually on device code 33 or 73. Make sure the diskette has unit 0 dialed (if it has a unit select switch). A high-density diskette on code 33 is named DPI0; a high-density diskette on code 73 is named DPI10. A low-density diskette on code 33 is named DPD0; a low-density diskette on code 73 is named DPD10. Proceed to step 4.

4. If your primary disk unit is a sealed unit, press READY. Soon, the READY lamp will light.

If your primary disk uses a removable disk pack, make sure a pack is inserted in the unit. Doing this is described in the illustrated disk instruction booklet. Press DC POWER ON; press WRITE ENABLE; press DRIVE START: soon, the READY lamp will light. For either type of disk, don't touch the switches until you have generated a tailored system, tested it, and shut it down (if then).

If you want to format multiple disks, make sure they are ready as above.

5. Turn diskette unit power on, if it is not already on from step 4.

Have all the DG-supplied AOS diskettes handy and in order (number 1, 2, 3, and so on).

If you don't have a DESKTOP GENERATION Model 10/SP system, insert AOS diskette number 1 in the slot (it should slide in smoothly and come to a complete stop). Skip to Step 6.

For certain Model 10/SP systems (for example, 10/SPs with German keyboards), there may be a diskette labeled "BOOTABLE D200 EMULATOR" packed with the computer. If you find such a diskette, this is the first diskette you'll use.

5a. Insert the "BOOTABLE D200 EMULATOR" diskette: Hold the diskette with the write-enable notch up and your fingers on the label. Slide it into the unit slot. (If you have two units, use the rightmost one.) The diskette should slide in smoothly and come to a firm stop.

5b. Next to the *!* prompt, type **20H**, for example:

```
! 20H
```

This loads an "emulator" program that enables the system console to handle certain characters. The console should display the *!* prompt again.

5c. Remove the emulator diskette from the unit and replace it in its envelope. Insert AOS diskette number 1 in the unit.

You've finished powering up and can run the Disk Formatter.

Running the Disk Formatter

The Disk Formatter makes physical disks into LDUs; it can also change LDU specifications. The Disk Formatter does this by writing identifiers so that AOS will know what disk and LDU it is accessing. The Formatter can also check the disk surface also for *bad blocks* (flawed areas that won't hold information).

Running the Disk Formatter is relatively easy, but — because the Formatter checks each bit on the disk — takes more time than other system generation procedures.

Mistakes and Errors

If you type an incorrect answer to a Disk Formatter question, and have not yet pressed ⤶ to enter the answer, press the DEL key or CTRL-U to erase the wrong characters.

If you have pressed ⤶ and want to abort formatting, enter CTRL-C CTRL-A and go to step 10. If CTRL-C CTRL-A doesn't work, type the break sequence (Table 1-2) and return to step 6.

If you abort formatting by either method during surface analysis, be sure to run the entire FULL format again.

If the Disk Formatter reports a disk or other error, *make sure the disk unit is write-enabled;* if it is write-enabled, check the error message in the table near the end of the Disk Formatter chapter.

Disk Formatter Dialog

6. Next, you must program load. How you do this depends on the kind of computer you have, as follows:

6a. With an S/20, S/120, or S/280, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type nnH, where nn is the diskette device code (from step 3). For example,

```
!  33H      (S/20, S/120, S280)
```

and go to step 7.

6b. With an S/140, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type 11A. When you do, the system displays a number *xxxxxx*, which you can ignore. Following this number, type 1000nn⤶, where nn is the device code of the diskette (from step 3). Next to the second prompt, type 1000nnL. For example,

```
!  11A    xxxxxx    100033⤶    (S/140)
!  100033L
```

and go to step 7.

6c. With a DESKTOP GENERATION system (Model 10/SP, 20, or 30), the device code of your 5-1/4 inch diskette unit is 20. Type 20H next to the *!* prompt. For example:

```
!  20H
```

and go to step 7.

6d. If your computer has hardware data switches (numbered 0 (or X4/0) though 15), make sure they are set to 1000nn, where nn is the device code of the diskette (from step 3). For device code 33, set switches 0, 11, 12, 14, and 15 up; for device code 73, also set switch 10 up. Lift and release the RESET switch; lift and release the PR LOAD (PROG LOAD) switch.

7. The step(s) above program loads the disk bootstrap, SYSBOOT. SYSBOOT responds

SPECIFY EACH DISK IN THE LDU
DISK UNIT NAME?

Type the diskette unit name, from step 3. For example,

DPI0 ↵

DEVICE CODE?

8. It wants the device code of the diskette unit. If you know for sure that the diskette is connected to a nonstandard code (not 33 or 73), type the device code and ↵. In most cases, the diskette will be on a standard code, so type ↵.

↵

SYSBOOT now asks

SYSTEM PATHNAME?

9. Whenever you see this question, you can type the pathname of the program you want. In this case, you want to format your hard disk, using the Disk Formatter program. So type the Formatter filename:

DFMTR ↵

SYSBOOT now reads the Formatter file from diskette into memory. The Formatter responds

AOS DISK FORMATTER REV n

FULL FORMAT DESTROYS ANY AOS DISK STRUCTURE, PARTIAL RETAINS IT.

FULL (F) OR PARTIAL (P OR <NL>)?

10. Type

F ↵

FULL FORMAT

ENTER UNIT NAMES FOR EACH UNIT IN THE LDU (<NL> WHEN DONE):

DISK UNIT NAME?

11. The *DISK UNIT NAME* question starts a sequence of questions to identify this LDU.

If your disk and diskette share a controller, type the disk unit name given in step 3. If they *don't* share a controller, find the correct disk unit name in Table 3-1 and type this name. (If you cannot find the disk in this table, see Chapter 11, Table 11-1). For example, you might type

DPI1 ↵ (or DPI11 ↵ if diskette is on second controller)

DEVICE CODE?

11a. For any system, unless you *know* that this disk controller is connected to a nonstandard device code, press ↵ for the default.

↵

DISK UNIT NAME?

Table 3-1. Common AOS Disk Models and Names

Disk Model Number and Description	Default Device Code of Controller	Disk Number on Controller	Disk Unit Name(s)
4231A. A moving-head disk unit with a top-loading 92-Mbyte disk.	33	first (0) second (1) third (2) fourth (3)	DPE0 DPE1 DPE2 DPE3
	73	first (0) second (1) third (2) fourth (3)	DPE10 DPE11 DPE12 DPE13
6045. A dual moving-head disk unit that includes one 5-Mbyte fixed disk and one 5-Mbyte removable disk cartridge. (Note that the same controller that runs 6030 diskette units can run a 6045 disk unit.	33	first pair second pair third pair fourth pair	(Removable / Fixed) DPD0 / DPD4 DPD1 / DPD5 DPD2 / DPD6 DPD3 / DPD7
	73	first pair second pair third pair fourth pair	(Removable / Fixed) DPD10 / DPD14 DPD11 / DPD15 DPD12 / DPD16 DPD13 / DPD17
6060, 6061, 6067, and 6122; 6160 and 6161. All are moving-head disk units. The 6060, 6061, 6067, and 6122 are free-standing units with removable disks; a controller can run four units. The 6160 and 6161 are sealed units with bay-mounted fixed disks; a controller can run two units.	27	first (0) second (1) third (2) fourth (3)	DPF0 DPF1 DPF2 DPF3
	67	first (0) second (1) third (2) fourth (3)	DPF10 DPF11 DPF12 DPF13

(continues)

Table 3-1. Common AOS Disk Models and Names

Disk Model Number and Description	Default Device Code of Controller	Disk Number on Controller	Disk Unit Name(s)
A 6060 holds a 96-Mbyte disk; a 6061 holds a 190-Mbyte disk; a 6067 holds a 50-Mbyte disk; a 6122 holds a 277-Mbyte disk; a 6160 holds a	none; chosen at installation.	first (0) second (1) third (2) fourth (3)	DPF20 DPF21 DPF22 DPF23
73-Mbyte disk; a 6161 holds a 147-Mbyte disk.	none; chosen at installation.	first (0) second (1) third (2) fourth (3)	DPF30 DPF31 DPF32 DPF33
6063, 6064, 6066. A bay-mounted, fixed-head disk. The 6063 holds a 1 Mbyte disk; a 6064 holds a 2-Mbyte disk; a 6066 has two 6064 units and one controller.	26	first (0) second (1) third (2) fourth (3)	DKB0 DKB1 DKB2 DKB3
	66	first (0) second (1) third (2) fourth (3)	DKB10 DKB11 DKB12 DKB13
6070. A dual moving-head disk unit that includes one 10-Mbyte fixed disk cartridge and one 10-Mbyte removable disk cartridge.	33	first pair second pair third pair fourth pair	(Removable / Fixed) DPG0 / DPG4 DPG1 / DPG5 DPG2 / DPG6 DPG3 / DPG7
	73	first pair second pair third pair fourth pair	(Removable/ Fixed) DPG10 / DPG14 DPG11 / DPG15 DPG12 / DPG16 DPG13 / DPG17
6098, 6099, 6100, 6103. A bay-mounted moving-head unit. May have a 1.25-Mbyte diskette on the same controller. Disk holds 12.5 or 25 Mbytes.	33 73	first(0) first (0)	DPI0 DPI10
6101, 6104, 6220-D, 6222-D. A bay-mounted, sealed unit, with a 1.26-Mbyte diskette. Disk capacity is 12.5, 25, 5, or 15 Mbytes respectively. They are designed for S/20s and other microECLIPSE™ systems.	26	first (0)	DPK0 (diskette is DPK1)
	66	first (0)	DPK10 (diskette is DPK11)

(continued)

Table 3-1. Common AOS Disk Models and Names

Disk Model Number and Description	Default Device Code of Controller	Disk Number on Controller	Disk Unit Name(s)
6102, 6105, 6220, 6222. A bay-mounted, sealed unit. Disk capacity is 12.5, 25, 5, or 15 Mbytes respectively.	26 66	first (0) first (0)	DPK0 DPK10
6224. A 15-Mbyte unit that runs on the burst multiplexor channel (BMC) instead of the data channel. It's designed for S/20 and other microECLIPSE systems.	25 65	first (0) first (0)	DPL0 DPL1
6225, 6227, 6234. A fixed, bay-mounted moving-head disk unit that holds 5, 15, or 50 Mbytes respectively.	33 73	first(0) first (0)	DPI0 DPI10
6225-D, 6227-D, 6234-D. A fixed, bay-mounted moving-head disk unit that holds 5, 15, or 50 Mbytes respectively, with up to three 1.25-Mbyte diskettes on the same controller.	33 73	first(0) first (0)	DPI0 DPI10
6236 and 6237; 6239 and 6240 Each is a rack-mounted, sealed, moving-head unit with the power switch on the upper right. It has a LED display that shows the unit number and can show the current cylinder or disk fault code. A controller can run four units. Up to three units fit in a cabinet.	24 64	first (0) second (1) third (2) fourth (3) first (0) second (1) third (2) fourth (3)	DPJ0 DPJ1 DPJ2 DPJ3 DPJ10 DPJ11 DPJ12 DPJ13
A model 6236 unit holds 354 Mbytes; a model 6237 is three 6236 units in one cabinet, on one controller.	none; chosen at installation	first (0) second (1) third (2) fourth (3)	DPJ20 DPJ21 DPJ22 DPJ23
A model 6239 unit holds 592 Mbytes; a model 6240 is three 6239 units in one cabinet, on one controller.	none; chosen at installation	first (0) second (1) third (2) fourth (3)	DPJ30 DPJ31 DPJ32 DPJ33
6271. A compact, 15-Mbyte unit designed for DESKTOP GENERATION systems.	26	first (0) second (1)	DPN0 DPN1
6280. Same as model 6224, but holds 50 Mbytes instead of 15 Mbytes.	25 65	first (0) first (0)	DPL0 DPL10
6290. Two 6239 units in one cabinet on one controller.	see 6236		
6301. Same as model 6271, but holds 38.6 Mbytes instead of 15 Mbytes.	26	first (0) second (1)	DPN0 DPN1
6336. Same as model 6271, but holds 71.2 Mbytes instead of 15 Mbytes.	26	first (0) second (1)	DPN0 DPN1

(concluded)

11b. The Disk Formatter will cycle the *UNIT NAME?/DEVICE CODE?* questions until you answer ↵ to *UNIT NAME?*. This allows you to create an LDU that includes more than one physical disk. An LDU can include up to eight physical disks.

Generally, for your first system, you want a single-disk LDU — and in most cases, you will want every LDU to be a single-disk LDU. The pros and cons of multiple-disk LDUs are detailed in the Disk Formatter chapter. For now, answer the question with

↵

DO YOU WANT TO ALLOCATE A DIAGNOSTIC AREA? [N]

11c. This question lets you reserve an area on disk for later installation of DG's Advanced Diagnostic Executive System (ADES). ADES can run from a medium other than disk, but it runs much faster from disk; also, diagnostics are easier to run remotely if ADES is on disk. To use ADES, you must purchase it and have it installed on the disk by a DG field engineer. Also, ADES requires a minimum of 8000 disk blocks (3.6 Mbytes). Consequently, your disk must be larger than 25 Mbytes. Space reserved for ADES is lost for AOS file storage.

Unless you really want to install ADES and run it from disk, answer No by pressing ↵. ADES runs only from the system disk — which means you should always say No if the disk isn't the system disk. If you say no, skip to step 11e.

If you really want to reserve an area for ADES, type Y ↵. Then, the Disk Formatter asks

ENTER THE NUMBER OF BLOCKS (11610 TO 35230) THAT ARE REQUIRED. [23420]

11d. The displayed figures are octal. ADES needs at least 8000 blocks (17500 octal). The default, 23420, is 10,000 blocks. Decide on the number of disk blocks needed for the diagnostics you want installed; then type this number (octal!). The Formatter will now assume that this disk is a system disk.

DISK NUMBER 1: 00000000000 THRU n

LOGICAL DISK UNIQUE ID (1 TO 6 CHARS) [ ]?

The numbers 0 through n are the first and last logical addresses on the disk, in octal.

11e. The Disk Formatter wants a unique ID for the disk. Later, AOS will use this to keep track of the physical disks in this LDU. The ID must be 1 to 6 characters long. Any AOS filename character is legal: A through Z (uppercase and lowercase are treated the same), 0 through 9, period (.), dollar sign ($), question mark (?), and underscore (_).

Generally, use an ID that is as close as possible to the name you want to give the LDU. For example, you could use an ID of ROOT1:

ROOT1 ↵

LOGICAL DISK NAME (1 TO 31 CHARS) [ ]?

11f. The Formatter wants a name for the LDU. Later, when you boot this LDU or initialize it from the CLI, the name you enter now will be displayed.

This is the master LDU; it will be the system root directory (:). So the name you type is not important in terms of file access.

But for any disk that is not the system disk, the name you type here becomes the filename of the LDU. People can use this name just as any other directory filename. For example, if you have a lot of users and want to put some of them on a nonsystem LDU, you might name the LDU something like UDD1. Or, for a big multidisk LDU, you might choose the name

DATABASE. You can always change the LDU name later with the Disk Formatter Partial format (covered in the Disk Formatter chapter).

For optimum performance, if you have many large disks and can afford the disk space, you might want to dedicate a physical unit to the AOS SWAP file. To do this with AOSGEN, you'll give this disk's unit name as the SWAP file parameter. Then, when you bring up your tailored system, AOS will use this physical unit for swapping. A physical unit used for swapping should — ideally — be on its own controller.

For your first LDU, come up with a name with from 1 to 31 filename characters and type it. For simplicity, you can make this the same as the unique ID. For example,

ROOT1 ↵

Whatever name (and unique ID) you chose, note them for future reference.

ACCESS CONTROL LIST

USER NAME OR TEMPLATE (1 TO 15 CHARS)?

11g. A user, identified by a user name, or a group of users, identified by a template, can have different kinds of access to a logical disk. A good general-purpose username template is +, which specifies all users:

\+ ↵

PRIVILEGES (O,W,A,R,E, NEW-LINE)?

11h. The Disk Formatter wants to know which privileges to give the user(s) you just specified. There are five types of privileges: Owner, Write, Append, Read, Execute (OWARE). A NEW LINE gives the user no privileges. Execute (E) access will suffice for most disks. So type

E ↵

USER NAME OR TEMPLATE (1 TO 15 CHARS)?

11i. The Disk Formatter will cycle the *USER NAME/PRIVILEGES* questions, allowing you to give very specific user/access information, until you answer ↵ to *USER NAME...* You can change the access control list of any LDU when AOS is running — so the answers you give in steps 11e and 11f are not critical. Generally, answer +↵ to question 11e, answer E↵ to question 11f, and answer ↵ to this question. Later, if needed, you can change access to the LDU. So type

↵

SURFACE ANALYSIS? [N]

12. This step starts a series on surface analysis for this LDU. (The value in brackets is the default, which the Formatter will use if you simply answer ↵.)

For each new disk, you should answer Y↵ to this question. During analysis, the Formatter writes a pattern to each 16-bit word on the disk and reads it back. This destroys all existing information on the disk — but a new disk has no information. Answer yes by typing

Y ↵

DISK NUMBER?

12a. The Formatter wants the number of the disk to analyze. For a single-disk LDU, press

↓

YOU MAY RUN UP TO FIVE (5) PATTERNS...HOW MANY WOULD YOU LIKE TO RUN?

12b. To ensure the validity of the disk structure, the Formatter writes to the disk a variety of bit patterns, reads from the disk, and compares the results. When the results differ, AOS marks the block as bad. You can specify up to five bit patterns, and we recommend that you do.

Each pattern takes up to 50 minutes, depending on the disk model, as shown in Table 3-2. So five patterns may take a while. To run five patterns, type

5 ↓

ANALYZING DISK #n
--RUNNING PATTERN nnnnnn

The Formatter runs the patterns you specified, one by one. If it finds too many bad blocks, it aborts; this usually means that the disk heads are misaligned or that the disk needs hardware formatting. But in most cases, it's simply a matter of waiting.

When the Formatter has finished the patterns, it describes the bad blocks on the disk.

12c. If the Formatter found no bad blocks, it says *0 BAD DISK BLOCKS*. Skip to step 12d, *ADDITIONAL BAD BLOCK NUMBER?* question.

If it found any bad blocks, it says

n BAD DISK BLOCKS
PRINT BAD BLOCK STATISTICS? [N]

Table 3-2. Surface Analysis Times for Popular Disks

Disk Model	Capacity (megabytes)	Time per Test Pattern
6060	96	13 minutes
6061	190	26 minutes
6063	1	3 minutes
6064, 6066	2	6 minutes per disk unit
6122	277	26 minutes
6160	73	11 minutes (approximately)
6161	147	22 minutes (approximately)
6227, 6271	15	15 minutes (approximately)
6234	50	8 minutes (approximately)
6236	354	50 minutes (approximately)

12d. The Formatter is asking if you want to see the bad block statistics. These statistics may be useful, so answer yes.

Y ↓

The Formatter now displays the bad block statistics on the console. If there are bad blocks, you might want to note them. There should be few, if any, bad blocks on a new disk.

ADDITIONAL BAD BLOCK NUMBER (<NL>) WHEN DONE):

12e. You have no additional bad blocks to enter, so press

)

BITMAP SIZE: n

The bitmap is a system table that describes which blocks are in use and which are free for data storage.

BITMAP ADDRESS? [default]

13. Select the default by pressing

)

SYSTEM DISK? [Y]

14. This step starts a series that determines whether and where an AOS system will reside on the disk.

Your first LDU must be a system disk, so answer) for the default:

)

OVERLAY AREA SIZE? [default]

14a. Choose the default area size by pressing

)

OVERLAY AREA ADDRESS? [default]

14b. Choose the default address by pressing

)

DISK NUMBER n REMAP AREA SIZE? [default]

15. Choose the default remap area size by pressing

)

DISK NUMBER n REMAP AREA ADDRESS? [default]

15a. Choose the default remap area address by pressing

)

--LOGICAL DISK CREATED

DONE!

Congratulations! You've formatted an LDU as a system disk. It will rarely — if ever — need full formatting again. If this LDU may be run as a nonmaster LDU, we suggest that you note the date, LDU ID and unit name, and any bad block information — and attach the note to the disk unit (or, for a removable disk, the plastic housing).

The Formatter is done. If you have other new disks, someone must format them into LDUs before they can be used. If you know what LDU configuration(s) you want, you might want to create the LDU(s) now — while you're familiar with the procedure. To do it now, type P) or press the CONTINUE panel switch and return to step 10. To format a disk *not* described in this chapter, see the Disk Formatter chapter.

If you don't want to format other disks now, proceed and install the AOS starter system.

Installing the AOS Starter System

The Installer program installs an AOS system from diskette 3 onto an LDU. (AOS must reside on disk before it can run.)

The steps described here work the same way for a tailored system tape that you will create later: the Installer simply installs the tailored system, not the starter system.

Mistakes and Errors

If you make a mistake, you can handle it the same way as with the Disk Formatter.

If the Installer reports a disk or other error, check the error message in the table near the end of the Installer chapter.

If the Installer stops with an *ABORT* message, return to step 16 and start the Installer dialog again.

Running the Installer

16. The *!* or the *SCP-CLI>* prompt is showing on the console. Repeat step 6 to program load from diskette again (device code 20, 26, 33 or 73).

    SPECIFY EACH DISK IN THE LDU
    DISK UNIT NAME?

17. As before, type the diskette unit name (step 3); for example

    DPI0 ⤶ (or DPM0 ⤶)

    DEVICE CODE?

18. Choose the default device code (⤶) or type it as before.

    SYSTEM PATHNAME?

19. Type the filename of the Installer program (INSTL):

    INSTL ⤶

    SYSBOOT now loads the Installer into memory. The Installer responds

    AOS INSTALLER REV n

    ENTER ALL UNITS IN LDU:

    DISK UNIT NAME?

20. Now, *open the diskette door, remove diskette 1, and insert diskette 2.* (Diskette 2 has SYSBOOT on it in a form that can be installed on the disk.)

21. Type the name of your disk. If this disk shares a controller with the diskette, use the disk unit name given in step 3. If the disk is on its own controller, type the name from Table 3-1. For example

    DPI1 ⤶

    DEVICE CODE?

22. Unless you know the disk is on a nonstandard code, press ⤶ to select the default code.

    ⤶

    --DISK BOOTSTRAP INSTALLED

    INSTALL A SYSTEM BOOTSTRAP?

23. You must install a system bootstrap the first time you install a system; you will rarely need to do so thereafter. The first disk in a system LDU must have a disk bootstrap and a system bootstrap; it usually has a system as well.

Y ↓

FROM MAG TAPE (M) OR DISKETTE (D)?

23a. Type D ↓.

DISKETTE UNIT NAME?

23b. Type the same diskette unit name you have been giving all along; for example

DPI0 ↓ (or DPM0 ↓)

The Installer now reads SYSBOOT from the diskette and writes it to the disk. Then it responds

-- SYSTEM BOOTSTRAP INSTALLED

INSTALL A SYSTEM?

24. Now, remove diskette 2 from the diskette unit, and insert diskette 3. This diskette has the starter system on it.

25. You want to install a system, so type Y ↓:

Y ↓

FROM MAG TAPE (M) OR DISKETTE (D)?

25a. Type D ↓.

DISKETTE UNIT NAME?

25b. Type the diskette unit name; for example

DPI0 ↓

The Installer now reads the AOS system from diskette 3 and copies it to the disk. Then it responds

-- SYSTEM INSTALLED

DONE!

You've installed the needed bootstraps and an AOS system on your LDU. Now you can bring up the AOS system.

Bringing Up the AOS Starter System

26. If your disk and diskette share a controller, it's now time to change the disk back to unit 0 and the diskette to unit 1. Reverse the action of step 3; e.g., move toggle switch to NRM; or move toggle switch to "DSK 0/FPY 1"; or dial 0 on the disk and 1 on the diskette.

27. Now to program load again, this time from the *disk*. If the disk/diskette share a controller, the disk device code is generally 26 or 33. For other disks: the DPF-type device code is 27; the DPJ-type device code is 24; for other disks, see Table 3-1. Follow these steps:

27a. With an S/20, S/120, or S/280, the system console displays a *!* prompt. Next to the prompt, type nnH, where nn is the device code. For example,

! 33H (S/20, S/120, S280)

and go to step 28.

27b. With an S/140, or the system console displays a *!* prompt. Next to the prompt, type 11A. The system will display a 6-digit number, after which you type 1000nn↵, where nn is the device code. Next to the second prompt, type 1000nnL. For example,

! 11A *xxxxxx* 100033 ↵ (S/140)
! 100033L

and go to step 28.

27c. For a DESKTOP GENERATION system, the hard disk device code is 26 and the disk unit name of the primary hard disk is DPN0. So type 26H.

27d. If your computer has hardware data switches (numbered 0 (or X4/0) though 15), make sure they are set to 1000nn, where nn is the device code. For device code 27, set switches 0, 11, 13, 14, and 15; for device code 24, set switches 0, 11, and 13 up; for device code 26, set switches 0, 11, 13, and 14 up; for device code 33, set switches 0, 11, 12, 14, and 15 up. Lift and release the RESET switch; lift and release the PR LOAD (PROG LOAD) switch.

28. Your program load steps load SYSBOOT from the *disk*. SYSBOOT asks

SPECIFY EACH DISK IN THE LDU

DISK UNIT NAME?

Type the name of the hard disk unit. This name ends in 0 (shown in Table 3-1). For example,

DPF0 ↵ (or DPN0 ↵ or DPJ0 ↵)

DEVICE CODE?

29. As before, press ↵ for the default device code:

↵

SYSTEM PATHNAME?

30. Having installed a system, you can press ↵ to load and execute it. Press

↵

AOS REV n

DATE (MM/DD/YY)?

31. Enter the date as numbers for month, day, and year. Spaces or slashes can separate each number. For example, for April 15, 1985, you'd type

4 15 85 ↵

TIME (HH:MM:SS)?

32. Enter the time, based on a 24-hour clock, in hours, minutes, and seconds. (Minutes and seconds are optional. If you omit them, the system sets each to 0.) Use spaces or colons to separate each number pair. For example, for 2:30 p.m., you'd type

14 30 ↵

OVERRIDE DEFAULT SPECS [N] ?

33. SPECS means the parameters in the system specification file created during AOSGEN. For your first system, you must answer yes, so type

Y ⤶

MASTER LDU: ROOT1 (logical disk name you gave to Formatter)

NUMBER OF BUFFERS IN CACHE [default]?

34. Press

⤶

SWAP FILE DEFINITION [default]?

35. Press

⤶

INITIAL LOAD [N]?

36. In an *INITIAL LOAD,* the system loads the CLI and other needed files onto the LDU. These files must be loaded the first time you bring up the starter system; they need not be loaded again unless — later on — you want to load a new revision of AOS. Answer yes by typing

Y ⤶

FILENAME [MTA0:6]?

37. Remove the diskette (diskette 3) from the diskette unit. Insert diskette 4 (this has the first dump file, with the CLI and other essential files).

38. Type the diskette unit name. If the disk/diskette share a controller, this has changed from unit 0 to unit 1. It is now DPI1 or DPK1 (high-density diskettes) or DPD1 (low-density diskettes or DPM0 (5-1/4 inch diskettes). For example,

DPI1 ⤶

39. AOS now reads the diskette and copies files from it to the disk and displays

AOS CLI REV n date time

)

Congratulations! You've brought up AOS and its CLI. The CLI's prompt,), tells you that it is ready for a command.

(If you get a *FATAL ERROR 25* message, a needed file wasn't loaded. Perhaps you forgot to answer Y⤶ to the *INITIAL LOAD* question. In any case, run Emergency Shutdown (ESD). (See Chapter 6 for information on running ESD.) Then, return to step 27 to try again. For a description of errors by numeric code, see Appendix A.)

The CLI is far more sophisticated than any program you've been using. It has many commands and fine error handling; it has a HELP feature that you can use after you've loaded the next diskette. You can interrupt executing CLI commands by typing CTRL-C CTRL-A. As always, you can delete characters with DEL and delete bad lines with CTRL-U; and you can type CTRL-S to suspend display and CTRL-Q to restore it (useful for reading long files on a display console.)

If your system console is a CRT display, change the console characteristics by typing

) CHAR/605X/OFF/NAS ⤶
)

40. Now you can load the rest of the diskettes with all the other AOS programs. First, turn on SUPERUSER to provide write access.

```
) SUPERUSER    ON ⤶
*)
```

With AOS running, you are subject to file access controls — which means that you might get an error message if you tried to load files into the root directory. SUPERUSER allows you to bypass all file access controls. The asterisk before the) prompt means that SUPERUSER is on.

41. Remove the diskette from its unit. Get the next DG-supplied AOS diskette and insert it in the unit. Load via the diskette device name (@DPI1 for high density, @DPD1 for low density, @DPM0 for 5-1/4 inch diskettes, or @DPK1). For example,

```
*) DIR   : ⤶

*) LOAD/V   @DPI1 ⤶

  .
  .
  .
```

The CLI verifies (/V switch) the directory and filenames copied from diskette by displaying their names on the system console. The whole directory structure on the diskettes is copied, creating directory :UTIL (with utilities), directory :SYSGEN (for system generation), and directory :HELP (for help).

All these directories (along with files like DFMTR, the Disk Formatter) are copied into the *root* directory. The root directory's name is : (colon), so the pathname to any of these newly-created directories is

```
:directory-name
```

for example, :UTIL.

After all these files have been loaded, the CLI SUPERUSER prompt returns.

```
*)
```

Repeat this step (step 41) until you have loaded the last AOS diskette.

42. Now, if you received an AOS *update* diskette, get it. If not, skip this step. Updates have revision numbers with the last two digits not 00. For example, 7.01 is an update number. Mount the diskette and type

```
*) LOAD/V/R    @DPI1 ⤶    (or LOAD/V/R   @DPK1 ⤶ or @DPD1 ⤶ or @DPM0 ⤶)
.
.  (CLI verifies load of update notice and patch files.)
.
```

This puts AOS update and patch files on the LDU, for access later on.

43. Turn SUPERUSER off by typing

```
*) SUPERUSER   OFF ⤶
)
```

The) is the standard CLI prompt.

Well done! You've powered up, formatted at least one LDU, installed an AOS system on it, brought up AOS, and loaded all files you need to generate your tailored system.

If you're interested in the files on the diskettes, see Table 3-3; all of these files are now on your LDU. The LDU also contains directory :PATCH, with current patch files (if you loaded an update diskette).

Figure 3-1 is a summary of all the steps you've taken — from turning on the system console to removing the last diskette.

Table 3-3. AOS System Diskettes File Format

Diskette Number	Program Filename	File Contents
1	SYSBOOT	Disk bootstrap; a program that can load other programs from disk or diskette.
	FIXUP	Disk Fixer utility, which finds and optionally corrects disk file errors if abnormal AOS shutdown occurs. SYSBOOT loads this program into memory and executes it after you type FIXUP ⤶ to the *SYSTEM PATHNAME?* query.
	DFMTR	Disk Formatter utility, which formats physical disks into LDUs. SYSBOOT loads DFMTR into memory after you answer DFMTR⤶ to the *SYSTEM PATHNAME* query.
	INSTL	Installer utility, which installs an AOS system on an LDU. SYSBOOT loads INSTL into memory after you type INSTL⤶ to the *SYSTEM PATHNAME?* query.
2	SYSBOOT	The system bootstrap program loads an AOS system or other program into memory *from disk,* then executes the program. The Installer installs SYSBOOT on an LDU.
3	AOS System	On the DG-supplied diskette, this is the AOS starter system. On a system diskette you make, it is a tailored AOS system. INSTL installs this file on an LDU.
4	First Dump File	The CLI and other system program files, including the system GHOST, peripheral manager (PMGR), and copies of programs on diskettes 1 through 3. AOS copies the contents of this file into the LDU root directory when you specify INITIAL LOAD.
5-n	Second Dump File	You copy the contents of this file onto disk with the CLI LOAD command, as part of the initial load procedure. This file contains nearly all AOS support software files, including the - Disk File Editor (DEDIT) - DISPLAY file display - Error message (ERMES) and message object files (.OBs) - EXEC and PREDITOR - HELP directory and files - LABEL tape labeler - Link - Library File Editor (LFE) - Macroassembler (MASM) - Process Enviroment Display (PED) - Release Notice (latest on software) - SED and SPEED text editors - System macros - Utility program symbol table files (.ST) - SYSGEN directory with AOSGEN system generation program and libraries

Powering Up

1. Have system console ON and ON LINE.

2. Unlock computer, turn computer power on.

3. Ready diskette (procedure varies):

   a. With disk model 6227-D, remove panel and push switch to SWP.

   b. With 6098, 6100, 6101, or 6104, remove panel and push switch to "DSK1-FPY0."

   c. With 6045-50, dial 0 on diskette and 1 on disk.

   d. With a 6271, 6301, or 6336 disk, the disk device code is 26 and the unit name is DPN0. The diskette device code is 20 and its unit name is DPM0.

   e. With disk and diskette on different controllers, set diskette to 0 (if there is a switch).

4. Turn disk unit on; have disk write-enabled and READY.

5. Turn diskette unit on, if not on from step 4. If computer is not a DESKTOP GENERATION Model 10/SP, insert AOS diskette number 1 and go to step 6.

5a. With a DESKTOP GENERATION Model 10/SP, insert "BOOTABLE D200 EMULATOR" diskette.

5b. *!* 20H (Type 20H next to *!* prompt.)

5c. When *!* prompt returns, remove EMULATOR diskette; insert AOS diskette number 1.

Running the Disk Formatter

6. The next step is to program load from diskette (device code 20, 26, 33 or, on second controller, 73) as follows:

6a. With an S/20, S/120, or S/280, type nnH, where nn is the device code. For example

    ! 33H (S/20, S/120, or S/280)

6b. With an S/140, type 11A. The system will display a 6-digit number, after which you type 1000nn ↓, where nn is the device code. Then, type 1000nnL. For example

    ! 11A *xxxxxx* 100033 ↓ (S/140)
    ! 100033L

6c. With a DESKTOP GENERATION system, type 20H.

6d. With hardware data switches, make sure they are set to 1000nn, where nn is the device code. (For device code 33, set switches 0, 11, 12, 14, and 15 up; for device code 73, set switches 0, 10, 11, 12, 14, and 15 up. Lift and release RESET, then lift and release PROG LOAD (PR LOAD).

    SPECIFY EACH DISK IN THE LDU

7. *DISK UNIT NAME* DPI0 ↓ (or DPD0 ↓, or other diskette unit name)

8. *DEVICE CODE?* ↓

DG-26893T

Figure 3-1. Step Summary, Bringing Up the AOS Starter System on Blank Disk(s) from Diskettes (continues)

9. *SYSTEM PATHNAME?* DFMTR ↵

AOS DISK FORMATTER REV n

FULL FORMAT DESTROYS ANY AOS DISK STRUCTURE,
PARTIAL FORMAT RETAINS IT.

10. *FULL (F) OR PARTIAL (P OR <NL>)?* F ↵

FULL FORMAT

ENTER UNIT NAMES FOR EACH UNIT IN THE LDU (<NL> WHEN DONE):

11. *DISK UNIT NAME?* DPI1 ↵ (or DPF0 ↵, or other disk unit name)

11a. *DEVICE CODE?* ↵

11b. *DISK UNIT NAME?* ↵

11c. *DO YOU WANT TO ALLOCATE A DIAGNOSTIC AREA? [N]*

Unless you want to allot 8,000 blocks for later installation of the Advanced Diagnostic system, press ↵ and skip to step 11e. To reserve an area, type Y ↵.

11d. *ENTER THE NUMBER OF BLOCKS (11610 TO 35230)...[23420] ?* ↵

DISK NUMBER 1: 00000000000 THRU n

11e. *LOGICAL DISK UNIQUE ID (1 TO 6 CHARS) [ ]?* ROOT1 ↵ (Valid disk ID)

11f. *LOGICAL DISK NAME (1 TO 31 CHARS) [ ]?* ROOT1 ↵ (Valid LDU name)

11g. *USER NAME OR TEMPLATE (1 TO 15 CHARS)?* + ↵

11h. *PRIVILEGES (O, W, A, R, E, NEW-LINE) ?* E ↵

11i. *USER NAME OR TEMPLATE (1 TO 15 CHARS)?* ↵

12. *SURFACE ANALYSIS? [N]* Y ↵

12a. *DISK NUMBER?* ↵

12b. *YOU MAY RUN UP TO FIVE (5) PATTERNS...HOW MANY WOULD*
YOU LIKE TO RUN? 5 ↵ (Choose pattern(s))

ANALYZING DISK #n

--RUNNING PATTERN nnnnn

12c. If it found no bad blocks, go to 12e.

12d. *n BAD DISK BLOCKS*
PRINT BAD BLOCK STATISTICS? [N] Y ↵

Note bad blocks (except for a model 6214 disk).

12e. *ADDITIONAL BAD BLOCK NUMBER (<NL>) WHEN DONE):* ↵
BITMAP SIZE: nn

13. *BITMAP ADDRESS? [default]* ↵

14. *SYSTEM DISK? [Y]* ↵

DG-26893T

Figure 3-1. Step Summary, Bringing Up the AOS Starter System on Blank Disk(s) from Diskettes (continued)

14a. *OVERLAY AREA SIZE? [default]* ↵

14b. *OVERLAY AREA ADDRESS? [default]* ↵

15. *DISK NUMBER n REMAP AREA SIZE? [default]* ↵

15a. *DISK NUMBER n REMAP AREA ADDRESS? [default]* ↵
-- LOGICAL DISK CREATED
DONE!
To format other disks, type C ↵ or press the CONTINUE panel switch and return to step 10.

Running the Installer

16. Program load from diskette; repeat step 6.
SPECIFY EACH DISK IN THE LDU

17. *DISK UNIT NAME?* DPI0 ↵ (or DPD0 ↵, or DPM0 ↵, or other diskette unit name)

18. *DEVICE CODE?* ↵

19. *SYSTEM PATHNAME?* INSTL ↵
AOS INSTALLER REV n
ENTER ALL UNITS IN LDU:

20. Remove diskette 1, insert diskette 2.

21. Type disk unit name (e.g., DPI1 ↵, or DPD1 ↵, or DPN0 ↵)

22. *DEVICE CODE?* ↵
-- DISK BOOTSTRAP INSTALLED

23. *INSTALL A SYSTEM BOOTSTRAP?* Y ↵

23a. *FROM MAG TAPE (M) OR DISKETTE (D)?* D ↵

23b. *DISKETTE UNIT NAME?* DPI0 ↵ (or DPD0 ↵, or DPM0 ↵)
-- SYSTEM BOOTSTRAP INSTALLED

24. Remove diskette 2; insert diskette 3.

25. *INSTALL A SYSTEM?* Y ↵

25a. *FROM MAG TAPE (M) OR DISKETTE (D)?* D ↵

25b. *DISKETTE UNIT NAME?* DPI0 ↵ (or DPD0 ↵, or DPM0 ↵)
-- SYSTEM INSTALLED
DONE!

DG-26893T

Figure 3-1. Step Summary, Bringing Up the AOS Starter System on Blank Disk(s) from Diskettes (continued)

Bringing Up the AOS Starter System

26. Reverse any action you took in step 3:

    a. With 6227-D, set switch to NRM and cover panel.

    b. With 6098, 6100, 6101, or 6104, set switch to "DSK0-FPY1" and cover panel.

    c. With 6045-50, dial 1 on diskette and 0 on disk.

27. Program load from disk (with shared controller, device code is 36 or 33). For DPF-type disk, device code is 27; for DPJ-type disk, device code is 24; all others are in Table 3-1.

27a. With an S/20, S/120, or S/280, type nnH, where nn is the device code. For example

! 33H (S/20, S/120, or S/280)

27b. With an S/140, type 11A, and then type 1000nn ↵, where nn is the device code. Next, type 1000nnL. For example

! 1A *xxxxxx* 100033 ↵ (S/140)
! 100033L

27c. With a DESKTOP GENERATION system, type 26H.

27d. With hardware data switches, set switches to 1000nn, where nn is the disk device code. Lift and release RESET, then lift and release PROG LOAD (PR LOAD).

SPECIFY EACH DISK IN THE LDU

28. *DISK UNIT NAME?* DPI0 ↵ (Or other disk unit name)

29. *DEVICE CODE?* ↵

30. *SYSTEM PATHNAME?* ↵

    AOS REV n

31. *DATE (MM/DD/YY) ?* 4 15 85 ↵ (current date)

32. *TIME (HH:MM:SS) ?* 14 30 ↵ (current time, 24-hour clock)

33. *OVERRIDE DEFAULT SPECS [N] ?* Y ↵

    MASTER LDU: ROOT1 (LDU name you gave to Disk Formatter)

34. *NUMBER OF BUFFERS IN CACHE [default]?* ↵

35. *SWAP FILE DEFINITION [default]?* ↵

36. *INITIAL LOAD [N]?* Y ↵

37. Remove diskette 3 and insert diskette 4.

38. *FILENAME [MTA0:6]?* DPI1 ↵ (Or DPD1 ↵, or DPM0 ↵ for low density)

    .
    . (first dump file is loaded)
    .

    AOS CLI REV n date time

39.) CHAR/605X/OFF/NAS ↵

DG-26893T

Figure 3-1. Step Summary, Bringing Up the AOS Starter System on Blank Disk(s) from Diskettes (continued)

40.) SUPERUSER ON ↵

41. *) DIR :↵
*) LOAD/V/R @DPI1 ↵ (Or DPK1 ↵, or @DPD1 ↵, or @DPM0 ↵ for low density)
.
. (CLI displays file and directory names loaded.)
.
Repeat step 41 until all diskettes have been loaded.

42. Get and insert AOS update diskette (if any). Type
*) LOAD/V/R @DPI1 ↵ (Or @DPK1 ↵, or @DPM0 ↵ for low density)
.
. (CLI verifies patch files loaded.)
.

43. *) SUPERUSER OFF ↵
)
Store all diskettes.

DG-26893T

Figure 3-1. Step Summary, Bringing Up the AOS Starter System on Blank Disk(s) from Diskettes (concluded)

What Next?

If you want to stop for a while, fine. To shut down AOS, you can type BYE↵, then Y↵ when the system asks if you really want to shut down. To bring it up again, follow the program load procedure described in step 27; then default each question except the DATE and TIME by pressing ↵.

If this is your first system, you should now generate a tailored system. Go to Chapter 4.

If you are rebuilding/restoring a tailored system, the next next steps are to start EXEC (Chapter 5) and load user and application directories and files from backup media.

End of Chapter

Chapter 4
Generating a Tailored AOS System

Read this chapter

- when you want to generate your first tailored AOS system;
- whenever you want to generate a new AOS system.

This chapter tells you how to run AOSGEN, the AOS system generation program. First, it describes bootstrapping (in case you shut down AOS earlier); then it explains AOSGEN and leads you through a AOSGEN session — devices, parameters, naming and building. Finally, it shows you how to test, patch, and optionally install the tailored system. The major sections are:

- Bootstrapping (Power On)
- About the AOSGEN Program
- AOSGEN Session
- Install Patches
- Testing the New System
- Installing the Terminal Emulator (DESKTOP GENERATION Model 10/SP Only)
- Making a Tailored System Tape
- Installing the Tailored System
- Generating Other Tailored Systems
- Supporting Synchronous Devices
- Creating AOS PREGEN Diskettes for DESKTOP GENERATION Systems
- What Next?

AOSGEN is really very easy — the only tricky part is specifying the device that handles user consoles. If you know about this device, the only other things you need to specify are a disk, tape unit (optional), probably a line printer, and the system parameters. The whole thing may take only a few minutes. This chapter is large because it attempts to give all the details. Figures 4-5 through 4-6 are simple AOSGEN sessions.

There must be an AOS system running before you can run AOSGEN. This can be either the DG-supplied starter system or your own previously generated tailored system.

Bootstrapping (Power On)

Read this section only if AOS is currently not running; if it is running (shown by the CLI) prompt), skip to the next section.

Before you can generate a tailored system with AOSGEN, you must program load the AOS starter system from disk. How you do this depends on what type of computer you have:

- If you have an S/20, S/120, S/280, or DESKTOP GENERATION system, go to step 1.
- If you have an S/140, go to step 2.
- If you have an MV/8000, go to step 3.
- If your computer has data switches, go to step 4.

1. (S/20, S/120, S/280, and DESKTOP GENERATION systems only) The system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type nnH, where nn is the device code. For example

   ! 27H   or 26H   (S/20, S/120, S/280, or DESKTOP GENERATION systems)

   and go to step 5.

2. (S/140 only) The system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type 11A. The system will display a 6-digit number, after which you type 1000nn⤶, where nn is the device code. Now re-enter 1000nn. For example

   ! 11A   *xxxxxx*   100033 ⤶   (S/140)
   ! 100033L

   and go to step 5.

3. (MV/8000 only) The system console displays an *SCP-CLI>* prompt. Next to the prompt, type

   SCP-CLI>   RESET ⤶
   SCP-CLI>   BOOT 27 ⤶   (or 24 ⤶ for model 6236 disks;
   or 33 ⤶ for model 6234 disks)

   and go to step 5.

4. If your computer has hardware data switches (numbered 0 (or X4/0) through 15), make sure they are set to 1000nn, where nn is the device code. For code 27, set switches 0, 11, 13, 14 and 15 up, the others down. For code 33, set switches 0, 11, 12, 14, and 15 up, the others down. Lift and release the RESET switch; lift and release the PR LOAD (PROG LOAD) switch.

5. Answer the following questions:

   SPECIFY EACH DISK IN THE LDU

   DISK UNIT NAME?   DPF0 ⤶   (or DPJ0 ⤶ for model 6236 disks;
   or DPI0 ⤶ for model 6234 disks)

   DEVICE CODE?   ⤶

   SYSTEM PATHNAME?   ⤶

   DATE (MM/DD/YY)?   4 15 85 ⤶   (enter current date)
   TIME (HH:MM:SS)?   15 20 ⤶   (current time)
   OVERRIDE DEFAULT SPECS? [N]   ⤶
   MASTER LDU: name

   (a pause occurs)

   AOS CLI REV n date time
)

The master CLI process is running. Now you can run AOSGEN to generate a tailored system.

About the AOSGEN Program

AOSGEN is a utility program that creates an operating system tailored for the hardware you specify; it also allows you to select some parameters for its own operation. You can specify CPU model, add devices, review current specifications, change parameters, list devices, create a specification file (a source file from which AOSGEN can build a system), or build a system (instruct AOSGEN to build specification information into an operating system).

The first time you run AOSGEN, you'll be starting from scratch and adding all devices. Later, you'll probably use an existing AOSGEN spec file for a base — and you'll be adding or editing only a few devices.

AOSGEN is interactive, with extensive help messages. You can get help from it at any point by typing ?↓.

AOSGEN includes the following files:

AOSGEN.PR	The program file.
AOSGEN.OL	The overlay file.
AOSGEN.ST	The symbol table file.
AOSGEN.DATA	An AOSGEN data file.
AOSGEN.QUES	An AOSGEN data file.

Directory :SYSGEN — created during the AOS initial load — holds all AOSGEN files and needed system generation libraries. So directory :SYSGEN is ready for system generation.

Files AOSGEN Creates

AOSGEN creates several files, including temporary files, in directory :SYSGEN. The temporary files carry a .TMP filename suffix. Table 4-1 lists all files that AOSGEN creates.

Table 4-1. Files AOSGEN Creates

Filename	Comments
sys.SY	This is the tailored operating system file, ready to run. AOSGEN creates it after you issue the Build command. sys is the name you give the system with the Name command.
sys.ST	This is the operating system symbol table file, which may be useful to DG personnel if the new system doesn't run perfectly. AOSGEN creates it along with the tailored system file.
sys.CSF	The Customer Specification (spec) File. It describes this system's hardware and software parameters. This file is in ASCII so you can print or type it. AOSGEN creates this spec file in response to your Spec or Build command.
sys.SSF	The System Specification File. It describes each system in a form that AOSGEN can read. AOSGEN creates this spec file in response to your Spec or Build command.
?pid.AOSGEN.CURR.TMP	A temporary data file where AOSGEN keeps information during a AOSGEN session. pid is the 3-digit AOSGEN process ID; e.g., 003. AOSGEN deletes this file when it terminates normally. If AOSGEN aborts, this file remains.
sys.CONFIG.pid.TMP	A temporary configuration file, in a format designed for a system build. The pid is the three-digit Process ID.
sys.KS_IN.pid.TMP	A temporary input file that contains the CLI macros AOSGEN needs to build a system.
sys.KS_OUT.pid.TMP	A temporary output file. If the new system does not run properly, you should TYPE this file and check for error messages. If you use the /SAVE switch on the X AOSGEN command line, AOSGEN saves this file and the two preceding .TMP files. Otherwise, when you run AOSGEN interactively, AOSGEN will ask you if you wish to save the .TMP files when you build a system.

Execute AOSGEN as follows for your first system, or with the switches of your choice for subsequent systems:

*) X AOSGEN ↵

INITIALIZING AOSGEN DATA FILES...
WELCOME TO AOSGEN — TYPE H FOR HELP

ENTER A COMMAND:

Getting Help

Type

H ↵

AOSGEN responds

The following are legal commands to AOSGEN:

A — Add a system device
B — Build a system
C — List current configuration
D — Delete a system device
E — Edit a system device
L — List a system device
M — Change system model
N — Change system name
P — Change system parameters
Q — Quit this session
S — Create a spec file
V — Verify current configuration

(To receive further information about any of these commands, type H <SPACE> X, where X is any of the above commands.)

Note —
Typing "?" in response to any question will list possible responses. ...

ENTER A COMMAND:

Get help on Add:

H A ↵

A - ADD A CONTROLLER TO THE CURRENT SYSTEM

Routine asks for the name of the controller....

Give the Add command and ask for help again:

ENTER A COMMAND: A ↵
NAME OF DEVICE TO BE ADDED? ? ↵

ALM	*ASLM*	*ASLM1*	*ASLM2*	*ASLM3*	*BBU*	*BMC*	*CON1*	*CRA*
CRA1	*DKB*	*DKB1*	*DKB2*	*DKB3*	*DKB4*	*DKB5*	*DKB6*	*DKB7*
DPD	*DPD1*	*DPE*	*DPE1*	*DPF1*	*DPF2*	*DPF3*	*DPF4*	*DPF5*
DPF6	*DPF7*	*DPG*	*DPG1*	*DPI*	*DPI1*	*DPJ*	*DPJ1*	*DPJ2*
DPJ3	*DPJ4*	*DPJ5*	*DPJ6*	*DPJ7*	*DPK*	*DPK1*	*DPL*	*DPL4*
DPM	*DPM1*	*DPN*	*DPN1*	*IAC*	*IAC1*	*IAC2*	*IAC3*	*IAC4*
IAC5	*IAC6*	*IAC7*	*IAP*	*ISC*	*ISC1*	*ISC2*	*ISC3*	*LPA*
LPA1	*LPB*	*LPB1*	*LPB2*	*LPB3*	*LPB4*	*LPB5*	*LPB6*	*LPB7*
LPC	*LPC1*	*LPD*	*LPD1*	*MCA*	*MCA1*	*MTA1*	*MTB*	*MTB1*
MTC	*MTC1*	*MTC2*	*MTC3*	*PLA*	*PLA1*	*SDCU*	*SDCU1*	*SDCU2*
SDCU3	*SLM*	*TPA*	*TPA1*	*TRA*	*TRA1*	*ULM*		

Typing ?↵ in response to any AOSGEN question gives you a list of legal answers.

The peripheral names used by AOSGEN are *controller* names. For example, if you have two disk units on the first DPJ controller, their unit names are DPJ0 and DPJ1 — but the controller name is DPJ. This list of devices reflects the AOSGEN default system, which includes a DPF and MTA controller and a system console (CON0). Since the system already includes these controllers, they can't be added, so their names are not shown in this Add HELP list. The controller names and meanings are explained in each section.

For a description of any device, type the List command, then the device name; e.g., L↲, BBU↲. AOSGEN will display the current (or default) device specifications. It will also show how many data channel slots on the A or B map the device is using (these are used by tapes, LPB and LPD-type printers, and DPI and some other disk units). Get out of the Add command by pressing

↲

ENTER A COMMAND:

Model of CPU

To specify the Model of CPU, use the M command:

M ↲
ENTER NEW MODEL [C/350]:

Type the model of the computer on which the new system will run. Or, for a C/350, press ↲ to select the default. For example,

↲
ENTER A COMMAND:

After you describe the computer, you can add or edit devices. The default system has one DPF disk and one MTA tape controller, and the system console (CON0). You may have this disk and this tape controller, and/or other device controllers.

Most systems also have *user* consoles, attached to their own device. There may be one or more line printers, communications devices, and a backup battery. Each of these is described in its own section.

You can add or edit devices in any order, but let's do the disks first.

Disks

An AOS system can support up to eight DKB, DPF, or DPJ controllers, if the system has a burst multiplexor channel (or up to four DKB and DPF controllers if it doesn't have a BMC), and up to two of any other model disk controller. All the controller types and models are shown in Table 4-2.

Some CPU models support a burst multiplexor channel (BMC) which provides a high-speed direct communications pathway between main memory and high-speed disks such as DKB, DPF, DPJ, and DPL-type disks. If your system has a BMC, this is a convenient time to add it. To add the BMC, use the Add command:

ENTER A COMMAND: A ↲
NAME OF DEVICE TO BE ADDED? BMC ↲

The *default* system supports a DPF disk controller on device code 27. Disk unit names on the first DPF controller are DPF0, DPF1, DPF2, and DPF3.

If you want DPF disk controller support, you need not add a DPF to the default system. Check the DPF definition by typing E↲ and DPF↲; then run though the dialog. If you want the new system to support other controllers, add them by typing A↲ and the controller name from Table 4-2. Take the default device code unless you know that a controller is connected to a nonstandard code.

If you *don't* want the new system to support a DPF controller, delete the DPF (type D↲, then DPF↲). Then add your primary disk controller (DPJ or DPI), by typing A↲, then DPJ↲ or DPI↲. The DPJ controller supports four units, named DPJ0, DPJ1, DPJ2, and DPJ3. The DPI controller supports up to two units, DPI0 and DPI1. Take the default device code unless you are sure the controller is connected to a nonstandard device code. For example, if your primary controller is DPJ, type

ENTER A COMMAND: D ⤶
NAME OF DEVICE TO BE DELETED: DPF ⤶
DPF HAS BEEN DELETED
ENTER A COMMAND: A ⤶
NAME OF DEVICE TO BE ADDED: DPJ ⤶
DEVICE CODE [24]: ⤶
ENTER A COMMAND:

If you want the new system to support another controller, add it using the A⤶ command and controller name from Table 4-2. Take the default device code, if any, unless you know the controller is connected to a nonstandard code.

During processing, if AOS encounters a condition severe enough to make continued processing dangerous, AOS will display an error message on the system console and request that you take a core memory dump. An AOS core dump records the contents of all main memory on magnetic tape or diskette — it is useful for debugging should the error condition recur. If you are describing a disk controller to AOSGEN and that controller has a diskette, AOSGEN will ask

SYSTEM DUMP DEVICE [N]:

If you want AOS to use this diskette when you take a core memory dump, type Y ⤶, otherwise take the default by typing

⤶

You may only select one system dump device. The default dump device is MTA, unit 0.

Table 4-2. Disk Controller Names and Device Codes

Disk Description	Controller Name	Default Device Code
DPF-series. Free standing with removable packs (6060, 6061, 6067, 6122); or rack-mounted and sealed (6160, 6161). Capacity ranges from 47 to 277 Mbytes. A 6160, or 6161 supports two units; others support four units.	DPF (1st) DPF1 (2nd) DPF2 (3rd) DPF3 (4th) DPF4 (5th) DPF5 (6th) DPF6 (7th) DPF7 (8th)	27 67 None; chosen at installation. None, chosen at installation. None; chosen at installation. None; chosen at installation. None; chosen at installation. None; chosen at installation.
DPJ-series, model 6236, sealed, rack-mounted, with LED digit display. Capacity is 354 Mbytes. A controller supports four units.	DPJ (1st) DPJ1 (2nd) DPJ2 (3rd) DPJ3 (4th) DPJ4 (5th) DPJ5 (6th) DPJ6 (7th) DPJ7 (8th)	24 64 None; chosen at installation. None; chosen at installation. None; chosen at installation. None; chosen at installation. None; chosen at installation. None; chosen at installation.
DPI-series, rack-mounted. Model 6234, 6234-D, 6227, 6277-D, 6225, 6225-D, 6098, 6099, 6100 or 6103 each runs a hard, sealed disk, range 50 down to 5 Mbytes. Model 6097 runs one or two 1.26-Mbyte diskettes.	DPI0 (1st) DPI10 (1st)	33 73

(continues)

Table 4-2. Disk Controller Names and Device Codes

Disk Description	Controller Name	Default Device Code
Fixed-head disk, models 6063,6064, 6066.	DKB0 (1st) DKB1 (2nd) DKB2-DKB7	26 66 None; chosen at installation.
Two-disk rack-mounted unit; top disk removable, bottom disk sealed, 5 Mbytes per disk; total 10 Mbytes. Controller can also run 0.3-Mbyte diskettes. Models 6045-6050.	DPD (1st) DPD1 (2nd)	33 73
Two-disk unit as above, but 10 Mbytes per disk; total 20 Mbytes; no diskettes. Model 6070.	DPG (1st) DPG1 (2nd)	33 73
Diskette (0.3 Mbyte). Model 6030.	DPD (1st) DPD1 (2nd)	33 73
DPK series, rack-mounted, models 6222-D, 6101, 6104, 6220-D hard-disk unit with a 1.26-Mbyte diskette, for microECLIPSE computers. (6102, 6105, 6220, 6222 do not have a diskette unit.)	DPK (1st) DPK10 (1st)	26 66
DPL series, Model 6280 or 6224, either a 50- or 15-Mbyte sealed disk on the Burst Multiplexor Channel (BMC).	DPL (1st) DPL1 (2nd)	25 65
DPN, compact, sealed disks used in DESKTOP GENERATION systems. Model 6271 holds 15 Mbytes; model 6301 holds 38.6 Mbytes; model 6336 holds 71.2 Mbytes.	DPN	26
DPM Model 6267 and 6268 5-1/4 inch diskette, 368 Kbyte, used in DESKTOP GENERATION and other systems.	DPM DPM1	20 60

(concluded)

The Current (C) Command

The C command displays the current system configuration. You may want to use it to check after you add a controller. For example,

ENTER A COMMAND: C ⤶

CURRENT SYSTEM: none *DEFAULT SYSTEM: <system defaults>*

SYSTEM PARAMETERS:

MODEL:	*C/350*
SWAP:	*2000*
CACHE SIZE:	*0*
FREQUENCY:	*10*
SYNC BUFFERS	*1*
ACCESS CONTROL ENABLED:	*YES*
DEMAND PAGING ENABLED:	*NO*

CURRENT SYSTEM DEVICES:

...(all current devices)...

OPTIONAL SYSTEM DEVICES:

...(all devices not part of configuration)

NUMBER OF DATA CHANNEL A MAP SLOTS AVAILABLE: n
NUMBER OF DATA CHANNEL B MAP SLOTS AVAILABLE: n

ENTER A COMMAND:

The new system has no name yet, but its parameters and devices are shown. If, at any point, you want to change a device spec, use the Edit command. And you can verify the system — to see if it is valid — with the V command.

With the exception of S/20, C/30, and DESKTOP GENERATION systems, which have one data channel map, most CPUs have five data channel maps, but AOSGEN describes only the A and B maps. The A and B maps have 32 1K-word slots each. Each LPB or LPD-type line printer takes two slots on the A map. If your system does not have a BMC, DKB controllers take five slots and DPF controllers take nine slots. DKB, DPF, DPG, and DPI controllers use the B map; DPK, DPM, DPN, DPD and DPE disks use the A map. DPK controllers take two slots, DPM controllers take six slots, and all other non-BMC controllers take five slots.

Each MTA, MTB, or MTC tape controller takes five slots. A tape controller will use more slots if you specify a larger MAX BYTE TRANSFER buffer than the AOSGEN default, and a tape controller will use fewer slots if you specify a smaller MAX BYTE TRANSER buffer than the AOSGEN default. (MTA tape controllers use the A map; MTB and MTC tape controllers use the B map.)

When you use the List command to list a printer, tape or disk device, AOSGEN tells you which data channel map the device uses and the number of slots the device will use. If the device uses the B map, but there are not enough slots available, the device will use the A map instead.

Every time you Add, Delete, or Edit a device, AOSGEN makes sure that the number of map slots taken does not exceed the maximum number of slots available. If too many map slots are taken, AOSGEN will tell you so.

Tapes

The default system supports an MTA tape controller with default options. If the current specification according to the C command describes the tape controller(s) you want supported, fine; you may want to edit the controller to see if its options (described below) are the ones desired.

If you don't want the new system to support the current controller(s), delete it (them), then add the controller(s) and options that you want.

There are several types of tape controller, as follows.

MTA — The MTA unit model number is 6021; its density is 800 bpi. The controller can handle up to eight units, named MTA0, MTA1, MTA2, ..., MTA7.

MTB — The MTB unit model number is 6026; its density can be 800 or 1600 bpi. A unit has a panel DENSITY switch. The controller can handle up to eight units, named MTB0, MTB1, MTB2, ..., MTB7.

MTC — The MTC unit model number is 6123 or 6125 (reels side by side); its density is 1600 bpi. MTC is also the controller name for a cartridge tape unit, model 6230, 6231, or 6270 (these are 6400 bpi.) On a DESKTOP GENERATION computer, there can be only one MTC unit. On ECLIPSE computers, an MTC controller can handle up to four units, named MTC0, MTC1, MTC2, MTC3.

Be sure to type the correct controller name here. The starter system recognizes type MTA or MTB. But for a *tailored* system to support any type well, you must specify the proper type. For example:

```
                    MTC ↵  (or MTB ↵)
DEVICE CODE [22]:
```

Press ↵ to specify the default device code. For an MTA controller, skip to the *SYSTEM DUMP DEVICE [Y]:* query. For an MTC controller, skip to the next *MAX BYTE TRANSFER:* query.

↵
DEFAULT DENSITY [default]:

You can select ADM, a valid bpi number, or NC as the default density. The meanings are as follows:

ADM Automatic Density Matching. When a unit on this controller reads a tape, the new system will try to match controller density to tape density — regardless of the DENSITY switch setting (if any). On writes, an MTB will use the tape panel DENSITY switch setting (800 or 1600 bpi). Users can override the write default by specifying a valid density with the /DENSITY= switch on CLI commands.

n (1600 or 800 bpi). Use density n as default. This overrides the DENSITY switch, if any. Automatic density matching will not occur; but users can override the default n with the the /DENSITY= switch in CLI commands.

NC No change from current density. (The current density is the density last specified with the /DENSITY switch in a CLI command. But if no density has been specified on an MTB, the current density is that selected with the DENSITY panel switch. On reads or writes, only the current density will be used. Users can override the current density with the /DENSITY= switch in CLI commands.

The default answer (ADM) is the best general-purpose choice for its type of controller. We recommend it unless you know that you want another choice. To take it, press

↵
MAX BYTE TRANSFER [nK]:

This selects the maximum size of the buffer used for tape I/O. Valid answers are 2K (2,048 bytes), 4K, 6K, and so on through 32K for MTB tape controllers and 2K, 4K, 6K and so on through 62K for MTC tape controllers. For both MTB and MTC controllers, the 8K default is a good general-purpose choice; it's big enough for efficient reads and writes, yet not so big that it may slow down the program doing the I/O.

Be careful about selecting a large maximum size buffer. The larger the buffer size, the more data channel maps slots you will use. If you select more map slots than are available — particularly on S/20, C/30, or DESKTOP GENERATION systems, which have one map — your system may not come up. Therefore, we recommend a buffer size of 8K for systems with only one map. Also note that tape models 6230, 6231, and 6270 do not support a buffer size greater than 8K.

You can select any buffer size up to the maximum with a /BUFFERSIZE= switch on a LOAD or DUMP command; or you can take the default size by omitting this switch. The same buffer size used to write a tape must be used to read it back.

Unless you have a good reason for doing otherwise, take the default by typing

↵

SYSTEM DUMP DEVICE: [N]:

During processing, if AOS encounters a condition severe enough to make continued processing dangerous, AOS will display an error message on the system console and request that you take a core memory dump. An AOS memory dump records the contents of all main memory on magnetic tape or diskette — it is useful for debugging should the error condition recur. If you want AOS to use this tape controller for the memory dump type Y ↵. Otherwise, take the default by typing

↵

You may only select one system dump device. The default dump device is MTA.

ENTER A COMMAND:

If you want the new system to support another mag tape controller, type A↵, then the controller name: MTA1↵, MTB1↵, or MTC1↵. In most cases, choose the default device code. Unit names on the second controller are MTx10, MTx11, MTx12, and so on.

After the last tape controller, you might want to review the specification (C command) before you proceed.

System Console

The system console name is CON0. It is included in the default system as a CRT3 (DASHER D2, D200, D210, D211, or other D200-compatible CRT). If this is the kind of console you want supported, fine; skip to the next section.

If you want the new system to support a *different type* of system console, then you need to edit the CON0 spec. The other types of console are hard copy, TTY, CRT1 (DG 4010I CRT console), CRT2 (DG 6012 CRT console), CRT4 (another DG-compatible CRT console) and CRT6 (graphics terminal, DASHER D400, D410, D450, D460, or G300). For example, to specify a hard-copy system console

```
ENTER A COMMAND:    E ↵
  NAME OF DEVICE TO EDIT?    CON0 ↵
  CONSOLE TYPE [CRT3]:    TTY ↵
  INPUT BUFFER BYTE LENGTH [96]    ↵
  OUTPUT BUFFER BYTE LENGTH [128]    ↵
  CHARACTERISTIC WORD 0 [STANDARD]    ↵
  CHARACTERISTIC WORD 1 [STANDARD]    ↵
  CHARACTERISTIC WORD 2
        LINES PER PAGE [STANDARD]    ↵
        CHARS PER LINE [STANDARD]    ↵
        CHARACTERISTIC WORD 3 [STANDARD]:    ↵
ENTER A COMMAND:
```

The *BUFFER BYTE LENGTH* questions relate to buffers used by the system to communicate with the device; you can usually default them. *CHARACTERISTICS WORD 0, 1,* and *3* relate to things like echoing, lower-to-uppercase conversion, modem interfaces, and so on. For the system console, you will usually want to take the defaults on these — as above. The same applies to *LINES PER PAGE* and *CHARACTERS PER LINE.*

If you *do* want to specify nondefault console values, Table 4-3 later on, describes the mnemonics you can use.

User Consoles (ALMs, ASLMs, IACs, and ULMs)

You don't specify user consoles directly; instead, you describe the device that handles them. Then you describe the consoles connected to the *lines* of the device.

User consoles are handled by one or more of the following devices:

Asynchronous Line Multiplexors (ALMs)
Asynchronous/Synchronous Line Multiplexors (ASLMs)
Intelligent Asynchronous Controllers (IACs)
Universal Line Multiplexors (ULMs)

You'll tell AOSGEN which kind you have, then describe the console lines attached to it.

NOTE: If the consoles are not labeled, someone should label them — preferably with the console name, @CONn. You can use sticky-backed tape for this. If you don't know a console's name, you can figure it out from the line number, as described in each section below. Labeling the consoles will make things a lot easier later on.

Ranges of Lines

You can describe console lines individually or in groups. AOSGEN will ask questions about each entry. If you choose a group of lines, all consoles in the group will be treated the same way, even if the consoles are different. After you describe one or more groups of lines, AOSGEN asks if you want to describe more lines. This allows you to proceed sequentially to describe them all. You need not specify groups sequentially, but we do so here for clarity.

To specify individual lines, separate the entries with one or more spaces. To specify a range of lines, separate the first and last number with a space-dash-space. For example,

0 12	specifies lines 0 and 12
0 - 12	specifies lines 0 through 12

Letter-Quality Printers

If your system has one or more letter-quality printers, you must identify these to AOSGEN as user consoles, type TTY. For each line connected to a letter-quality printer, specify TTY and default all characteristic words. In the initialization word, specify a small input buffer and maximum output buffer, data rate of 1200 baud, 8 data bits, and 1 stop bit. An example of letter-quality printer specification appears in Figures 4-1 through 4-4.

DESKTOP GENERATION System Printers and Plotter

Several economical printers and one plotter are available with AOS — primarily for use with DG DESKTOP GENERATION systems, and usable with either systems.

The printer models are the 4434 dot-matrix printer (up to 160 characters per line); the 4518 letter-quality printer (up to 203 characters); and the 4433 printer (up to 233 characters). The plotter — which uses two pens — is model 4435.

You identify each of these devices to AOSGEN as a *terminal* of the default console type (CRT3). If you're generating a system for a desktop Model 10/SP, and want to support a printer/plotter on the CPU printer port, add it via AOSGEN as CON1. If generating a system to support a printer/plotter *not* on the 10/SP printer port, add the printer/plotter as a console line on the asynchronous multiplexor (ALM, ASLM, IAC, or ULM) as if it were a user terminal. For any printer, you can take all AOSGEN defaults for console type. For a 4435 plotter, specify 7 data bits, even parity, 9600 baud.

The devicename of the printer/plotter will be @CONn when the new system runs. To have EXEC manage the printer, you'll need to create a queue of type print and open the queue. The UP macro for the new system will then need to start the device on the queue and continue the queue — just as for a line printer. If you want maximum characters per line and lines per page to differ from EXEC defaults (80 and 63), UP should set these before continuing the printer.

ALMs

An ALM has one or more 8- or 16-line asynchronous multiplexor (mux) boards. An ALM with an 8-line mux is called an ALM-8. This ALM can handle either local or modem lines; it can be part (or all) of an ALM. The standard device device code for the ALM is 34.

With three 16-line muxes, an ALM can support console lines 0 through 47. Each mux board is hardware strapped to the next, which may or may not provide a contiguous sequence of lines from one mux board to the next. The sequence is contiguous only if all lines in the preceding mux board are used.

ALM Dialog

To add an ALM, Add a device, then specify ALM:

A ↵
NAME OF DEVICE TO BE ADDED: ALM ↵

I/O PROCESSOR [NONE]: ? ↵
DCU50, IOP, NONE

Some CPU models support I/O processors that handle the data input and output from low-speed devices such as CRT consoles and hard-copy terminals, DASHER printer terminals, card readers, plotters, and paper tape. A DCU/50 data control unit is one of these processors. If your system has a DCU/50, you should type DCU50 ↵. M/600 and MV/8000 systems include an IOP, which is another type of I/O processor. If your system is an M/600 or MV/8000, you should type IOP ↵. If your system does not have an I/O processor, type ↵ to select the default.

LINES [??]:

Now you need to describe the lines on this ALM — individually or in groups.

For lines attached to the first mux board in the ALM, lines are numbered 0 through n, where n is the number of consoles up to a maximum of 15 or 7 (depending on the mux board). For lines attached to the second and subsequent mux boards, lines are numbered from the total number of lines in preceding mux boards. For example, if the first mux has 16 lines (even if only 12 are used), the number of the first line on the second mux board is 16.

ALM Console Names

When the new system runs, each console on the ALM has the console name of

CON (line-number + 2)

For example, the name of the console on line 5 is CON7, and the name of the console on line 17 is CON19.

From mux board to mux board, the console names are not contiguous unless you have consoles attached to all previous mux lines. For example, even if you have only 12 consoles attached to the first mux (producing the console names CON2 through CON14), the name of the console attached to the first line on the second mux is CON18.

ALM Console Lines Example

The following example, Figure 4-1, tries to show you how to handle all common ALM line-console configurations. It assumes that your ALM has a 16-line mux followed by an 8-line mux, and that your system has a DCU/50.

It also assumes that the ALM lines are connected as follows:

- Lines 0 - 2 are attached to local CRTs;
- Line 3 is attached to a local TTY;
- Lines 4 - 12 are attached to local graphics CRTs;
- Line 13 is attached to a letter-quality printer;
- Lines 16 - 17 (lines 0 and 1 on second mux) are attached to modems.

```
ENTER A COMMAND:     A ⤶
  NAME OF DEVICE TO BE ADDED:     ALM ⤶
  I/O PROCESSOR [NONE]:     DCU50 ⤶
  LINES [??]:     0 - 2 ⤶
  CONSOLE TYPE [CRT3]:     ⤶
  INPUT BUFFER BYTE LENGTH [96]:     ⤶
  OUTPUT BUFFER BYTE LENGTH [128]:     ⤶
  CHARACTERISTIC WORD 0 [STANDARD]:     ⤶
  CHARACTERISTIC WORD 1 [STANDARD]:     ⤶
  CHARACTERISTIC WORD 2
    LINES PER PAGE [STANDARD]:     ⤶
    CHARACTERS PER LINE [STANDARD]     ⤶
  CHARACTERISTIC WORD 3 [STANDARD]:     ⤶
  INITIALIZATION WORD [STANDARD]:     ⤶

  DO YOU WANT TO ADD ANOTHER GROUP OF LINES [N]?     Y ⤶

  LINES [??]:     3 ⤶
  CONSOLE TYPE [CRT3]:     TTY ⤶
  INPUT BUFFER ....     ⤶

  (default the buffer and characteristic questions)

  INITIALIZATION WORD [STANDARD]:     ⤶

  DO YOU WANT TO ADD ANOTHER GROUP OF LINES [N]?     Y ⤶

  LINES [??]:     4 - 12 ⤶
  CONSOLE TYPE [CRT3]:     CRT6 ⤶
  INPUT BUFFER ....     ⤶

  (default the buffer and characteristic questions)

  INITIALIZATION WORD [STANDARD]:     ⤶

  DO YOU WANT TO ADD ANOTHER GROUP OF LINES [N]?     Y ⤶

  LINES [??]     13 ⤶
  CONSOLE TYPE [CRT3]     TTY ⤶
  INPUT BUFFER BYTE LENGTH [96]:     32 ⤶
  OUTPUT BUFFER BYTE LENGTH [128]:     255 ⤶
  CHARACTERISTIC WORD 0 [STANDARD]     ⤶
  CHARACTERISTIC WORD 1 [STANDARD]     ⤶
  CHARACTERISTIC WORD 2
    LINES PER PAGE [STANDARD] :     ⤶
    CHARACTERS PER LINE [STANDARD] :     ⤶
  CHARACTERISTIC WORD 3 [STANDARD]:     ⤶
  INITIALIZATION WORD [STANDARD]:     ?COD3   ?CLK1   ?STP0 ⤶

  DO YOU WANT TO ADD ANOTHER GROUP OF LINES [N]?     Y ⤶
  LINES [??]     16 17 ⤶
  CONSOLE TYPE [CRT3]:     ⤶
  INPUT BUFFER ....     ⤶
  OUTPUT BUFFER ...     ⤶
```

DG-26894T

Figure 4-1. Sample ALM Dialog, with ALM-16 and ALM-8 (continues)

```
CHARACTERISTIC WORD 0 [STANDARD]:      ? ↵
?M<ST SFF 8BT RAF RAT REC NAS EOL UCO MRI...>
CHARACTERISTIC WORD 0 [STANDARD]:      ?MST   ?MEOC ↵
CHARACTERISTIC WORD 1 [STANDARD]:      ? ↵
?TTY, ?CRT<3,6>, ?M<ULC PM NRM MOD DT<0-3> TO ....>

CHARACTERISTIC WORD 1 [STANDARD]:      ?CRT3   ?MMOD   ?MULC   ?MWRP ↵
CHARACTERISTIC WORD 2
  LINES PER PAGE [STANDARD]:    ↵
  CHARS PER LINE [STANDARD]:    ↵
CHARACTERISTIC WORD 3 [STANDARD]:    ↵
INITIALIZATION WORD [STANDARD]:    ?PAR0   ?COD3   ?STP0   ?CLK3 ↵

DO YOU WANT TO ENTER ANOTHER GROUP OF LINES [N]?    ↵

ENTER A COMMAND:
```

DG-26894T

Figure 4-1. Sample ALM Dialog, with ALM-16 and ALM-8 (concluded)

In Figure 4-1, for ALM, we described lines 0-2, hard-copy line 3, graphics lines 4-12, and specified the letter-quality printer on line 13 (with 1200 baud; on our system, CLK1 is jumpered to 1200 baud).

Next we specified the modem lines: lines 16 and 17 (0 and 1 on the second ALM mux). We checked legal responses to CHARACTERISTIC WORD 0 and took the default. Proceeding to CHARACTERISTIC WORD 1, we checked and found the characteristics we wanted: ?MMOD for the modem lines; and we entered these and the other standard characteristics. All these characteristics are described in Table 4-3.

When you specify one nondefault characteristic, you must specify *all* the ones you want. The angle brackets, <>, are notation abbreviations; interpret them as if they were parentheses in an arithmetic expression. We chose the default for CHARACTERISTIC WORD 2 and 3.

For the initialization word, we specifed clock 3 (which, in our system, is jumpered to 300 baud) for the modem lines; and we gave the other characteristics. Then we chose no more lines to add; and we were finished with the ALM.

When the new system runs, the console names on lines 0-2 will be CON2-CON4; on line 3 the name will be CON5; on lines 4-12 CON6-CON14; on line 13 the letter-quality printer will be CON15; and on lines 16 and 17 they will be CON18-CON19.

The line-specification procedure would have been much simpler without the modem lines; we'd have simply defaulted all the characteristics and initialization words.

ASLMs

Each Asynchronous/Synchronous Line Multiplexor is one circuit board that supports both asynchronous and synchronous lines. (On S/20s and DG DESKTOP GENERATION systems, it's called a USAM.) This section outlines how to describe asynchronous lines on an ASLM. If you wish to describe synchronous ASLM lines you will find additional information in a later section of this chapter.

To describe ASLM lines, specify a base line number for each ASLM, then describe lines 0 to n, where n is the number of lines attached to consoles (up to a maximum of 3).

ASLM Dialog

To add an ASLM, use the Add command, then specify ASLM:

```
                    A ⤶
NAME OF DEVICE TO BE ADDED:    ASLM ⤶
DEVICE CODE [34]:
```

The default device code for ASLM is 34; for ASLM1 it is 74. Unless you know that this ASLM is connected to a nonstandard device code, type ⤶ to select the default. ASLM2 and ASLM3 do not have a default device code — you should enter the nonstandard device code that the ASLM board is connected to.

```
                    ⤶
ASLM BASE LINE NUMBER [0]:
```

You can select any valid base line number for this ASLM. The number must be 0 or divisible by 4. The names of its consoles will be CON (base-line-number + lines-number-on-this-ASLM + 2). For the first ASLM, you might choose the default:

```
                    ⤶
LINES [??]:
```

Now you need to describe the lines on this ASLM — individually, or in groups. The line numbers you enter are specific to this ASLM — ranging from 0 through 3. In this example, let's say you want to describe line 0:

```
                    0 ⤶
ARE THESE LINES SYNC OR ASYNC [ASYNC]:
```

Since you are describing asynchronous lines, type ⤶ to select the default.

ASLM Console Lines Example

The following example, Figure 4-2, tries to show you how to handle all common line-console configurations. It assumes that lines are arranged as follows:

- ASLM line 0 is attached to a local graphics CRT
- ASLM line 1 is attached to a letter-quality printer
- ASLM lines 2 and 3 are attached to modems

```
ENTER A COMMAND:  A ↵

  NAME OF DEVICE TO BE ADDED:   ASLM ↵
  DEVICE CODE [34]:   ↵
  ASLM BASE LINE NUMBER [0]:   ↵
  LINES [??]:   0 ↵
  ARE THESE LINES SYNC OR ASYNC [ASYNC]:   ↵
  CONSOLE TYPE [CRT3]:   CRT6 ↵
  INPUT BUFFER LENGTH [96]:   ↵
  OUTPUT BUFFER LENGTH [128]:   ↵
  CHARACTERISTIC WORD 0 [STANDARD]:   ↵
  CHARACTERISTIC WORD 1 [STANDARD]:   ↵
  CHARACTERISTIC WORD 2
    LINES PER PAGE [STANDARD]:   ↵
    CHARACTERS PER LINE [STANDARD]:   ↵
  CHARACTERISTIC WORD 3 [STANDARD]:   ↵
  INITIALIZATION WORD [STANDARD]:   ↵

  DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]:   Y ↵

  LINES [??]:   1 ↵
  ARE THESE LINES SYNC OR ASYNC [ASYNC]:   ↵
  CONSOLE TYPE [CRT3]:   TTY ↵
  INPUT BUFFER LENGTH [96]:   32 ↵
  OUTPUT BUFFER LENGTH [128]:   255 ↵
  CHARACTERISTIC WORD 0 [STANDARD]:   ↵
  CHARACTERISTIC WORD 1 [STANDARD]:   ↵
  CHARACTERISTIC WORD 2
    LINES PER PAGE [STANDARD]:   ↵
    CHARACTERS PER LINE [STANDARD]:   ↵
  CHARACTERISTIC WORD 3 [STANDARD]:   ↵
  INITIALIZATION WORD [STANDARD]:   ?CLN8  ?CR12H  ?CS10 ↵

DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]:  Y ↵

  LINES [??]:   2 - 3 ↵
  ARE THESE LINES SYNC OR ASYNC [ASYNC]:   ↵
  CONSOLE TYPE [CRT3]:   ↵
  INPUT BUFFER LENGTH [96]:   ↵
  OUTPUT BUFFER LENGTH [128]:   ↵
  CHARACTERISTIC WORD 0 [STANDARD]:   ↵
  CHARACTERISTIC WORD 1 [STANDARD]:  ?CRT3  ?MMOD  ?MULC  ?MWRP ↵
  CHARACTERISTIC WORD 2
    LINES PER PAGE [STANDARD]:   ↵
    CHARACTERS PER LINE [STANDARD]:   ↵
  CHARACTERISTIC WORD 3 [STANDARD]:   ↵
  INITIALIZATION WORD [STANDARD]:   ?CPR0  ?CR300  ?CS10  ?CLN8 ↵

  DO YOU WANT TO ENTER ANOTHER GROUP OF LINES [N]:   ↵

ENTER A COMMAND:
```

DG-26895T

Figure 4-2. Sample of ASLM Dialog

IACs

Each IAC is one circuit board and has its own device code. The sequence of lines from board to board is established during AOSGEN. You specify a base line number for each board; then describe the lines from 0 to n, where n is the number of lines attached to consoles (up to a maximum of 15 or 7, depending on whether the IAC has 16 or 8 lines). At a small price in memory, you can specify *all lines* attached to an IAC, even if all lines aren't currently attached to consoles. An IAC-16 can handle up to 16 local lines; an IAC-8 can handle a total of 8 *local and modem* lines.

Note that you must specify your IAC types and their sequence correctly — the new system may fail if you don't.

IAC Dialog

To add an IAC, Add a device, then specify IAC:

A ↵
NAME OF DEVICE TO BE ADDED: IAC ↵
DEVICE CODE [65]:

The default device code for IAC is 65; for IAC1 it is 50; for IAC2 it is 51, and so on. Unless you know that this IAC is connected to a nonstandard device code, type ↵ to select the default:

↵
IAC BASE LINE NUMBER [0]:

You can select any valid base line number for this IAC. The number must be 0 or divisible by 8. The names of its consoles will depend on the number, as described below. For the first IAC, you might choose the default:

↵

and go on to the next question.

For the second IAC and each subsequent IAC, you could type the number

base-line-of-preceding-IAC
+
total-number-of-lines-on-preceding IAC (8 or 16)

For example, 16 ↵.

You need not make the base lines sequential. For example, you might choose to make your modem base line 48. But each base line must be larger than the preceding IAC's base line and each must be divisible by 8. (AOSGEN imposes this to ensure unique console names, so that your multiuser system will work properly.)

IAC DEVICE TYPE [??]:

If this is a 16-line IAC, type 16 ↵; if it is an 8-line IAC, type 8 ↵. For example

16 ↵

If you type 8 ↵, AOSGEN asks about split baud rates. Answer ↵ for the default unless the console on this line has an attached slave printer. Then AOSGEN will respond

LINES [??]:

Now you need to describe the lines on this IAC — individually, or in groups. The line numbers you enter are specific to this IAC — ranging from 0 through 15 or 0 through 7, depending on the IAC.

IAC Console Names

When the new system runs, each console on an IAC will have the console name of

CON (base-line-number + line-number-on-this-IAC + 2)

For example, assume that the first IAC is an IAC-16, for which you have specified base line number 0 and lines 0-12. The console names on these lines will be CON2 through CON14. The second IAC, IAC1, is an IAC-8. You give IAC1 the base line number of 16; then you describe line 1 of IAC1. The console on line 1 (second line) of IAC1 will be CON(16+1+2), or

CON19

IAC Console Lines Example

The following example, Figure 4-3, tries to show you how to handle all common line-console configurations. It assumes that your first IAC is an IAC-16 and your second IAC (IAC1) is an IAC-8.

And it assumes that the lines are arranged as follows:

- IAC lines 0 - 2 are attached to local CRTs;
- IAC line 3 is attached to a local TTY;
- IAC lines 4 - 12 are attached to local graphics CRTs;
- IAC line 13 is attached to a letter-quality printer;
- IAC lines 14 -15 are not used;
- IAC1 lines 0 and 1 are attached to modems.

```
ENTER A COMMAND:     A ↵
  NAME OF DEVICE TO BE ADDED:     IAC ↵
  DEVICE CODE [65]:     ↵
  IAC BASE LINE NUMBER [0]:     ↵
  IAC DEVICE TYPE [??]:     16 ↵
  LINES [??]:     0 - 2 ↵
  CONSOLE TYPE [CRT3]:     ↵
  INPUT BUFFER BYTE LENGTH [96]:     ↵
  OUTPUT BUFFER BYTE LENGTH [128]:     ↵
  CHARACTERISTIC WORD 0 [STANDARD]:     ↵
  CHARACTERISTIC WORD 1 [STANDARD]:     ↵
  CHARACTERISTIC WORD 2
    LINES PER PAGE [STANDARD]:     ↵
    CHARACTERS PER LINE [STANDARD]     ↵
  CHARACTERISTIC WORD 3 [STANDARD]:     ↵
  INITIALIZATION WORD [STANDARD]:     ↵

  DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]     Y ↵

  LINES [??]     3 ↵
  CONSOLE TYPE [CRT3]:     TTY ↵
  INPUT BUFFER ....     ↵

  (default the buffer and characteristic questions)

  INITIALIZATION WORD [STANDARD]:     ↵

  DO YOU WANT TO ADD ANOTHER GROUP OF LINES [N]?     Y ↵

  LINES [??]     4 - 12 ↵
  CONSOLE TYPE [CRT3]:     CRT6 ↵
  INPUT BUFFER ....     ↵

  (default the buffer and characteristic questions)

  INITIALIZATION WORD [STANDARD]:     ↵

  DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]     Y ↵
  LINES [??]     13 ↵
  CONSOLE TYPE [CRT3]     TTY ↵
  INPUT BUFFER BYTE LENGTH [96]:     32 ↵
  OUTPUT BUFFER BYTE LENGTH [128]:     255 ↵
  CHARACTERISTIC WORD 0 [STANDARD]     ↵
  CHARACTERISTIC WORD 1 [STANDARD]     ↵
  CHARACTERISTIC WORD 2
    LINES PER PAGE [STANDARD] :     ↵
    CHARACTERS PER LINE [STANDARD] :     ↵
  CHARACTERISTIC WORD 3 [STANDARD]:     ↵
  INITIALIZATION WORD [STANDARD]:     ?CLN8  ?CR12H  ?CS10 ↵
```

DG-26696T

Figure 4-3. Sample IAC Dialog, with IAC-16 and IAC-8 (continues)

```
    DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]       ↵
  ENTER A COMMAND:     A ↵
    NAME OF DEVICE TO BE ADDED:     IAC1 ↵
    DEVICE CODE? [50]:     ↵
    IAC BASE LINE NUMBER [0]:     16 ↵
    IAC DEVICE TYPE [??]     8 ↵
    IAC SPLIT BAUD RATE [NONE]:     ↵
    LINES [??]:     0   1 ↵
    CONSOLE TYPE [CRT3]:     ↵
    INPUT BUFFER ....     ↵
    OUTPUT BUFFER ...     ↵
    CHARACTERISTIC WORD 0 [STANDARD]:     ? ↵
    ?M<ST SFF 8BT RAF RAT REC NAS EOL UCO MRI...>

    CHARACTERISTIC WORD 0 [STANDARD]:     ?MMRI   ?MST   ?MEOC ↵
    CHARACTERISTIC WORD 1 [STANDARD]:     ? ↵
    ?TTY, ?CRT<3,6>, ?M<ULC PM NRM MOD DT<0-3> TO ....>

    CHARACTERISTIC WORD 1 [STANDARD]:     ?CRT3   ?MMOD   ?MULC   ?MWRP ↵
    CHARACTERISTIC WORD 2
      LINES PER PAGE [STANDARD]:     ↵
      CHARS PER LINE [STANDARD]:     ↵
    INITIALIZATION WORD [STANDARD]:     ?CPR0   ?CR300   ?CS10   ?CLN8   ?CSBDS ↵
    DO YOU WANT TO ENTER ANOTHER GROUP OF LINES [N]?     ↵

  ENTER A COMMAND:
DG-26896T
```

Figure 4-3. Sample IAC Dialog, with IAC-16 and IAC-8 (concluded)

In Figure 4-3, for IAC, we defaulted the device code and base line number, and we described it as a 16-line IAC. Then we described lines 0-2, hard-copy line 3, graphics lines 4-12, and specified the letter-quality printer on line 13. We were then finished with this IAC.

Next we added IAC1, with default device code and base line number 16, and described it as an 8-line IAC. We specified the modem lines, 0 and 1. Then we checked legal responses to CHARACTERISTIC WORD 0 and specified ?MMRI for the modems followed by other desired characteristics. All these mnemonics are described in Table 4-3. When you specify one nondefault characteristic, you must specify *all* the ones you want. The angle brackets, <>, are notation abbreviations; interpret them as if they were parentheses in an arithmetic expression.

NOTE: If a console is uppercase only, be sure to specify lower to uppercase conversion (?MUCO, from Table 4-3) for its line. If you don't, the console may drop all lowercase letters sent to it.

Proceeding to CHARACTERISTIC WORD 1, we checked and found the characteristics we wanted: ?MMOD for the modem lines; and we entered these and the standard other characteristics.

We defaulted CHARACTERISTIC WORD 2 and 3.

For the initialization word, we specified the clock (baud rate) mnemonic ?CR300, 300 baud, for the modem lines, and gave the standard other characteristics. Then we chose no more lines to add; and we were finished with the IACs.

When the new system runs, the console names on lines 0-2 will be CON2-CON4; on line 3 the name will be CON5; on lines 4-12 CON6-CON14; on line 13 the letter-quality printer will be CON15; and on IAC1 lines 0-1 will be CON18-CON19.

The line-specification procedure would have been much simpler without the modem lines; we'd have simply defaulted all the characteristics and initialization words on IAC1.

ULMs

A Universal Line Multiplexor is a single board which controls asynchronous lines, synchronous lines, or both, depending upon the ULM model. (On S/20s, the ULM is two boards.) ULM's which control asynchronous lines can handle either local or modem lines. The standard device code for the ULM is 34.

The ULM we will describe here controls asynchronous lines. If your ULM controls a synchronous line and you wish to use that line, you should describe the multiplexor to AOSGEN as an SLM rather than a ULM. Describing SLMs to AOSGEN is covered in a later section of this chapter. You cannot describe both synchronous and asynchronous lines on a ULM.

ULM Dialog

To add a ULM, use the Add command, then specify ULM:

A ↵
NAME OF DEVICE TO BE ADDED: ULM ↵

I/O PROCESSOR [NONE]: ? ↵
DCU50, NONE

Some CPU models support I/O processors, which handle the data input and output from low-speed devices such as CRT consoles and hard-copy consoles, DASHER printer consoles, card readers, plotters, and paper tape. A DCU/50 data control unit is one of these processors. If your system has a DCU/50, you should enter DCU50, otherwise type ↵ to select the default.

LINES [??]:

Now you need to describe the lines on this ULM — individually or in groups.

Lines attached to the ULM are numbered 0 through n, where n is the number of consoles up to a maximum of 3. If your system included a ULM expansion board, the lines attached to this board are numbered 4 through 7. When your new system runs, each console on the ULM will be named CON(line-number + 2). For example, the name of the console on line 4 will be CON6.

ULM Console Lines Example

The following example, Figure 4-4, tries to show you how to handle all common line-console configurations. It assumes that lines are arranged as follows:

- ULM line 0 is attached to a local D200 CRT
- ULM line 1 is attached to a letter-quality printer
- ULM line 2 is attached to a graphics CRT
- ULM line 3 is attached to a modem line

```
ENTER A COMMAND:      A ↲
  NAME OF DEVICE TO BE ADDED:      ULM ↲
  I/O PROCESSOR:      DCU/50 ↲
  LINES [??]:      0 ↲
  CONSOLE TYPE [CRT3]:      ↲
  INPUT BUFFER LENGTH [96]:      ↲
  OUTPUT BUFFER LENGTH [128]:      ↲
  CHARACTERISTIC WORD 0 [STANDARD]:      ↲
  CHARACTERISTIC WORD 1 [STANDARD]:      ↲
  CHARACTERISTIC WORD 2
    LINES PER PAGE [STANDARD]:      ↲
    CHARACTERS PER LINE [STANDARD]:      ↲
  CHARACTERISTIC WORD 3 [STANDARD]:      ↲
  INITIALIZATION WORD [STANDARD]:      ↲

  DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]:      Y ↲

  LINES [??]:      1 ↲
  CONSOLE TYPE [CRT3]:      TTY ↲
  INPUT BUFFER LENGTH [96]:      32 ↲
  OUTPUT BUFFER LENGTH [128]:      255 ↲
  CHARACTERISTIC WORD 0 [STANDARD]:      ↲
  CHARACTERISTIC WORD 1 [STANDARD]:      ↲
  CHARACTERISTIC WORD 2
    LINES PER PAGE [STANDARD]:      ↲
    CHARACTERS PER LINE [STANDARD]:      ↲
  CHARACTERISTIC WORD 3 [STANDARD]:      ↲
  INITIALIZATION WORD [STANDARD]:      ?PAR0   ?COD3   ?B1200   ?STP0 ↲

  DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]:      Y ↲

  LINES [??]:      2 ↲
  CONSOLE TYPE [CRT3]:      CRT6 ↲
  INPUT BUFFER LENGTH [96]:      ↲
  OUTPUT BUFFER LENGTH [128]:      ↲
  CHARACTERISTIC WORD 0 [STANDARD]:      ↲
  CHARACTERISTIC WORD 1 [STANDARD]:      ↲
  CHARACTERISTIC WORD 2
    LINES PER PAGE [STANDARD]:      ↲
    CHARACTERS PER LINE [STANDARD]:      ↲
  CHARACTERISTIC WORD 3 [STANDARD]:      ↲
  INITIALIZATION WORD [STANDARD]:      ↲

  DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]:      Y ↲

  LINES [??]:      3 ↲
  CONSOLE TYPE [CRT3]:      ↲
  INPUT BUFFER LENGTH [96]:      ↲
  OUTPUT BUFFER LENGTH [128]:      ↲
  CHARACTERISTIC WORD 0 [STANDARD]:      ↲
  CHARACTERISTIC WORD 1 [STANDARD]:      ?MMOD   ?MULC   ?MWRP   ?CRT3 ↲
  CHARACTERISTIC WORD 2
    LINES PER PAGE [STANDARD]:      ↲
    CHARACTERS PER LINE [STANDARD]:      ↲
  CHARACTERISTIC WORD 3 [STANDARD]:      ↲
  INITIALIZATION WORD [STANDARD]:      ?PAR0   ?COD3   ?B300   ?STP0 ↲

  DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]:      ↲
```
DG-26897T

Figure 4-4. Sample ULM Dialog

CRT1 = 4010I CRT
CRT2 = 6012 CRT
CRT3 = D2, D200, D210, D211
CRT4 = OTHER DG COMPATIBLE CRT
CRT6 = GRAPHICS, D400, D410, D450, D460, G300

Console Line Characteristics

The characteristics that you specify during AOSGEN are not cast in bronze. You can always edit the line specifications later. Also, you can change the default system console characteristics at runtime with the CLI command CHARACTERISTICS/DEFAULT. Users can change their console characteristics temporarily with the CHARACTERISTICS command, without the /DEFAULT switch.

All characteristics that you can default or specify during AOSGEN are included in Tables 4-3, 4-4, and 4-5. Table 4-3 shows the line characteristics: an S within a column means that the characteristic is standard; an A means that it is available. Tables 4-4 and 4-5 show line initialization characteristics for ASLMs/IACs and ALMs/ULMs, respectively.

If you specify one nondefault characteristic, you must specify all the ones you want for the line(s). AOSGEN includes only those you actually specify.

Table 4-3. AOS Console Line Characteristics

Word	Characteristic	Bit Mnemonic	CRT1	CRT2	CRT3 and CRT6	CRT4	TTY
0	Simulate Tabs: tab key moves cursor right 8 columns.	?MST	S	S	S	S	S
	Simulate Form Feeds: new screen (CRT) or form feed (TTY).	?MSFF	A	A	A	A	A
	Even parity on input.	?MEPI	A	A	A	A	A
	8 data bits per character.	?M8BT	A	A	A	A	A
	Set even parity on output.	?MSPO	A	A	A	A	A
	Output 21 rubout characters after each form feed.	?MRAF	A	A	A	A	A
	Output 2 rubout characters after each TAB character.	?MRAT	A	A	A	A	A
	Output 2 rubout characters after each NEW LINE character.	?MRAC	A	A	A	A	A
	Non-ANSI Standard console. Such consoles have a small LF key and large RETURN key (ANSI standard consoles have a large NEW LINE key and small CR key).	?MNAS	S	S	A	S	S
	Convert characters 175 and 176 to 33 (octal). Specify this mnemonic for any console with an ESCAPE character of 175 or 176 (octal).	?MOTT	A	A	A	A	A
	Enforce End Of Line: truncate any line longer than that given in word 2. The alternative is ?MWRP (word 1).	?MEOL	A	A	A	A	A
	Uppercase Only: convert lowercase to uppercase. The alternative (word 2) is ?MULC.	?MUCO	A	A	A	A	A
	Monitor Ring Indicator on modem control line(s).	?MMRI	A	A	A	A	A
	Form Feed: send form feed to console(s) when it is opened for I/O.	?MFF	A	A	A	A	A
	Echo control. If you specify ?MEB0 and omit ?MEB1, all characters will echo as typed except CTRL will echo as ^ and ESC as $. This is the default. If you specify both, CTRL and ESC won't echo.	?MEB0 ?MEB1	S A	S A	S A	S A	S A
	Echo control. ?MEOC is equivalent to ?MEB0; ?MEOS is equivalent to ?MEB1.	?MEOC ?MEOS	S A	S A	S A	S A	S A

(continues)

Table 4-3. AOS Console Line Characteristics

Word	Characteristic	Bit Mnemonic	CRT1	CRT2	CRT3 and CRT6	CRT4	TTY
1	Teletypewriter: line is attached to a TTY.	?TTY	—	—	—	—	S
	CRT1: line is attached to a model 4010I CRT console	?CRT1	S	—	—	—	—
	CRT2: line is attached to a model 6012 CRT console	?CRT2	—	S	—	—	—
	CRT3: line is attached to a D200-compatible console. (includes D210, D211, and desktop system oriented printers and plotter).	?CRT3	—	—	S for CRT3 A for CRT6	—	—
	CRT4: line is attached to another DG-compatible CRT console.	?CRT4	—	—	—	S	—
	CRT6: line is attached to a D400, D410, D450, D460 or G300 graphics CRT.	?CRT6	—	—	S for CRT6	—	
	Upper- and Lowercase: accept both upper- and lowercase characters as input. If you default word 1, ?MULC is the default. But if you specify characteristics for this word and omit ?MULC, the system converts lowercase to uppercase.	?MULC	A	A	S	A	A
	Page Mode: console will display LPP (Lines Per Page) lines, then wait for CTRL-Q before displaying next sequence of LPP lines.	?MPM	A	A	A	A	A
	Do Not Receive Messages; prevents console from receiving SEND messages.	?MNRM	A	A	A	A	A
	Modem: line is attached to a modem interface.	?MMOD	A	A	A	A	A
	Type of Device is n: ignore this characteristic; omit it.	?MTD0-3	—	—	—	—	—
	Time Out: enable time-out on console line(s).	?MTO	A	A	A	A	A
	ESC character has same interrupt effect as CTRL-C CTRL-A.	?MESC	A	A	A	A	A
	Wrap line onto next line if line is too long (done in hardware).	?MWRP	S	A	S	S	A
	Function Keys Terminate text input.	?MFKT	A	A	A	A	A

(continued)

Table 4-3. AOS Console Line Characteristics

Word	Characteristic	Bit Mnemonic	CRT1	CRT2	CRT3 and CRT6	CRT4	TTY
2	Values are taken from AOSGEN query. The ones shown here are the defaults.						
	Lines per page.		20	24	24	24	30
	Characters per line.		80	80	80	80	72
3	Data flow control over console lines (X-OFF, X-ON). Can prevent character loss when the input buffer of the host computer or character device is full. The X-OFF character is octal 23, which CTRL-S commonly generates from a keyboard. The X-ON character is octal 21, which CTRL-Q commonly generates from a keyboard.						
	Console line is enabled to recognize X-OFF characters from host computer. The console line stops sending data until the host issues X-ON.	?MIFC	A	A	A	A	A
	Console line is enabled to send X-OFF characters to host computer; the host stops sending data until the console line issues X-ON. Useful on on D400, D450, or G300 graphics console lines.	?MOFC	A	A	A	A	A
	When output flow control is enabled (?MOFC), the X-ON and X-OFF characters are recognized, even when data on the line is being read in binary mode.						

(concluded)

Table 4-4. Line Initialization Word Characteristics for ASLMs/IACs

Mnemonic	Meaning
?CLNn	Select the number of data bits per character, excluding the parity bit (if any):
?CLN5	5 bits per character.
?CLN6	6 bits per character.
?CLN7	7 bits per character.
?CLN8	8 bits per character; the IAC may generate or ignore hardware parity as specified via the ?CPR mnemonic (standard).
?CS10	Transmit 1 stop bit per character (standard).
?CS15	Transmit 1.5 stop bits per character.
?CS20	Transmit 2 stop bits per character.
?CPR0	Disable parity checking; ignore parity bit (standard).
?CPR1	Odd parity.
?CPR2	Even parity.
?CSBEN	Enable split baud rate (IAC-8s only). Receive rate is that given to AOSGEN question *IAC SPLIT BAUD RATE.* Useful for a console with attached printer.
?CSBDS	Disable split baud rate (standard).
?CRn	Select the data (baud) rate for this group of lines. n can be: 75, 110, 134, 150, 300, 600, 12H (1200), 18H, 20H, 24H, 36H, 48H, 72H, 96H, or 19K (19,200). The standard for CRTs is 96H (9600); for TTYs it is 300.

Table 4-5. Line Initialization Word Characteristics for ALMs/ULMs

Mnemonic	Meaning
?PARn	Select parity:
?PAR0	Disable parity checking; ignore parity bit (standard).
?PAR1	Odd parity.
?PAR2	Even parity.
?CODn	Select the number of data bits per character, including the parity bit (if any).
?COD0	5 bits per character.
?COD1	6 bits per character.
?COD2	7 bits per character.
?COD3	8 bits per character; the ALM/ULM may generate or ignore hardware parity as specified via the PAR mnemonic (standard).
?STP0	Transmit 1 stop bit per character (standard).
?STP1	Transmit 2 stop bits per character.
?CLKn	Each ALM has four clocks with jumper-selectable baud rates. Choose the clock that provides the desired baud rate for this group of lines:
?CLK0	Select clock 0 rate (standard) (ALM only)
?CLK1	Select clock 1 rate. (ALM only)
?CLK2	Select clock 2 rate. (ALM only)
?CLK3	Select clock 3 rate. (ALM only)
?Bn	AOS supports 11 line speeds for ULM devices. Choose the speed that provides the desired baud rate for this group of lines:
?B110	Use 110 baud. (ULM only)
?B150	Use 150 baud. (ULM only)
?B200	Use 200 baud. (ULM only)
?B300	Use 300 baud. (ULM only)
?B600	Use 600 baud. (ULM only)
?B1200	Use 1200 baud. (ULM only)
?B1800	Use 1800 baud. (ULM only)
?B2400	Use 2400 baud. (ULM only)
?B4800	Use 4800 baud. (ULM only)
?B9600	Use 9600 baud. (ULM only)
?B19K	Use 19,200 baud. (ULM only)

About 8-Bit Character Handling

DG DASHER model D211, D220, D410, and D460 consoles can send and display 8-bit characters. The main advantage of 8-bit character handling is the ability to read and display characters with values above 177 octal — which includes many characters in the international character set (like the U.K. currency symbol) and other special characters. In 7-bit mode, the high bit is ignored — which means the console can't see codes above 177 octal.

One disadvantage of 8-bit character handling is that the CLI macros that use the !ASCII pseudomacro may display garbled messages. People often add 200 (octal) to the ASCII value of a character to conceal it from the CLI. For example, to display a comma in a CLI macro, you must use the string [!ASCII 254] ([!ASCII 54] will produce a comma, which the CLI will display as a space). But if 8-bit character handling is enabled on a console and someone runs a macro with [!ASCII 254], the system will see the value of code 254, which is *not* a comma. Note that also, on a console with a non-U.S. keyboard, some keys send *different codes* in 7-bit mode from those in 8-bit mode. For example, on a French keyboard the C cedilla key will produce C with a cedilla in 7-bit mode, but will produce ^\ (ASCII 34) in 8-bit mode.

If you want 8-bit character handling, a simple way to get it is to generate your consoles with 7-bit handling (the default). Then, use the CHARACTERISTICS command the with 8BT/ON switch to enable 8-bit handling. To restore 7-bit handling (on U.S. keyboards), the person at the terminal can type CHARACTERISTICS/RESET), or simply log off and on again.

The character mode is controlled by small dual in-line package (DIP) switches on the back of the terminal (HOST group) as follows.

- For 8-bit mode, which you can enable and disable via software, set the bit mode DIP switch to 8-bit mode; set the parity DIP switches to no parity (parity none). With these DIP switch settings, U.S. keyboards will be able to handle either 7-bit or 8-bit mode.
- For 7-bit mode, which cannot provide 8-bit characters but does allow certain non-U.S. characters to display properly, set the bit mode DIP switch to 7-bit mode and set the parity DIP switches to mark parity.

The master CLI (PID 2) can change the default characteristics (but not the mode) of a console with the /DEFAULT switch and console name, *before* EXEC enables the console. For example, the command CHARACTERISTICS/DEFAULT/ON/8BT @CON22) sets the /8BT switch on at console 22. If need be, you can put such commands in the system UP macro.

Output Flow Control

Some programs like CEO®, or programs your site may choose to write, use binary mode to handle data on most consoles — including DASHER D210, D211, D220, D400, D410, D450, D460, D470, and G300 terminals. If a program that uses binary mode will write to any of these terminals, the output-flow control characteristic must be selected on the console line to that terminal. You can do this during AOSGEN by specifying the mnemonic ?MOFC in word 3, and add any other mnemonics as well. Or, you can do it via the CHARACTERISTICS command switches, /ON/OFC, in the UP macro, or via a user log-on macro (Chapter 5).

Proceeding

After you have described the console lines, you might want to examine the spec file with the Current command. If you have another IAC or ASLM to describe, return to "IAC Dialog," or "ASLM Dialog," above. If you have finished describing the console lines, proceed to the next section.

Line Printers

AOS supports two types of programmed I/O line printers:

LPA An LPA is a model 4034 commercial I/O printer. AOS supports up to two LPA printers, device names LPA and LPA1.

LPC An LPC is a DASHER LP2 line printer. AOS supports up to two LPC printers, device names LPC and LPC1.

If your primary printer is an LPA or LPC, add it using the Add command:

ENTER A COMMAND: A)
NAME OF DEVICE TO BE ADDED: LPA) (or LPC))
DEVICE CODE [17]:

As with other devices, press) to select the default code unless you know that the first line printer is connected to a different code:

)
OUTPUT BUFFER BYTE LENGTH [136])
CHARACTERISTIC WORD 0 [STANDARD])
CHARACTERISTIC WORD 1 [STANDARD])
CHARACTERISTIC WORD 2
LINES PER PAGE [STANDARD])
CHARS PER LINE [STANDARD])

There are two types of data channel line printer, as follows.

LPB An LPB is a commercial I/O printer, with a vertical forms unit. AOS supports up to eight LPB printers, device names LPB, LPB1, LPB2, ..., LPB7.

LPD An LPD is either a laser document printer (model 4225), or DASHER model 4325-4328 printer. AOS supports up to two LPD type printers, device names LPD and LPD1.

If your primary printer is an LPB or LPD, when AOSGEN is asking for a COMMAND — add it:

```
                    A ↵
NAME OF DEVICE TO BE ADDED:     LPB ↵  (or LPD↵)
DEVICE CODE [17]:
```

As with other devices, press ↵ to select the default code unless you know that the first line printer is connected to a different code:

```
                    ↵
ENTER A COMMAND:
```

If you have a second line printer, type A↵ and specify the name. If the second printer is the same type as the first, its name is name1; e.g., LPB1. For example

```
                    A ↵
NAME OF DEVICE TO BE ADDED:     LPB1 ↵
DEVICE CODE [57]:     ↵
ENTER A COMMAND:
```

If the type of the second printer *differs* from the first (for example, the first printer is an LPB and the second an LPD), you can enter the name of the second printer as the primary (e.g., LPD). But if you do this, be sure to specify the correct device code for the secondary printer. The default secondary printer device code is 57. For example,

```
                    A ↵
NAME OF DEVICE TO BE ADDED:     LPD ↵
DEVICE CODE [17]:     57 ↵
ENTER A COMMAND:
```

Plotters

AOS supports up to two model 4017 digital plotters, device names PLA and PLA1. To have the new system support a plotter, Add a device, then specify the name, and, unless the plotter is connected to a nonstandard device code, default the device code. For example

```
                    A ↵
NAME OF DEVICE TO BE ADDED:     PLA ↵
DEVICE CODE [15]:  ↵
OUTPUT BUFFER LENGTH [64]:  ↵
CHARACTERISTIC WORD 0 [STANDARD]:  ↵
CHARACTERISTIC WORD 1 [STANDARD]:  ↵
ENTER A COMMAND:
```

There are no standard characteristics for plotters. The nonstandard characteristics you can specify are shown in Table 4-6.

Card Readers

AOS supports up to 2 model 4016 card readers, device names CRA and CRA1. To have the new system support a card reader, Add a device, then specify the name, and, unless the reader is connected to a nonstandard device code, default the device code. For example

```
                    A )
NAME OF DEVICE TO BE ADDED:     CRA )
DEVICE CODE [16]:  )
INPUT BUFFER LENGTH [164]:  )
CHARACTERISTIC WORD 0 [STANDARD]:  )
CHARACTERISTIC WORD 1 [STANDARD]:  )
ENTER A COMMAND:
```

The default buffer holds the images of two cards. There are no standard characteristics for card readers. The nonstandard characteristics you can specify are shown in Table 4-6.

Paper Tape Reader/Punch

AOS supports up to two model 6013 or model 4011B paper tape readers, device names TRA and TRA1. To have your system support a paper tape reader, Add a device, then specify the name, and unless the reader is connected to a non-standard device code, default the device code. For example:

```
                    A )
NAME OF DEVICE TO BE ADDED:     TRA )
DEVICE CODE [12]:  )
INPUT BUFFER LENGTH [128]:  )
CHARACTERISTIC WORD 0 [STANDARD]:  )
CHARACTERISTIC WORD 1 [STANDARD]:  )
ENTER A COMMAND:
```

The standard and nonstandard characteristics you can specify for paper tape readers are shown in Table 4-6.

AOS supports up to two model 4012 paper tape punches, device names TPA and TPA1. To have your system support a paper tape punch, Add a device, then specify the name, and unless the punch is connected to a non-standard device code, default the device code. For example:

```
                    A )
NAME OF DEVICE TO BE ADDED:     TPA )
DEVICE CODE [13]:  )
OUTPUT BUFFER LENGTH [64]:  )
CHARACTERISTIC WORD 0 [STANDARD]:  )
CHARACTERISTIC WORD 1 [STANDARD]:  )
CHARACTERISTIC WORD 2
  LINES PER PAGE [STANDARD]:  )
  CHARACTERS PER LINE [STANDARD]:  )
ENTER A COMMAND:
```

The standard and nonstandard characteristics you can specify for paper tape punches are shown in Table 4-6.

Table 4-6. AOS Plotter, Card Reader, Paper Tape Characteristics

Word	Characteristic	Bit Mnemonic	PLA PLA1	CRA CRA1	TPA TPA1	TRA TRA1
0	Simulate Tabs: tab key moves cursor right 8 columns.	?MST	n/a	n/a	A	A
	Simulate Form Feeds: new screen (CRT) or form feed (TTY).	?MSFF	n/a	n/a	A	A
	Set even parity on output.	?MSPO	n/a	n/a	S	A
	Output two rubout characters after each NEW LINE character.	?MRAC	n/a	n/a	A	A
	Enforce End Of Line: truncate any line longer than that given in word 2.	?MEOL	n/a	n/a	A	n/a
1	Upper and LowerCase: accept both upper- and lowercase characters as input. By default, lowercase characters are converted to uppercase.	?MULC	n/a	n/a	n/a	A
	Time out: enable time-out	?MTO	A	A	A	S
	Retain trailing blanks in text mode. By default, the system inserts a NEW LINE after the last nonblank characters and drops trailing blanks.	?MTSP	n/a	A	n/a	n/a
	Use packed format for binary mode. This format consists of four 12-bit columns packed into three memory words. By default, the system puts each 12-bit column, right justified, in each memory word and uses the leftmost 4 bits for reader status.	?MPBN	n/a	A	n/a	n/a
	Time out: enable timeouts on device.	?MTO	A	A	A	A
	Wrap line onto next line if line is too long (done in hardware).	?MWRP	A	A	A	A
	No New Line: do not append NEW LINE character to end of card image. By default, the system appends a NEW LINE to allow for data sensitive reads later.	?MNNL	n/a	A	n/a	n/a
	On output, provide a leader or trailer (60 null characters) on each OPEN and CLOSE command. On input, read past any leading null characters on each OPEN.	?MLT	n/a	n/a	S	S
	Lines per page		n/a	n/a	60	n/a
	Characters per line		n/a	n/a	136	n/a

Integral Array Processors (IAPs)

An IAP is a processor designed to do signal and/or array processing. If your system is an AP/130, it has an IAP. If your system is an S/250, it may optionally include an IAP. If your system has an IAP, Add a device, then specify the name:

A ↵

NAME OF DEVICE TO BE ADDED: IAP ↵

ENTER A COMMAND:

Multiprocessor Communication Adapters (MCAs)

An MCA is a small processor, used in compressed Local Area Networks with DG systems that are no more than 150 feet apart. Each MCA can handle up to 16 asynchronous lines (nodes).

(For Local Area Network communications with systems up to *1 mile* apart, there's a device called a Network Bus Adapter (NBA). If your system has one or more NBAs, it must be configured with the XODIAC networking NETGEN program. You do not identify NBAs to AOSGEN, so we won't describe them further in this book.)

Each *MCA,* however, must be identified to AOSGEN.

AOS supports up to two MCAs — device names MCA and MCA1. Internally, an MCA is two devices: the transmitter, MCAT(1), and the receiver, MCAR(1). MCAT must be on device code 6 and MCAR on device code 7; MCAT1 must be on device code 46, and MCAR1 on device code 47.

AOSGEN sees the transmitter and receiver as one device. Simply add a device and type the name. For example,

```
                    A ↵
    NAME OF DEVICE TO BE ADDED:      MCA ↵
ENTER A COMMAND:
```

If you have a second MCA, add MCA1 as you did MCA.

Supporting Synchronous Devices (SLMs, SDCUs, ISCs, ASLMs)

Synchronous communications are handled by one or more of the following devices:

- Synchronous Line Multiplexors (SLMs)
- Data Control Units (SDCUs) with SLMs
- Intelligent Synchronous Controllers (ISCs)
- Asynchronous/Synchronous Line Multiplexors (ASLMs)
- (Universal Line Multiplexors (ULMs))

The last device has been included in parentheses because you will be describing a ULM which has synchronous lines as an SLM to AOSGEN. If you Add a ULM device, AOSGEN assumes that you will be only be using the asynchronous lines on the ULM.

If you want your system to use synchronous lines, you'll tell AOSGEN which kind of synchronous device you have, then describe the synchronous lines attached to it. If your synchronous lines will be used by the XODIAC networking software, you should run the network generation program supplied with XODIAC in addition to describing the sync lines to AOSGEN.

After you describe the synchronous device, you should use the Parameter command to select the number of communications buffers AOS will reserve for the device. The default number of buffer is one. (See "Parameters for the New System" for more information on system parameters.)

Ranges of Lines

You can describe synchronous lines individually or in groups. AOSGEN will ask questions about each entry, so, if you choose a group of lines, all the lines in the group will be treated the same way, even if they are different. After you describe each line, or group of lines, AOSGEN asks if you want to describe more lines, so you can proceed sequentially to describe them all.

To specify individual lines, separate the entries with one or more spaces. To specify a range of lines, separate the first and last line number with a space-dash-space. For example,

0 2	specifies lines 0 and 2
0 - 2	specifies lines 0 through 2

SLMs

An SLM has one or more synchronous multiplexor (mux) boards. Each board handles synchronous lines and the modems attached to these lines. The default device code for the SLM is 44.

A DCU/50 or DCU/200 data control unit is sometimes combined with an SLM to handle the data input and output to the sync lines. If your system has a DCU which will be used for synchronous communications, skip to the section 'SDCUs'. You do not need to describe both the SLM and DCU to AOSGEN. When you describe the DCU, AOSGEN takes care of the rest.

SLM Dialog

To add an SLM, Add a device, then specify SLM:

A ⤶

NAME OF DEVICE TO BE ADDED: SLM ⤶

LINES [??]:

Now you need to describe the lines on this SLM, individually or in groups. When the new system runs, each line on the SLM will have the line name of SLNn, where n is the line number you give to AOSGEN. You need not describe the lines consecutively, however the lowest line in each line sequence or group must be higher than any line previously described.

SLM Lines Example

The following example shows you how to describe your synchronous lines to AOSGEN.

```
ENTER A COMMAND:   A ⤶
    NAME OF DEVICE TO BE ADDED:   SLM ⤶
    LINES [??]:   0 ⤶
    FULL OR HALF DUPLEX [HDX]:   ⤶
    SWITCHED OR DEDICATED [DED]:   ⤶

    DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]   Y ⤶

    LINES [??]:   1 ⤶
    FULL OR HALF DUPLEX [HDX]:   FDX ⤶
    SWITCHED OR DEDICATED [DED]:   SWT ⤶

    DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]   ⤶
ENTER A COMMAND:
```

This example describes synchronous lines 0 and 1. We took the default description for line 0: half duplex, which means that the line only transmits data in one direction at a time, and dedicated, which means that the phone line you use is only attached to your device. Line 1 is different, it is full duplex, which means that the line can transmit data in both directions at once, and switched, meaning that the line may be connected to more than one device.

The way you describe your synchronous lines to AOSGEN depends on the modem you are using. Modems that are connected all the time are usually *fully dedicated* so you would specify FDX ⤶ and ⤶. Dial-up modems are usually half duplex and switched, so you would specify ⤶ and SWT ⤶.

SDCUs

An SDCU is a DCU/50 or a DCU/200 processor that has one or more synchronous line multiplexors (SLMs). Each DCU can handle up to 32 synchronous lines, and you can specify up to four DCUs during AOSGEN. However, the total number of synchronous lines you specify during AOSGEN cannot exceed 32. SDCU is default device code 40, SDCU1 is default device code 41, SDCU2 is default device code 42, and SDCU3 is default device 44.

SDCU Dialog

To add a DCU, use the Add command, then specify SDCU (or SDCU1, SDCU2, SDCU3):

```
ENTER A COMMAND:    A ↵
    NAME OF DEVICE TO BE ADDED:    SDCU ↵

    DCU TYPE [DCU50]:    ? ↵
    DCU50 OR DCU200
```

AOS supports both DCU/50 and DCU/200 devices for synchronous communications. Although the devices are similar, the DCU/200 has more local memory. If your computer is an MV/8000, your DCU is a DCU/200. If you have a DCU/200, type DCU200 ↵, otherwise type ↵ to take the default.

From here on in AOSGEN asks the same questions we described in the previous section. The next section gives you an example of how to describe your DCU and the synchronous lines attached to it.

SDCU Lines Example

The following example will show you how to describe your synchronous lines to AOSGEN.

```
ENTER A COMMAND:    A ↵
    NAME OF DEVICE TO BE ADDED:    SDCU1 ↵
    LINES [??]:    0 1 ↵
    FULL OR HALF DUPLEX [HDX]:    FDX ↵
    SWITCHED OR DEDICATED [DED]:    ↵

    DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]    Y ↵

    LINES [??]:    2 3 ↵
    FULL OR HALF DUPLEX [HDX]:    ↵
    SWITCHED OR DEDICATED [DED]:    SWT ↵

    DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]    ↵
ENTER A COMMAND:
```

This example tells AOSGEN that our system has a DCU/200. Then, it describes two line sets: lines 0 and 1 are full duplex and dedicated, lines 2 and 3 are half duplex and switched.

How you describe your synchronous lines to AOSGEN depends on the modem you are using. Modems that are connected all the time are usually *fully dedicated,* so you would specify FDX ↵ and ↵. Dial-up modems are usually half duplex and switched, so you would specify ↵ and SWT ↵.

You must specify SDCU lines sequentially to AOSGEN, as shown in the example. If you don't specify them sequentially, the new system may not run.

ISCs

An ISC has a small processor and synchronous multiplexors on one circuit board; it can handle up to two synchronous lines.

ISC Dialog

To add an ISC, Add a device, then specify ISC:

```
                    A ↵
    NAME OF DEVICE TO BE ADDED:    ISC ↵

    ISC TYPE [ISMC2]:    ↵
```

There are two types of ISCs. An ISMC2 is a 7 by 9 inch board designed for DESKTOP GENERATION systems. An ISC2 is a 15 by 15 inch board designed for S/120, S/140, and S/280 computers. So, if you have a DESKTOP GENERATION system, press ↵ to accept the default. On the other systems, type ISC2 ↵.

```
    ISC BASE LINE NUMBER [0]:    ↵
    LINES [??]:
```

Now you need to describe the lines on this ISC, individually or in groups. When the system runs, each line on the ISC will have the line name of SLNn, where n is the line number you give to AOSGEN. You need to describe the lines consecutively; however, the lowest line in each sequence or group must be higher than any line previously described.

FULL OR HALF DUPLEX [HDX]:

Half-duplex synchronous lines are more common than full-duplex synchronous lines. The bisynchronous protocol is a half-duplex protocol. So, generally, you should take the half-duplex default by pressing ⤶. But if you *know* that this line will be on a full-duplex modem, *and* that the modem takes a while to switch from transmit to receive, you should answer FDX ⤶.

SWITCHED OR DEDICATED [DED]:

If this synchronous line will use a dedicated (leased) phone line, accept the default by pressing ⤶. If it uses a switched phone line, type SWT ⤶.

IS THE CLOCK FOR THIS LINE EXTERNAL [Y]:

Each ISC has an internal clock, but most modems have and depend on their own clocks. So, generally, if this is a modem line, or you want the line on an external clock, press Y ⤶ for the default.

If you want to use the device's internal clock, press N ⤶.

INTERNAL CLOCK FREQUENCY [?CR24H]:

If you answered N ⤶ to the last question, you must now choose the frequency of the device's internal clock, measured as baud rate. You can choose 300, 600, 12H (hundred), 24H, 48H, 96H, or 19K. So, type ?CRnnn ⤶, where nnn is the baud rate.

ISC Lines Example

The following example shows you how to describe your synchronous lines to AOSGEN.

ENTER A COMMAND: A ⤶
NAME OF DEVICE TO BE ADDED: ISC ⤶
ISC TYPE [ISMC2]: ⤶
ISC BASE LINE NUMBER [0]: ⤶
LINES [??]: 0 ⤶
FULL OR HALF DUPLEX [HDX]: ⤶
SWITCHED OR DEDICATED [DED]: ⤶
IS THE CLOCK FOR THIS LINE EXTERNAL [Y]: ⤶

DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N] ⤶
ENTER A COMMAND:

If you want to add another group of lines, type Y ⤶. Otherwise, press ⤶ to accept the default.

This example describes synchronous line 0. We took the default description for line 0; half duplex, which means that the line only transmits data in one direction at a time, and dedicated, which means that the phone line you use is only attached to your device.

ASLMs

Each Asynchronous/Synchronous Line Multiplexor is one circuit board which supports both asynchronous and synchronous lines. (On S/20s and DG DESKTOP GENERATION systems, it's called a USAM.) This section outlines how to describe synchronous lines on an ASLM. If you wish to describe asynchronous ASLM lines you will find additional information in an earlier section of this chapter.

Describing ASLM lines is very similar to describing IAC lines:

You specify a base line number for each ASLM, then describe lines 0 to n, where n is the number of lines attached to consoles (up to a maximum of 3).

ASLM Dialog

To add an ASLM, use the Add command, then specify ASLM:

```
                        A ↵
    NAME OF DEVICE TO BE ADDED:     ASLM ↵
    DEVICE CODE [34]:
```

The default device code for ASLM is 34; for ASLM1 it is 74. Unless you know that this ASLM is connected to a nonstandard device code, type ↵ to select the default. ASLM2 and ASLM3 do not have a default device code. You should enter the nonstandard device code that the ASLM board is connected to.

```
                        ↵
    ASLM BASE LINE NUMBER [0]:
```

You can select any valid base line number for this ASLM. The number must be 0 or divisible by 4. The names of its sync lines will be SLN (base-line-number + lines-number-on-this-ASLM). For the first ASLM, you might choose the default:

```
                        ↵
    LINES [??]:
```

Now you need to describe the lines on this ASLM — individually, or in groups. The line numbers you enter are specific to this ASLM — ranging from 0 through 3. In this example, let's say you want to describe line 0:

```
                        0 ↵
    ARE THESE LINES SYNC OR ASYNC [ASYNC]:
```

Since you are describing synchronous lines, type SYNC ↵. The following example shows you how to describe the synchronous lines on your ASLM.

ASLM Lines Example

```
ENTER A COMMAND:    A ↵
    NAME OF DEVICE TO BE ADDED:     ASLM ↵
    DEVICE CODE [34]:    ↵
    ASLM BASE LINE NUMBER [0]:    ↵
    LINES [??]:    0 ↵
    ARE THESE LINES SYNCH OR ASYNC [ASYNC]:     SYNC ↵
    FULL OR HALF DUPLEX [HDX]:    FDX ↵
    SWITCHED OR DEDICATED [DED]:    ↵
    IS THE CLOCK FOR THIS LINE EXTERNAL [Y]:    ↵

    DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]    Y ↵

    LINES [??]:    1 ↵
    ARE THESE LINES SYNC OR ASYNC [ASYNC]:    SYNC ↵
    FULL OR HALF DUPLEX [HDX]:    ↵
    SWITCHED OR DEDICATED [DED]:    SWT ↵
    IS THE CLOCK FOR THIS LINE EXTERNAL [Y]:    N ↵
    INTERNAL CLOCK FREQUENCY [?CR24H]:    ? ↵
    ?CR<75 110 134 150 300 600 12H 18H 20H 24H 36H 48H 72H 96H 19K>
    INTERNAL CLOCK FREQUENCY [?CR24H]:    ↵

    DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]    ↵
ENTER A COMMAND:
```

This example tells AOSGEN that our synchronous device is an ASLM on device code 34 and that the base line number is 0. Then, it describes two synchronous lines, line 0 and line 1. Line 0 is full duplex and dedicated. Line 1 is half duplex and switched.

If you want your ASLM to handle synchronous lines, AOSGEN must know whether you want to use the external or the internal clock. Each ASLM has an internal clock with a frequency that you can select. But, most modems depend on their own clocks. So, generally, if a synchronous line is a modem line, or if you want the line on the external clock, take the default by typing ⤶. If you select the internal clock, AOSGEN asks you to specify the internal clock frequency. This example lists the possible responses from which we selected the default by typing ⤶.

Battery Backup (BBU) and Auto Restart

Your computer may have a battery for full backup. If it has no battery, skip this section.

If your computer has a backup battery, tell AOSGEN about it, adding the device named BBU:

ENTER A COMMAND: A ⤶
  NAME OF DEVICE TO BE ADDED: BBU ⤶
  FULL OR PARTIAL BACKUP [F]: ⤶

If your system has IACs or an IOP and your backup battery is strong enough to provide power to these devices, then you should select full backup. If your system does not have IACs or an IOP, or if your backup battery will not support them, then you should select partial backup.

  ENABLE AUTO RESTART AFTER POWER FAIL [Y]

Selecting auto-restart allows AOS to continue normally when power returns after a power failure. We recommend that you answer yes:

  ⤶

For auto-restart to work, all of the following must be true:

- The battery must provide full backup.
- Power must return before the battery is exhausted.
- BBU with auto restart must have been specified to AOSGEN.
- The computer LOCK switch (if any) must have been in the ON or LOCK position when power went down.

Handling power failures is described near the end of Chapter 6.

Having dealt with the backup battery issue, you're done with the hardware specifications for the new system. The next section explains system software parameters that you can default or specify.

As always, you can display all current system specifications with the Current command, list device specs with the List command, or respecify one or more specs with the Edit command.

Parameters for the New System

You can change certain software parameters to optimize system performance. AOSGEN supplies defaults for these. But, even for your first system, you may want to pick values other than the defaults for the SWAP parameter. In any case, you should at least examine this section before deciding to choose the default parameters.

To change one or more system parameters, type P⤶. AOSGEN then asks about the following parameters. As usual, it displays the current default in brackets.

ENTER A COMMAND: P ⤶

Parameter	What it Specifies
SWAP [default]	Swap file size in 512-byte disk blocks. The original default is 2000. We recommend 256*total-number-of-processes. Details are below.
CACHE [0]	Number of 512-byte buffers allotted for the system to do its I/O. The default is okay. Details are below.
FREQUENCY [10]	The frequency of the CPU real-time clock in cycles per second (Hz). The default is 10 Hz. Other choices are 50, 60, 100, or 1000 Hz. Several higher level languages — like AOS BASIC — expect the default frequency for their time-oriented statements. However, if the new system *must* synchronize with ac line frequency, you might select 50↵ or 60↵, depending on your ac power. If the new system *must* check something often, as in a some kind of process control situation, you might choose 1000↵ (despite the overhead involved in 1000 RTC interrupts per second).
SYNC BUFFERS [1]	AOS reserves from 1 to 26 memory pages for synchronous communications buffers. Once reserved, the system cannot use this memory for any other applications. If you specify a large number of synchronous buffer pages, your system will have less memory available for other users. On the other hand, if you specify a small number, you may impact the performance of your synchronous communications system. You should reserve at least # sync lines * maximum transmission size per line words. For example, if your system has four synchronous lines and you can transmit 512 bytes of data at a time, you would specify 4 * 512 = 2048 bytes = 1 page Refer to the documentation provided with the transmission utility for additional information about transmission and buffer sizes. If you do not specify any synchronous devices during the system-generation procedure, AOS does not reserve any communications buffers.
ACCESS CONTROL [YES]	Enables (↵) or disables (N↵) the AOS file access control mechanism. All file security, which is very important in a multiuser system, depends on access control. If you disable control (N↵), the multiuser environment will work, but every user will be able to access, modify, and delete every file in the system. Therefore, you will probably want to choose the default for this parameter.
DEMAND PAGING [NO]	Enables (Y ↵) or disables (↵) the Demand Paged Memory Management facility on M/600 CPUs. We recommend that you choose the default for this parameter.

Processes and Swap Files

AOS runs each program as a *process*, with its own process identifier (PID). There can be many processes — up to 64 — all running simultaneously.

Memory contention occurs when all currently active processes (including the AOS system and its peripheral manager) require more memory than the computer's main (physical) memory contains. Memory contention can occur much of the time.

In "heavy" memory contention, AOS removes whole *processes* (selecting blocked processes first), and keeps them in a swap file named SWAP.SWAP. AOS automatically creates and manages SWAP.SWAP. Later, AOS returns the processes in SWAP.SWAP to main memory.

SWAP.SWAP is a contiguous file whose size you specify to AOSGEN. (You can also change the size of SWAP.SWAP when you bring up the system.) To specify an ideal size for SWAP.SWAP, you need to know the maximum number of processes that will run on your system.

To some extent, you can control the total number of processes by limiting the number of "son" processes each user process can create. When the multiuser environment is running, the ?.CLI macro and Process Enviroment Display (PED) utility will tell you about all current PIDs. The ? macro, PREDITOR, and PED are described in the chapters that follow.

Generally, a size of highest-PID-number * 256 blocks is sufficient for the SWAP.SWAP file.

For example, if there will be a maximum of 40 PIDs, you might specify 10240⤶ to the AOSGEN SWAP question.

The disk space you specify for SWAP.SWAP is a maximum; AOS may not use all of it. However, the space you specify is unavailable for general data storage. If you specify a disk unit for the swap file, only the SWAP.SWAP file can use it.

A Physical Unit for SWAP.SWAP – If you can afford the disk space, you may want to designate a physical unit for the SWAP.SWAP file.

To do this, answer the AOSGEN SWAP question by specifying the disk unit name. For example, if the unit you want to use for the SWAP.SWAP file is DKB0, type

SWAP [2000]: DKB0 ⤶

In this case, only the SWAP.SWAP file can reside on physical unit DKB0. You cannot store any other AOS data files on DKB0.

Note that you can override the AOSGEN specifications for SWAP when you bring up an AOS system. For example

OVERRIDE DEFAULT SPECS [N]? Y ⤶

NUMBER OF BUFFERS IN CACHE [default]? ⤶
SWAP [2000]? 10000 ⤶

Cache of Buffers

Your answer to the CACHE question sets the number of system buffers. These are 512-byte buffers that the system uses for internal I/O. The valid range is 0 through 128. If your system will have a lot of free memory (as it might if it will run only a few processes), a large number of buffers can help cut down on disk requests. But, if there will be a lot of processes, and/or memory contention, swapping will occur anyway. In this case, you may want to specify fewer buffers to free memory for active processes.

For your first system, you will generally want to choose the default. The ideal number of buffers will depend on your system load, the kind of things your system does, and your hardware. To determine the ideal buffer figure, you can time typical application programs.

As with the SWAP specification, you can override the CACHE specifcation when you bootstrap the system. This makes it easy to alter the buffer cache for testing — simply reboot the system, override default specs, and enter the new cache size.

Proceeding

When you have specified the parameters you want, you may want to review the whole system again with the Current command. If any device is wrong, fix it with the Edit command; or, in worst case, delete the device with Delete and Add it again.

Naming the System (N)

After you specify the CPU and all your devices and parameters, name the system. The name can be any valid AOS filename (that is, 1 to 27 alphanumeric characters and special characters ?, _, $, and period (.)). Try to create a descriptive and memorable name because you'll have to type the full pathname to bring the system up until you install it.

To name your system , use the N command:

N ↵

ENTER NEW SYSTEM NAME [none]:

Type the new name. For example

SYS_7.00 ↵

ENTER A COMMAND:

Creating the Specification File

All your device specifications are still in main memory. To write them to a disk file (sys.CSF), use the S command (Create a spec file):

S ↵

CREATING SPECIFICATION FILE

ENTER A COMMAND:

If you do not specify a valid system, AOSGEN does not create the specification file. Instead, it displays an error message and asks for a command. Fix the specification by adding or editing the appropriate controller. Then, try the S command again.

If a file named sys.CSF already exists in directory :SYSGEN, AOSGEN asks if you want to overwrite it (delete it and replace it with the new one). To overwrite sys.CSF, type Y↵; to save the old file, type N↵. Then, AOSGEN gives you another chance to enter a specification file name.

At this point, all your efforts are saved on disk. You could leave AOSGEN, execute it again later with the command X AOSGEN/SAVE/BATCH=sys↵, and have it build a system.

But let's assume you want to proceed.

Building the System

To build the current specification into a tailored AOS system, use the Build command. AOSGEN verifies the configuration and notifies you if it isn't valid. If it is valid, AOSGEN tries to create a specification file, then builds a system. If a file with the name sys.CSF already exists, AOSGEN asks if you want to write over it. Here, this will occur because you already created the specification file.

Type

B ↵

A SPEC FILE WITH THIS NAME ALREADY EXISTS
SHALL I WRITE OVER IT? [N] Y ↵
CREATING SPECIFICATION FILE

DO YOU WANT TO SAVE TMP FILES? [Y] ↵

SYSTEM BUILD IN PROGRESS

AOSGEN invokes first the macroassembler (MASM), then the Link utility to build the system. This takes 10 to 20 minutes. But soon, you'll see the message

SYSTEM BUILD PHASE COMPLETED

ENTER A COMMAND:

The tailored system is ready. Go to the next section.

If the build doesn't work and you see an error message, leave AOSGEN with Q⤶. Check the working directory with the CLI command DIRECTORY⤶; it should be :SYSGEN. Then, check your search list with SEARCH⤶; it should include :UTIL. If the working directory and search list are not what they should be, set them as shown earlier under "AOSGEN Session". Then, to build another system from the specification file, type

```
*) X    AOSGEN/SAVE/BATCH=sys ⤶
```

If the search list is not the problem, type the sys.KS_OUT.pid.TMP file (see Table 4-1) to check for other errors.

Quitting AOSGEN (Q)

The new system has been built, but is still waiting in the wings. To try it, you'll need to leave AOSGEN, patch the new system, shut down the current system, and bootstrap the new tailored system. To leave AOSGEN, type

```
                    Q ⤶
*)
```

If you try to quit AOSGEN without typing a Spec or Build command, AOSGEN warns you that

YOU HAVE NOT CREATED A SPEC FILE FOR THE CURRENT SYSTEM
DO YOU WISH TO CREATE A SPEC FILE [Y]:

To save the work you've done (if any) during this AOSGEN session, press ⤶ for the default, then type a specification file name. (For information on specification file names, see "Creating the Specification File".) To leave AOSGEN without creating a specification file, type N⤶.

You're back in the CLI.

Figure 4-5 is a summary of an AOSGEN session with an IAC-equipped S/140. Figure 4-6 is a session with an ALM-equipped C/350.

```
) SEARCHLIST   :UTIL ↵
) DIR   :SYSGEN ↵
) SUPERUSER   ON ↵
*) X   AOSGEN ↵

        INITIALIZING AOSGEN DATA FILES...
WELCOME TO AOSGEN — TYPE H FOR HELP

ENTER A COMMAND:     H ↵
    ...(text of HELP)...

ENTER A COMMAND:     M ↵
    ENTER NEW MODEL [C/350]:     S/140 ↵

ENTER A COMMAND:     A ↵
    NAME OF DEVICE TO BE ADDED:     DPJ ↵  (or DPI  ↵)
    DEVICE CODE [24]:     ↵

ENTER A COMMAND:     C ↵
    ...(text of current system description)...

ENTER A COMMAND:     E ↵
    NAME OF DEVICE TO EDIT:     DPF ↵
    DEVICE CODE [27]:     ↵

ENTER A COMMAND:     A ↵
    NAME OF DEVICE TO BE ADDED:     MTB1 ↵  (or MTC↵)
    DEVICE CODE [62]:     ↵
    DEFAULT DENSITY [ADM]:     ↵
    MAX BYTE TRANSFER [8K]:     ↵
    SYSTEM DUMP DEVICE [N]:     ↵

ENTER A COMMAND:     E ↵
    NAME OF DEVICE TO EDIT:     CON0 ↵
    CONSOLE TYPE [CRT3]:     TTY ↵
    INPUT BUFFER BYTE LENGTH [96]:     ↵
    OUTPUT BUFFER BYTE LENGTH [128]:     ↵
    CHARACTERISTIC WORD 0 [STANDARD]:     ↵
    CHARACTERISTIC WORD 1 [STANDARD]:     ↵
    CHARACTERISTIC WORD 2
      LINES PER PAGE [STANDARD]:     ↵
      CHARS PER LINE [STANDARD]:     ↵
    CHARACTERISTIC WORD 3 [STANDARD]:     ↵

ENTER A COMMAND:     A ↵
    NAME OF DEVICE TO BE ADDED:     IAC ↵
    DEVICE CODE [65]:     ↵
    IAC BASE LINE NUMBER [0]:     ↵
    IAC DEVICE TYPE [??]:     16 ↵
    LINES [??]:     0 - 15 ↵
    CONSOLE TYPE [CRT3]:     CRT6 ↵
    INPUT BUFFER BYTE LENGTH [96]:     ↵
    OUTPUT BUFFER BYTE LENGTH [128]:     ↵
    CHARACTERISTIC WORD 0 [STANDARD]:     ↵
    CHARACTERISTIC WORD 1 [STANDARD]:     ↵
```

DG-26898T

Figure 4-5. AOSGEN Session for an IAC-equipped Computer System (continues)

```
CHARACTERISTIC WORD 2
  LINES PER PAGE [STANDARD]:  )
  CHARACTERS PER LINE [STANDARD]:  )
CHARACTERISTIC WORD 3 [STANDARD]:  )
INITIALIZATION WORD [STANDARD]:  )

DO YOU WANT TO ADD ANOTHER GROUP OF LINES? [N]  )

ENTER A COMMAND:  A )
  NAME OF DEVICE TO BE ADDED:  LPB )
  DEVICE CODE [17]:  )

ENTER A COMMAND:  A )
  NAME OF DEVICE TO BE ADDED:  BBU )
  FULL OR PARTIAL BACKUP [F]:  )
  ENABLE AUTO RESTART AFTER POWER FAIL [Y]:  )

ENTER A COMMAND:  P )
  SWAP [2000]:  8000 )
  CACHE [0]:  )
  FREQUENCY [10]  )
  SYNC BUFFERS [1]:  )
  ACCESS CONTROL [Y]  )
  DEMAND PAGING [NO]:  )

ENTER A COMMAND:  N )
  ENTER NEW SYSTEM NAME [none]:  SYS_7.00 )

ENTER A COMMAND:  B )
  CREATING SPECIFICATION FILE
 DO YOU WANT TO SAVE TMP FILES? [Y]  )

  SYSTEM BUILD IN PROGRESS

 SYSTEM BUILD PHASE COMPLETED

ENTER A COMMAND:  Q )

*)
```

DG-26898T

Figure 4-5. AOSGEN Session for an IAC-equipped Computer System (concluded)

```
) SEARCHLIST   :UTIL   ↵
) DIR   :SYSGEN ↵
) SUPERUSER   ON ↵
*) X   AOSGEN ↵

        INITIALIZING AOSGEN DATA FILES...
WELCOME TO AOSGEN — TYPE H FOR HELP

ENTER A COMMAND:    H ↵
    ....(text of HELP)..

ENTER A COMMAND:    M ↵
    ENTER NEW MODEL [C/350]:    ↵

ENTER A COMMAND:    A ↵
    NAME OF DEVICE TO BE ADDED:    DPF1 ↵
    DEVICE CODE [67]:    ↵

ENTER A COMMAND:    C ↵
    ...(text of current system description)...

ENTER A COMMAND:    A ↵
    NAME OF DEVICE TO BE ADDED:    MTB1 ↵
    DEVICE CODE [62]:    ↵
    DEFAULT DENSITY [ADM]:    ↵
    MAX BYTE TRANSFER [8K]:    ↵
    SYSTEM DUMP DEVICE [N]:    ↵

ENTER A COMMAND:    E ↵
    NAME OF DEVICE TO EDIT:    CON0 ↵
    CONSOLE TYPE [CRT3]:    TTY ↵
    INPUT BUFFER BYTE LENGTH [96]    ↵
    OUTPUT BUFFER BYTE LENGTH [128]    ↵
    CHARACTERISTIC WORD 0 [STANDARD]    ↵
    CHARACTERISTIC WORD 1 [STANDARD]    ↵
    CHARACTERISTIC WORD 2
      LINES PER PAGE [STANDARD]    ↵
      CHARS PER LINE [STANDARD]    ↵
    CHARACTERISTIC WORD 3 [STANDARD]:    ↵

ENTER A COMMAND:    A ↵
    NAME OF DEVICE TO BE ADDED:    ALM ↵
    I/O PROCESSOR [NONE]:    DCU50 ↵
    LINES [??]:    0 - 15 ↵
    CONSOLE TYPE [CRT3]:    ↵
    INPUT BUFFER BYTE LENGTH [96]:    ↵
    OUTPUT BUFFER BYTE LENGTH [128]:    ↵
    CHARACTERISTIC WORD 0 [STANDARD]:    ↵
    CHARACTERISTIC WORD 1 [STANDARD]:    ↵
    CHARACTERISTIC WORD 2
      LINES PER PAGE [STANDARD]:    ↵
      CHARACTERS PER LINE [STANDARD]    ↵
    CHARACTERISTIC WORD 3 [STANDARD]:    ↵
    INITIALIZATION WORD [STANDARD]:    ↵
```

DG-26899T

Figure 4-6. AOSGEN Session for an ALM-equipped System (continues)

```
    DO YOU WANT TO ADD ANOTHER GROUP OF LINES [N]?      ⤶
ENTER A COMMAND:    A ⤶
    NAME OF DEVICE TO BE ADDED:    LPB ⤶
    DEVICE CODE [17]:     ⤶

ENTER A COMMAND:    A ⤶
    NAME OF DEVICE TO BE ADDED:    MCA ⤶

ENTER A COMMAND:    P ⤶
    SWAP [2000]     8000 ⤶
    CACHE [0 ]     ⤶
    FREQUENCY [10]     ⤶
    SYNC BUFFERS [1]:     ⤶
    ACCESS CONTROL [Y]     ⤶
    DEMAND PAGING [N]     ⤶

ENTER A COMMAND:    N ⤶
    ENTER NEW SYSTEM NAME [none]:    SYS_7.00 ⤶

ENTER A COMMAND:    B ⤶
    CREATING SPECIFICATION FILE
    DO YOU WANT TO SAVE TMP FILES? [Y]     ⤶

    SYSTEM BUILD IN PROGRESS

    SYSTEM BUILD PHASE COMPLETED

ENTER A COMMAND:    Q ⤶

*)
```

DG-26899T

Figure 4-6. AOSGEN Session for an ALM-equipped System (concluded)

Installing Patches

Patches are corrections to a program, made on its .PR or .OL file, or, in the case of the system itself, to the .SY file. AOS revisions usually ship with an update that includes patches. Also, if you are on DG's Software Subscription Service, you will periodically receive AOS updates that include patches.

You should always install patches after generating a new AOS system. (You may also want to install patches after receiving each update — especially if there's a problem you want fixed. Even if AOS software appears to run correctly, it is not complete until the current update's patches — if any — have been installed.)

AOS revisions later than 6.00 include a set of macros (autopatch macros) that make patch installation easy. The autopatch macros patch all AOS programs (like PMGR, CLI, EXEC, and the SED text editor) that need patching.

Before patching, be sure that you have loaded the latest AOS update tape (supplied by DG with revision number n.xx, with xx not 00). For your first system, Chapter 2 or 3 had you load the latest update. But if you have generated a new AOS system without loading the latest update, then you must load the update before patching.

If the latest update is already installed, skip to "How to Install Patches."

Loading an AOS Update

If you received the update on tape, perform step 1 as follows. For an update on diskette, skip to step 2.

1. Get the latest AOS update tape, mount it on a tape unit, and type) SUPERUSER ON ↵

```
*) DIR :↵
*) DELETE/2=IGNORE PATCH_FILES↵              (List file for old patch filenames)
*) LOAD/V/DELETE/L=PATCH_FILES @MTxn:0↵      (x is A, B, or C; n
                                             is the unit number. With the starter
                                             system, use @MTA0:0 ↵.)
.
.  (CLI verifies old files deleted and new files loaded.)
.
```

   Skip to step 3.

2. Get the first (maybe the only) diskette in the update and insert it in unit 0. Then type

```
) SUPERUSER ON↵
*) DIR :↵
*) DELETE/2=IGNORE PATCH_FILES↵          (List file for old patch filenames)
*) LOAD/V/L=PATCH_FILES @DPM0↵
.
.  (CLI verifies old files deleted and new files loaded)
.
*)
```

   Remove the update diskette from unit 0.

3. Type

```
*) DIR PATCH↵

*) QPRINT AOS_UPDATE_NOTICE↵          (Print the update notice file.
                                      If EXEC is not running, use COPY @LPB
                                      (or @LPD) instead of the QPRINT command.
                                      If you have no line printer, use TYPE
                                      instead of QPRINT or COPY.)
```

Read the update notice, to check for any warnings or cautions. If you're curious about the patches themselves, print or type the file, PATCH_FILES. This file has the names of all patch files loaded. Each patch filename includes the AOS revision number, the program filename if known, and "_PATCHES." For example, file 7.00_GHOST.PR_PATCHES has patches for the revision 7.00 Ghost. And file 7.00_AOS_PATCHES has patches for a tailored revision 7.00 AOS system.

How to Install Patches

The autopatch installation macros are easy to use. The rules for using them are as follows.

- You must install patches from PID 2 (the master CLI). This CLI must be at level 0 (use the LEVEL command to check; type POP ↵ as needed).
- The CLI's searchlist must include fewer than 40 characters. You don't need *any* directories on the searchlist to install patches.
- The multiuser environment (EXEC and other sons of PID 2) must be shut down. There should be only two processes: PMGR and CLI.

- The working directory must be :PATCH. The patch program PATCH.PR and current patch files must be in this directory. (These files are put here by default.) During the patching process, the macros will copy PATCH.PR to :UTIL if needed.
- AOS program files must be accessible from the directory where they were installed (e.g., EXEC.PR must be accessible from :UTIL and PMGR.PR must be accessible from the root). Also, each program symbol table (program-name.ST) must be accessible from the program's default directory. (Symbol table and program files are loaded into the same directory by default.)

  If a program file should be patched, and the macros cannot find the program or its symbol table, they will not patch that program on this patch run. There will be a message to this effect in the listing file and on the console screen.

The form of the command to start the autopatch macros is

INSTALL_rev_PATCHES *[/L=listfile][/ROOT=subtree] [system-pathname]*

where

rev — is the AOS revision — for example, 7.00, 8.00, or 9.00.

/L=listfile — specifies the listing file pathname, to store copies of console messages. These messages include comments in each patch that explain the reason for the patch. If you omit this switch, the listing filename will be INSTALL_rev_LOG in directory PATCH. If this file already exists, new messages will be appended to it.

/ROOT=subtree — specifies an alternate root directory (useful for system programmers, or any site that has an alternate tree structure beneath the root). If you omit this switch, the default root is :. Generally, omit this switch.

system-pathname — is the pathname of the AOS system you want to patch; for example, :SYSGEN:SYS_7.00.SY.

If you omit the system-pathname, the autopatch macros will not patch the AOS system. It's very important to patch the system. Always include the tailored system pathname, even if you think the system has already been patched.

A typical autopatch sequence takes 5 to 15 minutes. When you're ready to do it, type the INSTALL command. For example:

```
) DIR   :PATCH ↵
) INSTALL_7.00_PATCHES   :SYSGEN:SYS_7.00.SY ↵
.
.  (patch messages)
.
)
```

This command patches all AOS files that need patching, including the system, SYS_7.00.SY. It records console messages in file :PATCH:INSTALL_7.00_LOG.

In addition to console messages, PATCH creates a patch history file (filename ends in .PH) for every program patched. The patch history files are stored in directory :PATCH:PATCH_HISTORY. Patch history files can help you keep track of patching activity.

Whenever a program is patched, the patch updates the program revision number, so you can get the last update level via the REVISION command. For example:

```
) REV   :PMGR.PR ↵             (Check program revision.)
07.00
) INSTALL_7.00_PATCHES ↵       (Install patches.)
.
.  (messages)
.
```

) REV :PMGR.PR ↲ (Check revision again.)
07.01 (Revision number shows
) update.)

After you patch, the applied patches will not take effect until you shut the system down and bring it up again. So, to try your software, shut down, and start up again (skip ahead to the "Testing the New System" section).

Autopatch Errors

The autopatch installation macros create files in a temporary directory, :UDD:OP:?PATCH.pid.TMP. Then, after patching is complete, they delete the temporary directory. If for any reason the macros do not finish normally, you might want to delete this directory before retrying the procedure.

Error messages have the form

ERROR
message

The message text should enable you to fix the problem and retry the installation. If not, make sure you're following the rules described above.

Patching Specific Programs

The PATCH program — used by the autopatch installation macros — installs patches in individual program files. Generally, we recommend that you use the autopatch macros. Under some circumstances, however, you (and/or a DG system engineer) may want to install an individual patch using the PATCH program.

As with autopatch, the system won't let you patch an open file. And, if you patch a program that's already running, the patch won't take effect until the program is restarted from disk.

Before using the PATCH utility, make :PATCH the working directory. The PATCH command line has the form

X PATCH/Y/P=patch-pathname/T=program-pathname *[/L=pathname]*

where

/Y	specifies Yes (tells PATCH to install the patch).
/P=patch-pathname	gives the pathname to the patch file. This patch file must be written according to PATCH rules, sketched in the PATCH help file (HELP *PATCH ↲). Generally, we suggest that you keep all patches in the :PATCH directory.
/T=program-pathname	specifies the target program to patch. You must include the .PR suffix, if the program has one.
/L=pathname	selects a disk file to store console messages from PATCH. These messages will be sent to your console screen in any case.

It's a good idea to copy a program to a backup file before patching it. An example of a manual patch sequence is

(copy program to a backup file)

*) DIR :PATCH ↲

(load the patch file into :PATCH)

*) X PATCH/Y/P=DG_UTILITY.PATCHFILE/T=:UTIL:DG_UTILITY.PR ↲

.
. (PATCH applies patches)
.

*)

If the patch you want to apply uses symbolic addresses, the program symbol table file (program-name.ST) will be in the same directory as the program file. The two files are placed in the same directory by default when you create a program with LINK or load an AOS revision.

For every program that it patches, PATCH creates a patch history file, named program-name.PH, in the working directory, usually :PATCH.

If the PATCH program reports an error, make sure that your command line syntax is correct, and that the symbol table file (.ST) exists in the program's directory. If a patched program does not work properly, you can try again with a new copy of the backup version. (Don't patch the backup version itself.)

Testing the New System

First, shut down the current system. (If the multiuser environment is up, make sure all users are logged off first). Type

*) BYE ⤶
DO YOU REALLY WANT TO SHUT THE SYSTEM DOWN? Y ⤶
STARTING SYSTEM SHUTDOWN date

SYSTEM SHUTDOWN

Boot the new system by

- typing nnH in response to the *!* prompt if you have an S/20, S/120, S/280, or a DESKTOP GENERATION system; e.g., 27H, 26H;
- typing 11A and 1000nn⤶ in response to the *!* prompts if you have an S/140; then type 1000nnL;
- setting the data switches to 1000nn where nn is the device code, and by lifting and releasing the RESET and PR LOAD (PROG LOAD) switches if your computer has hardware data switches;
- typing RESET⤶ and BOOT 27⤶ (or BOOT 24⤶ for model 6236) in response to the *SCP-CLI>* prompts if you have an MV/8000.

Then, answer the following questions.

SPECIFY EACH DISK IN THE LDU
DISK UNIT NAME? DPF0 ⤶ (or DPJ0 ⤶ for model 6236, or DPN0 ⤶ for DESKTOP GENERATION systems, or other unit name)

DEVICE CODE? ⤶

SYSTEM PATHNAME? :SYSGEN:sys.SY ⤶ (e.g., :SYSGEN:SYS_7.00.SY⤶)

sys is your system name. You must type the full pathname from the root directory; shown as :SYSGEN:SYS_7.00.SY ⤶.

DATE (MM/DD/YY)? (Type the current date; e.g., 4 15 85 ⤶)
TIME (HH:MM:SS)? (Type the current time; e.g., 16 30 ⤶)

OVERRIDE DEFAULT SPECS? [N]

It's asking about the specifications you just gave to AOSGEN for the new system. You don't want to override, so press

⤶
MASTER LDU: name

A pause occurs here; then

AOS CLI REV n date time
)

The new tailored system is running. Skip the next section.

If the New System Doesn't Come Up or Work Properly

If the new system doesn't come up, there will usually be an error message that describes the problem. The solution may be simply a matter of editing a device in the specification file. If the message is *FATAL AOS ERROR,* note the figures displayed. Then, run ESD (Emergency Shutdown) by typing RESET↵ and START 14↵ if your system is an MV/8000, by setting the data panel switches to 14 (switches 12 and 13 up) and pressing the START switch if your system has data switches, or by typing 14R if your system is an S/20, S/120, S/140, or an S/280. Then boot the starter system.

To bring up the starter system, proceed as shown above, but answer ↵ to the *SYSTEM PATHNAME?* question. Then, if you think the problem is in the AOSGEN specification, execute AOSGEN with the /DEFAULT=sys switch, fix the device specification, give the original name in the Name command, use the Spec command to overwrite the old specification file, use the Build command again, and proceed as described in the "Testing" section.

If the new system fails again, boot the starter system and type the panics file via command COPY @LPA :UTIL:AOS.PANICS.SR↵. See Chapter 6 for information on interpreting the panic values.

A Fast Checkout for the Tailored System

If you know that your AOSGEN specification file is okay (perhaps because you've used it to build a system that works), you can skip this section.

If your AOSGEN specification file is new, you may want to check a few basic things. (You won't really know that the system works until you've brought up the multiuser environment. For information on bringing up the multiuser environment, see Chapter 5.) As a basic test, try a few CLI commands:

```
) SEARCH  :UTIL ↵
) DIR  : ↵
) FILES/AS/S ↵
WARNING: FILE ACCESS DENIED, FILE =
) SUPERUSER  ON ↵
*) FILES/AS/S ↵

     ...(sorted list of files in directory :)...
```

These commands set the search list to :UTIL, check file access controls, and show you the FILESTATUS /S (SORT) and /AS (Assortment) switches.

Now, if you have any other formatted LDUs, try to initialize and release them. For example, assume you have an LDU with disk unit name DPF1 that you named UDD1. With unit DPF1 ready, type

*) INITIALIZE @DPF1 ↵	Initialize using unit name.
UDD1	System displays LDU name.
*) RELEASE UDD1 ↵	Release using LDU name.

Try this with all your LDUs. If the commands work, your disk controllers are properly generated. If a command doesn't work, you forgot to enter a controller specification or you entered it incorrectly. In this case, the AOSGEN specification file may need editing.

Turn your line printer ON, make sure the paper is properly aligned, put the line printer on line, and type

```
*) COPY  @LPB    :UTIL:AOS.PANICS.SR ↵    (or @LPA, @LPC, or @LPD)
*)
```

This shows that the line printer works. It also gives you a copy of the PANICS file, which is useful when you attempt to describe fatal AOS errors.

Type

*) CHAR/DEFAULT ⤶
... (characteristics)...
*)

This shows you the default console characteristics AOSGEN set up for the system console.

Type

*) PROCESS/DEF/IOC=@CON2 :CLI ⤶
*)

This creates a CLI process to run on console CON2. (The @ is shorthand for the peripherals directory.) Go to the console connected as CON2 (console line 0) and verify the baud rate and parity settings on the back (if it's a video-display console). Make sure it is on line.

Wait a few moments, then press ⤶ a few times. If the CLI prompt appears, this means that you have at least one user console line connected and generated properly. If nothing happens on CON2, the hardware may be wrong, or you may have made a mistake in AOSGEN.

To terminate the process on CON2, type BYE⤶ on CON2's keyboard.

Next, mount a blank magnetic tape (with the write ring in) on unit 0. Put the tape unit on line. Type

*) DIR : ⤶
*) DUMP/V @MTx0 ⤶ (x is A, B, or C, depending on tape unit model)

...(system verifies files dumped)...

CTRL-C CTRL-A (press CTRL-C, then CTRL-A)

ERROR: CONSOLE INTERRUPT

*) REW @MTx0 ⤶ (x is A, B, or C, depending on tape unit model)

This shows that the tape drive for unit 0 is okay. It also shows you how to interrupt CLI commands with CTRL-C CTRL-A. Now, dismount the tape.

You've done nearly all the testing that you can do without bringing up the multiuser environment.

Installing the Terminal Emulator (DESKTOP GENERATION Model 10/SP Only)

The first time you bring up AOS on a DESKTOP GENERATION Model 10/SP, you should also install a DASHER D200 terminal emulator file on the hard disk. Remove the diskette from its unit and insert the diskette labeled "D200 EMULATOR" into the primary unit.

The D200 EMULATOR diskette contains emulator files for various languages and for both the color and the monochrome monitors. The files for color monitors begin with 'C' and the files for monochrome monitors begin with 'M'.

The emulator filename indicates the language whose special characters are supported by the emulator in 7-bit mode. When an emulator runs in 8-bit mode, all the special characters in the language are available in 8-bit codes. In 7-bit mode the special characters of only one language are available. The following list correlates the emulator name with the supported languages.

Language	Emulator (Color)	Emulator (Monochrome)
Danish/Norwegian	CD211DAN.TX	MD211DAN.TX
English	CD211AMUK.TX	MD211AMUK.TX
French	CD211FRAN.TX	MD211FRAN.TX
German	CD211GER.TX	MD211GER.TX
Italian	CD211ITAL.TX	MD211ITAL.TX
Spanish	CD211SPAN.TX	MD211SPAN.TX
Swedish/Finnish	CD211SWED.TX	MD211SWED.TX
Swiss German	CD211SWISG.TX	MD211SWISG.TX

1. Identify the emulator that corresponds to your keyboard character set and monitor type. Then type the following commands to move it onto the system disk.

   ```
   *) INITIALIZE/S  @DPM0 ↵

   *) DIR  [!STRING] ↵

   *) MOVE/V/D  :  emulator-filename ↵        (Specify the emulator file you want;
                                               for example, MD211GER.TX.)
   .
   .                 (It verifies file(s)
   .                 moved.)

   *) MOVE/V/D  :  RELEASE+ ↵                 (Move the emulator Release Notice to
                                               the system disk.)

   *) DIR  : ↵

   *) RELEASE  [!STRING] ↵
   ```

2. Remove the diskette from the primary unit and store it safely. You may need it again to copy another emulator file to the hard disk.
3. You should read the emulator Release Notice to get the current information concerning the emulator.
4. When AOS starts up, it checks whether the system console is monochrome or color. If monochrome, AOS looks for an emulator file named MD200.TX; if color, it looks for an emulator file named CD200.TX. If it can't find either file, it comes up anyway, without an error message. If no emulator is loaded, the function keys and lowercase characters will not function. If AOS finds the correct file, the emulator is loaded automatically. The emulator cannot be changed while AOS is running (although the MODE of an emulator can be changed). The only way to load a new emulator is to repeat step 1, shut down, power down, and cold start.

   Since AOS looks for a certain filename when starting up, you must either rename the emulator file you just moved onto the system disk, or create a link with the default name to the emulator you want (this preserves the emulator filename on the disk). For example, assume you have a monochrome monitor. Also assume that you moved the French emulator (filename MD211FRAN.TX) onto the system disk as explained above. Then type the following commands:

   ```
   *) DIR  : ↵

   *) CREATE/LINK  MD200.TX  MD211FRAN.TX ↵
   ```

Thereafter, AOS will use the link resolution file as the emulator to load when it comes up. If you ever want to change the file used, you can delete the link (MD200.TX), and then recreate the link MD200.TX to the new file. Remember that you must shut down your system, power down, and cold start in order to load the new emulator.

When you create a link to the emulator file, be sure you type the emulator filename correctly. If you mistype the name, AOS cannot find the emulator to load it. If you make a mistake, DELETE the linkname (not the emulator!) and recreate the link.

You've copied the emulator file(s) onto your system disk and created a link to the one you want to use. From now on, AOS will load the one that you created a link to.

5. Now you must shut down AOS, power down, and restart to load the emulator. Type the following commands:

6.) BYE ↵

7. *DO YOU REALLY WANT TO SHUT THE SYSTEM DOWN?* Y ↵

   SYSTEM SHUTDOWN

   !

8. Turn computer unit power off and on again.

9. *!* 26H (Type 26H)

   SPECIFY EACH DISK IN THE LDU

10. *DISK UNIT NAME?* DPN0 ↵ (Type DPN0 ↵)

11. *DEVICE CODE?* ↵

12. *SYSTEM PATHNAME?* ↵

    AOS REV n

13. *DATE (MM/DD/YY) ?* 5 15 85 ↵ (Type the current date.)

14. *TIME (HH:MM:SS) ?* 15 20 ↵ (Type the correct time — 24-hour clock.)

15. *OVERRIDE DEFAULT SPECS? [N]* ↵

    AOS CLI REV n date time

)

AOS is up and running. To test for the correct emulator, try typing a lowercase letter (like q). If you can make q appear in lowercase, an emulator is loaded and working correctly. Continue to the next section. (If the q doesn't appear in lowercase, return to step 1.)

The emulator comes up in 8-bit mode. To put it in 7-bit mode, use the macro ENABLE7BIT.CLI, supplied in directory :UTIL. To return the emulator to 8-bit mode, use macro DISABLE7BIT.CLI, also supplied in :UTIL. For example:

26H (Bootstrap system.)

.

.

.

AOS CLI REV n date time

) UP ↵

```
) :UTIL:ENABLE7BIT ↵                         (Enable 7-bit mode.)
.
.                                            (Run in 7-bit mode.)
.
) :UTIL:DISABLE7BIT ↵                        (Disable 7-bit mode,
                                             returning to 8-bit.)
```

In 7-bit mode, certain characters (like square brackets and commercial at signs) have special meanings. Thus, you should always re-enable the 8-bit mode after executing the user program(s) that require the 7-bit character set.

Making a Tailored System Tape

After testing the new system, you should always make a system tape that includes your tailored system. To make a tailored system tape, follow these steps.

- Get a blank tape, with the write ring in. Mount and thread it on tape unit 0, the one you used for the initial AOS load. Make sure power is ON, then press BOT and ON LINE.
- Type

```
) SUPERUSER  ON ↵
*) DIR  :SYSGEN ↵
*) SYSTAPE  @MTx0  sys.SY ↵     (x is A, B or C, depending on the unit type. sys is your
                                 tailored system name.)
```

  Via commands in the SYSTAPE.CLI macro, the system makes you a tailored system tape.

- When the CLI SUPERUSER prompt returns, rewind the tape:

```
*) REW  @MTx0 ↵
*)
```

  Unless you want to install the system now (described next), remove the tape from the unit, remove the write ring from the tape, and clip the cover on the tape.

Any system tape you make using the SYSTAPE macro has the same format as the AOS system tape supplied by DG. However, it has your tailored, patched system in file 5 instead of the starter system. And, it has all files and programs currently in your root (:), :UTIL, :HELP, and :SYSGEN directories — including the ones your site created.

If you ever need to rebuild a blank LDU using this tape, you can abbreviate the procedures given in Chapter 2 or 3. You won't need to run AOSGEN again; in fact, after you load tape file 7, you'll come up in AOS with the root, :UTIL, :HELP, and :SYSGEN directories restored as of the time the system tape was made.

Making a Tailored System Diskette

If you received AOS on diskettes, you should now make a backup diskette of your tailored, patched system. Follow these steps:

1. Get a new, blank diskette and write-enable it.
2. Insert the diskette in a free unit (for example, the primary unit).
3. Copy the tailored system to the diskette as follows (do not INITIALIZE the diskette):

```
) SUPERUSER  ON ↵
*) DIR  :SYSGEN ↵
```

*) COPY @DPxn sys.SY ↵ — x is I or K for a high-density diskette; it is D for a low-density diskette. n is the unit number. sys is the name of your tailored system. For example,

COPY @DPK1 SYS_7.00.SY

(copies to diskette)

*)

Unless you plan to install this system, remove the diskette, strip the write-enable tape, and apply a paper label (if there is none). Then — with a felt-tipped pen — write a desription of the system on the label (for example, "Tailored AOS system, rev 7.00, patched to rev 7.02, name SYS_7.00.SY, May 15, 1985." Return the diskette to its outer envelope and store it safely, away from strong magnetic fields.

You'll need this diskette to install your tailored system, or if you ever need to restore your master LDU from scratch. To restore the LDU, proceed as in Chapter 3 — but in Step 24, use your tailored diskette instead of the DG system diskette. Then continue in Chapter 3.

Installing the Tailored System

You can install the system or leave it uninstalled. Installation is not required. The trade-offs are as follows.

- Advantages: installation allows you to bring up AOS by pressing ↵ at the *SYSTEM PATHNAME?* query, instead of typing a pathname like :SYSGEN:REV_7.00.SY↵. Installation puts a copy of the system into "invisible" space on the LDU, which means that it cannot be deleted.
- Disadvantages: The system is invisible, so you can't tell which system is running, or which is installed if you have multiple systems in :SYSGEN. This also means that you can't patch the installed system. You must install patches (which patches the version in :SYSGEN), make another systape or diskette, and install the patched version.

Only one system can be installed on an LDU. So if you install a system, it will overwrite any existing system on the LDU. This is why we had you test and patch the system — to make sure it was okay before installing it over the other system.

In any case — regardless of your decision — write the system pathname (e.g., :SYSGEN:REV_7.00.SY) on a paper label and stick it onto the system console in a conspicuous place. If you install the system, you might also write "INSTALLED date" on the label. If you don't want to install the system now, skip the rest of this section and return to it if you want to install the system.

If you decide to install the system, follow these steps.

1. For tape, mount and thread your tailored system tape on unit 0 (unless the tape is already mounted and threaded). For diskette, insert your tailored system diskette in a unit (unless it is already inserted).
2. Shut down the current system, if one is running, with the command BYE↵; then type Y↵. If the multiuser environment is running, make sure all users are logged off first.
3. Bootstrap the Installer from tape or diskette according to the bootstrap instructions at the beginning of this chapter. But, when it asks *SYSTEM PATHNAME?* type

   INSTL ↵

   AOS INSTALLER REV n

   ENTER ALL UNITS IN LDU:
4. *DISK UNIT NAME?* DPF0 ↵ (or DPJ0 ↵, or DPI0 ↵, or DPN0 ↵ — type your system disk unit name)
5. *DEVICE CODE?* ↵

   —DISK BOOTSTRAP INSTALLED

6. *INSTALL A SYSTEM BOOTSTRAP?* N ↵
7. *INSTALL A SYSTEM?* Y ↵
8. *FROM MAG TAPE (M) OR DISKETTE(D)*
9. For tape, type M↵. For diskette, go to step 10.

9a. *FROM MT-0, FILE #* 5 ↵

(Installer copies tailored system to LDU.)

Go to step 11.

—SYSTEM INSTALLED
DONE!

10. For diskette, type D ↵.
DISKETTE UNIT NAME ?

10a. Type the name of the unit that holds the system diskette; for example, DPK1 ↵ (or DPI ↵)

(Installer copies tailored system to LDU.)

—SYSTEM INSTALLED
DONE!

11. Now, bootstrap the system according to the bootstrap instructions at the beginning of this chapter.

SPECIFY EACH DISK IN THE LDU:

12. *DISK UNIT NAME?* DPF0 ↵ (or DPI0 ↵, or DPN0 ↵)
13. *DEVICE CODE?* ↵
14. *SYSTEM PATHNAME?* ↵

(pause)

15. *DATE (MM/DD/YY)?* (Type the current date; e.g., 4 15 85 ↵)
16. *TIME (HH:MM:SS)?* (Type the current time; e.g., 16 45 ↵)

17 *OVERRIDE DEFAULT SPECS [N]?* ↵

MASTER LDU: name

(pause)

AOS CLI REV n date time
)

Your tailored system is installed and running.

Generating Other AOS Systems

After you generate an AOS system, you can use the sys.CSF specification file as a base for all future systems.

There are two main reasons for generating a new AOS system: to change a specification (as when you acquire a new device); to install a new AOS revision.

In either case, to generate a new system, use the /DEFAULT switch. For example,

*) X AOSGEN/DEFAULT=SYS_7.00 ↵

AOSGEN warns you if any interrevision changes may cause problems.

When you name a new AOS system, you decide whether the old system will be deleted. If you give the new system the same name as an old one, the old system will be deleted (after AOSGEN gets confirmation). If you give the new system a different name, the old system will continue to exist in directory :SYSGEN.

If an old system is defective, or if its specifications are obsolete, you will probably want to delete it. You can use the CLI DELETE command for this, or your can run AOSGEN and give the new system the old name, whichever is most convenient. If you include the AOS revision in the system name, all names may differ. Therefore, you will have to use the DELETE command to remove the old systems. Do not delete the original specification files (with .CSF and .SSF suffix) that you will use as a base for new systems.

See Chapter 14 for information on loading new revisions of AOS.

Creating AOS PREGEN Diskettes for DESKTOP GENERATION Systems

This section tells how to create a set of AOS PREGEN diskettes, which can be used to install and run AOS PREGEN on a DESKTOP GENERATION system.

PREGEN AOS (model number 30506) is designed specifically for DESKTOP GENERATION systems. PREGEN AOS provides a runtime environment for CEO and programs written in high-level languages like COBOL and PL/I. PREGEN AOS features a friendly user interface, with a simplified Disk Formatter, Installer, FIXUP, and a pregenerated AOS system. It includes a number of CLI macros that ease system operations, like installing software, doing backups, and creating user profiles.

If you have SYSGEN AOS (model 30505), you can create your own version of PREGEN AOS. This allows you to set up a PREGEN with an AOS system that differs from the standard one; for example, it could support a synchronous communication line, which the standard PREGEN system doesn't.

(Any DESKTOP GENERATION computer for which you create PREGEN diskettes must be able to run AOS: it must be a Model 10/SP (floating point), Model 20, or Model 30. It must have at least 0.5 Mbyte of main memory. And it must have a Model 6271 15-Mbyte hard disk unit or a Model 6301 38.6-Mbyte hard disk unit, or a Model 6336 71.2-Mbyte hard disk unit, and at least one Model 6267 5-1/4 inch, 368-Kbyte diskette unit.)

Your AOS model 30505 contains all the software and macros needed; it also has several macros to help you create a custom PREGEN. All these files are in directory :UTIL:DTOP_ROOT.

Creating the PREGEN System Diskettes

1. First, create a control point directory for the PREGEN build, and make it your working directory:

   ```
   ) CREATE/MAX=100000   desktop-directory-name ↵
   ) DIR   desktop-directory-name ↵
   ```

2. To find the required files, set the searchlist as follows:

   ```
   ) SEARCHLIST   :UTIL:DTOP_ROOT   :UTIL   :SYSGEN ↵
   ```

3. The next step is to build the AOS system. The specs for the standard PREGEN AOS system are in file :SYSGEN:DTOP_DEFAULTSPEC.CSF; you can type this file to see them. If you want the standard system, then you don't need to run AOSGEN; skip to step 3a.

   If you don't want the standard AOS system, run AOSGEN and have it create the spec file (don't build a system, just create a spec file). When you execute AOSGEN, you might want to use the switch /DEFAULT=DTOP_DEFAULTSPEC to provide a useful base of specs, and then change only the one(s) you want.

3a. Next, you'll run the DTOP_STARTER macro to build the system and apply patches. For convenience (to avoid tying up a terminal for 10-20 minutes), we suggest you run the macro in batch.

To build a standard system, omit an argument to DTOP_STARTER; to build a custom system, use the name of your custom spec file as an argument. For example, type

) QBATCH/NOT/QOUT==Q.OUT DTOP_STARTER ↵ (to build the standard PREGEN system)

or

) QBATCH/NOT/QOUT==Q.OUT DTOP_STARTER spec-name ↵ (to build a custom PREGEN system; spec-name is the filename of the spec you just created with AOSGEN)

The DTOP_STARTER macro now builds the AOS system in the directory in which you are building the system, the "desktop" directory. This system is always named DTOP_STARTER.SY (it overwrites any existing DTOP_STARTER.SY), regardless of the spec filename. Then, the macro installs system patches for friendly dialog; and it patches a copy of EXEC to allow any user to issue EXEC commands (normally, only the username OP can issue EXEC commands).

4. When the macro is finished, type the Q.OUT file to be sure the system was built and patched correctly. If the system did not build, type the ?+OUT.TMP file to locate the errors. After identifying the problem(s), delete the batch Q.OUT file and AOSGEN ?+TMP files. Then rerun step 3.

   When the system builds and patches correctly, and EXEC patches correctly, you are ready to generate the diskettes.

5. A macro called CREATE_DTOP_MEDIA creates the AOS PREGEN system file structure in your desktop directory. To create this file system structure and start making diskettes, just type the macro name. To create the structure without starting to make diskettes, use the /N switch. For example, type

) CREATE_DTOP_MEDIA ↵ (to create file structure and start making diskettes)

) CREATE_DTOP_MEDIA/N ↵ (to create file structure without making diskettes)

6. When you start making diskettes, have nine 5-1/4 inch diskettes ready (five for AOS software; four for other software products). The diskettes must be hardware formatted (as are those supplied by DG); but they need not be formatted with the Disk Formatter.

   The CREATE_DTOP_MEDIA macro prompts you for each diskette, formats it if needed, and copies files to it. This macro requires only that you use the primary diskette unit (unit 0, @DPM0).

   When the macro has finished, it will have produced five PREGEN AOS diskettes. We list the contents of these diskettes as follows:

Diskette 1, in AOS LDU Format

Friendly Disk Formatter (DFMTR) for the hard disk
Friendly Installer (INSTL) to install AOS system
Friendly FIXUP for the hard disk
Friendly SYSBOOT (installed on diskette)
DTOP_STARTER.SY (installed on diskette)

Diskette 2, in CLI DUMP Format

CHECKLFD.PR
ERMES
CLI.<PR OL>
GHOST.<PR OL>
INSTALL.1.CLI *
INSTALL.2.CLI *
INSTALL.CLI
MMOVE.<PR OL>
PMGR.PR
VALIDATE_PRODUCT.CLI *

Diskettes 3, 4, and 5 in CLI Labeled DUMP Format

Friendly FIXUP for the hard disk
Friendly macros:
- BELL.CLI *
- BROADCAST.CLI
- CHECK_PRINTER.1.CLI *
- CHECK_PRINTER.CLI *
- CLEAR_SCREEN.CLI *
- CLI_RESTORE.CLI *
- CLOSE.CLI
- CONFIGURE.CLI
- CREATEDIR.CLI
- CREATETEXT.CLI
- CREATETEXT_MSG.CLI *
- DISABLE7BIT.1.CLI *
- DISABLE7BIT.CLI
- DOWN.CLI
- ENABLE7BIT.1.CLI *
- ENABLE7BIT.CLI
- FSTAT.CLI
- FULL_BACKUP.CLI
- GET_MODEM.CLI *
- GET_VALID_ANSWER.CLI *
- GET_VALID_RANGE.CLI *
- HELPV.CLI
- HELPV1.CLI
- INC_BACKUP.CLI
- INC_PCOPY.CLI
- LOGON.CLI
- MMOVE_RESTORE.CLI *
- OPEN.CLI
- PCOPY
- POSS.CLI
- POSS1.CLI
- PRINTER_ALIGN.CLI
- PRINTER_CONTINUE.CLI
- PRINTER_REDEFINE.CLI
- PRINTER_STOP.CLI
- PROFILE.CLI
- PROFILE_CREATE.CLI *
- PROFILE_DELETE.CLI *
- PROFILE_RENAME.CLI *
- PROTECT.CLI
- RESTORE.CLI

* This macro is called by another macro; don't execute it directly because the results will be incomplete.

SETUP CLI
UNPROTECT.CLI
UP.CLI
VALIDATE_COUNT.CLI *
VALIDATE_SPACES.CLI *
VALID_NUMBER.CLI *
WAIT_FOR_NO_PORT.CLI *
WAIT_FOR_PORT.CLI *
WHOS.CLI

UPD:OP (operator profile)

UTIL
EXEC.<PR OL>
FCU.PR
friendly DFMTR.PR for the diskette
friendly FIXUP.<PR OL> for the diskette
MMOVE.<PR OL>
PREDITOR.PR
QCMP.PR
XLPT.PR
XPLT.PR

AOS
INSTALL.AOS.CLI

HELP
All CLI command files
All CLI pseudomacro files
CLI.TPC.1_SWITCH
CLI.TPC.2_SWITCH
CLI.TPC.AFTER_SWITCH
CLI.TPC.CLI_INPUT
CLI.TPC.CONDITIONALS
CLI.TPC.CONTROL_CHARS
CLI.TPC.CURSOR_CONTROL
CLI.TPC.ENVIRONMENT
CLI.TPC.EXCEPTIONS
CLI.TPC.FILENAMES   CLI.TPC.GENERIC_FILES
CLI.TPC.I_SWITCH
CLI.TPC.LINKS
CLI.TPC.L_SWITCH
CLI.TPC.MACROS
CLI.TPC.M_SWITCH
CLI.TPC.NEWLINE
CLI.TPC.PATHNAMES
CLI.TPC.P_SWITCH
CLI.TPC.QUEUES
CLI.TPC.Q_SWITCH
CLI.TPC.SWITCHES
CLI.TPC.TEMPLATES
CLI.TPC.TOPICS

* This macro is called by another macro; don't execute it directly because the results will be incomplete.

While the PREGEN directory structure produced by steps 1 through 6 exists, you can use another macro that allows you to create just *one* of the four AOS PREGEN diskettes. For example, you might want to create and install a different AOS system (diskette 1). Another macro — CREATE_DTOP_FLOPPIES — lets you do this.

To use CREATE_DTOP_FLOPPIES, enter your desktop directory and set the searchlist as shown in steps 1 and 2 above. Then type CREATE_DTOP_FLOPPIES and append one of the following switches:

/1 To make diskette 1.

/2 To make diskette 2.

/345 To make diskettes 3,4, and 5.

7. To complete the PREGEN environment, you must make some product diskettes that are shipped with the model 30505 diskettes. These are the INFOS®, SORT, and MP/BASIC diskettes. For Model 10/SP, you must also copy the EMULATOR diskette. Each product has one diskette, except MP/BASIC, which has two.

If you received your software on cartridge tape, skip to step 9.

If you have two diskette units, skip to step 8. With one diskette unit, continue. Before copying diskettes, you must copy the contents of each diskette onto your hard disk.

7a. Create a CPD directory called PRODUCTS and make it your working directory:

```
) CREATE/MAX=100000   PRODUCTS ⤶
) DIR   PRODUCTS ⤶
```

7b. Insert a product diskette into the unit and copy it:

```
) COPY   product-name   @DPM0 ⤶
```

(For product-name, use the name on the diskette label; for example, SORT.)

Repeat this step (7b) for each single-product diskette. For MP/BASIC, copy each MP/BASIC diskette to a separate file. For example, with the first MP/BASIC diskette in unit 0, type

```
) COPY   MP_BASIC1   @DPM0 ⤶
)
```

Remove the first MP_BASIC diskette and insert the second. Type

```
) COPY   MP_BASIC2   @DPM0 ⤶
)
```

Remove the last product diskette copied.

7c. Insert a blank, hardware-formatted diskette. It need not be formatted with the Disk Formatter. Copy each product file onto a diskette:

```
) COPY   @DPM0   product-name ⤶
```

Repeat this step for each product file.

7d. When done, return to the desktop directory:

```
) DIR   ^ ⤶
```

And go to step 10.

8. If you have two diskette units, insert a DG-supplied product diskette in unit 0 and a blank, hardware-formatted diskette in unit 1. (The copy diskette in unit 1 need not be formatted with the Disk Formatter.) Make sure the source diskette is in unit 0, not 1!

   Then, copy the product:

) COPY @DPM1 @DPM0 ↵

   Repeat this step (8) for each of the product diskettes listed in step 7.

   And go to step 10.

9. If you received your software on cartridge tape, you have the diskette images already on the system disk. They are in :UTIL:DTOP_ROOT:DTOP_SIB. The four files in this directory must each be copied to a diskette.

   Insert a blank, hardware-formatted diskette. It need not be formatted with the Disk Formatter. Copy each product file onto a diskette:

) COPY @DPM0 product-file ↵

   Repeat this step for each product file, where product-file is INFOS_IMAGE, SORT_IMAGE, MPBASIC_IMAGE.A, and MPBASIC_IMAGE.B in turn.

10. You're done. Someone can take the PREGEN diskettes you've just created to a DESKTOP GENERATION system and install all the software there. Installing PREGEN on DESKTOP GENERATION systems — and other PREGEN operations — are described in a separate manual: *Using AOS on DESKTOP GENERATION™ Systems.*

11. To conserve disk space, you might want to delete the desktop-directory-name (you can use the SPACE command to check its space requirements). If you decide to do this, first copy the AOS system and spec file to directory :SYSGEN; then do the delete with the following commands:

```
) SUPERUSER ON ↵          (Provide write access to :SYSGEN)
*) MOVE/V :SYSGEN DTOP_STARTER.SY +.CSF +.SSF ↵
.
. (verifies)
.
*) DIR ^ ↵                (Move up to desktop directory's parent.)
*) DELETE desktop-directory-name:# ↵
.
.
.
*)
```

   Later on, to create another set of PREGEN diskettes, repeat the previous steps (to create a system with the same specs, you can specify the old spec file, which you moved to :SYSGEN).

   If you don't delete the desktop directory, you can create PREGEN diskettes very easily, via the steps described just before step 7, step 8, and step 9.

What Next?

If this is your first system, you will want to create the multiuser environment. If this is not your first system, you may want to bring up EXEC and user processes, and perhaps run a few applications programs.

End of Chapter

Chapter 5
Creating the Multiuser Environment

Read this chapter

- when you have generated and tested your first tailored AOS operating system and want to create an environment where many people can use it;
- whenever you want to create a brand-new multiuser environment, or some useful macros.

This chapter leads you through the steps needed to create a multiuser environment. It assumes that a tailored AOS operating system has been generated, tested, and is running.

The AOS multiuser environment is based on two utility programs:

PREDITOR, the user profile editor, which creates individual *profiles* for each user;

EXEC, the executive program that supervises user logon and logoff according to PREDITOR profiles, and manages printing and batch queues.

Using PREDITOR, you create a profile for each person that will use the system. Next you initialize EXEC, and try it. Then, you edit some macros with a text editor so that you can bring the multiuser environment up or down with one command.

Next you consider other DG software — like compilers — that you acquired with AOS; and you create a tailored error message file for these. Finally — to make life easier for users — you create log-on messages and perhaps help messages.

This chapter explains how to do all these things, in the following major sections:

- Creating the Initial Profiles
- Initializing EXEC and Its Queues
- Editing the UP and DOWN Macros
- Other DG Software
- Making Life Easier for Users
- Your Finished System and Its File Structure
- What Next?

Creating the Initial Profiles

This section leads you through a session in which you create two classes of profiles:

The operator profile, which provides all privileges and powers needed to control the system.

User profiles, which provide only those privileges that users actually need.

Later, as your system evolves, you may want to edit individual profiles according to user needs.

The Operator Profile

You — and the person who routinely operates the system (if any) — need a profile that gives you the SUPER powers needed to control the AOS system.

The master CLI (PID 2) that runs on the system console already has all these powers. But having a profile and user directory will allow you to log on as the operator from any console (instead of using the system console), and also provide a directory for your own files. So you should create an operator profile first. Type

) DIR :UTIL ↵
) XEQ PREDITOR ↵

AOS USER PROFILE EDITOR REV n date time
COMMAND?

PREDITOR has commands to Create a profile (C), List a profile's specifications (L), and Edit an existing profile (E), among others. Each command has its own dialog. If you make a mistake when you answer a PREDITOR question, you can "back up" by pressing ^ (SHIFT and 6 keys) until you reach the bad entry; then type the desired answer and proceed.

You want to create a profile. So type

C ↵

USERNAME:

The operator profile must have a username of OP, so type

OP ↵

PASSWORD CHANGE? (Y OR NL)

The values displayed in parentheses are valid answers to the question. For a new profile, you must type

Y ↵

NEW (3-15 CHARS):

On the system console, the master CLI is always available and a username and password aren't required to use it. But on any other console, you will need to type in the username and password to log onto the system. A password can be any combination of 3 to 15 of the following characters: UPPER- or lowercase letters (treated as UPPERCASE), numbers 0 through 9, period (.), underscore (_), dollar sign ($), or question mark (?). You will be able to change the password when you log on — so, for simplicity, choose something simple like

OPR ↵

INITIAL IPC FILE [] CHANGE? (Y OR NL)

The IPC file is a file that the initial program (usually the CLI) will execute when this user logs on. It usually contains a sequence of CLI commands. Users can edit the file to set their searchlists, default access control lists, prompts, and so on; but the file is not required. The empty brackets mean that the default is null (no initial IPC file). You want one, so type

Y ↵

NEW (0-63 CHARS):

PREDITOR wants the IPC file pathname. The IPC *filename* can be the same for all users, since a version of the file can reside in each user directory, pathname :UDD:username:filename.

You, or the user, can create the file later with a text editor. The filename should be descriptive and concise; e.g., LOGON.CLI. So, for OP, you might type

:UDD:OP:LOGON.CLI ↵

PROGRAM [CLI.PR] CHANGE? (Y OR NL)

Press NEW LINE to choose the default answer to this question. (This will allow the user to come up in the CLI.) Then, answer the following questions as shown:

↵

CREATE WITHOUT BLOCK [NO]? (Y, N, OR NL) Y ↵
USE IPC [NO]? (Y, N, OR NL) Y ↵
USE CONSOLE [YES]? (Y, N, OR NL) ↵
USE BATCH [YES]? (Y, N, OR NL) ↵
MODEM [NO]? (Y, N, OR NL)

You may not want to give the operator profile the privilege to use a modem. If you do give this privilege, and an unauthorized user somehow learns the operator password, he or she can access and use the system at will from outside the installation.

If you will want to use a modem yourself, you can create another profile later, with modem privileges. For now, say No by typing

↵

USE VIRTUAL CONSOLE [YES]? (Y, N, OR NL)

You may not want to give the operator profile the networking privilege. If you do give this privilege, an unauthorized user can access and use the system at will from outside the installation.

If you will want to use a network yourself, you can create another profile later, with networking privileges. For now, say No by typing

NO ↵

ACCESS LOCAL RESOURCES FROM REMOTE MACHINES [YES]? (Y, N, OR NL) ↵
CHANGE PASSWORD [YES]? (Y, N, OR NL) ↵

UNLIMITED SONS [NO]? (Y, N, OR NL) Y ↵
CHANGE PRIORITY [NO]? (Y, N, OR NL) Y ↵
CHANGE TYPE [NO]? (Y, N, OR NL) Y ↵
CHANGE USERNAME [NO]? (Y, N, OR NL) Y ↵
ACCESS DEVICES [NO]? (Y, N, OR NL) ↵
SUPERUSER [NO]? (Y, N, OR NL) Y ↵
SUPERPROCESS [NO]? (Y, N, OR NL) Y ↵

BECOME INFOS [N] CHANGE (Y, N, OR NL)

Any process that will start an INFOS II process needs this privilege. This includes the operator process, so answer Y ↵.

MEMORY [32] CHANGE (Y OR NL)

This governs the maximum number of 2,048-byte memory pages a user's processes can use. Take the default (32 pages) by pressing ↵.

PRIORITY [2] CHANGE (Y OR NL) ↵
MAX QPRIORITY [0] CHANGE (Y OR NL) ↵
DISK QUOTA [500] CHANGE (Y OR NL)

500 512-byte blocks is not much disk space. For now, change the QUOTA to something like 15000. You can always change it again later with the PREDITOR's Edit command. Type

Y ↵
NEW (0-2147483647): 15000 ↵

User comments are simply text strings placed in the profile file; they are purely informational. They are handy for users' full names and/or dates. To enter one for the OP profile, type Y↵ and enter a useful comment; for example,

Y ↵
NEW (0-79 CHARS): OP PROFILE, 30 APR 85 ↵

COMMAND:

You've finished the Operator profile. As with any existing profile, you can list its specs by typing L↵, then the username↵. You can edit its specs one by one by typing E↵, then username↵.

The first time you run it on an LDU, PREDITOR creates a user profile directory, :UPD, and user directory directory, :UDD. For every profile you create (like the Operator profile), PREDITOR creates a profile file in :UPD and a user directory (with the amount of disk space specified) in :UDD. Each user directory and profile is named username; e.g., :UDD:OP for username OP.

Standard User Profiles

Users' needs vary. Some may need large amounts of disk space or specific privileges. After all, you bought your computer system to run programs and the process that will run these programs will need a user profile (unless you plan to run all large programs interactively from the master CLI, which has the SUPERUSER privilege and can do anything it wants).

Given these variables, we suggest general-purpose default values for all users. You can then tailor these for individual users. To set up the default profile, edit PREDITOR's internal default profile (!DEFAULT!) as in the following dialog. But, for DG Comprehensive Electronic Office (CEO®) user profiles see *Managing the CEO® System* or Chapter 7.

```
COMMAND:     E ↵
USERNAME:     !DEFAULT! ↵
PASSWORD CHANGE? (Y OR NL)     Y ↵
NEW (3-15 CHARS): GENERAL     ↵
INITIAL IPC FILE [] CHANGE? (Y OR NL)     Y ↵
NEW (0-63 CHARS):     :UDD:GENERAL:LOGON.CLI ↵
PROGRAM [CLI.PR] CHANGE? (Y OR NL)     ↵
CREATE WITHOUT BLOCK [NO]? (Y, N, OR NL)     ↵
USE IPC [NO]? (Y, N, OR NL)     ↵
USE CONSOLE [YES]? (Y, N, OR NL)     ↵
USE BATCH [YES]? (Y, N, OR NL)     ↵
MODEM [NO]? (Y, N, OR NL)     ↵
USE VIRTUAL CONSOLE [YES]? (Y, N, OR NL)     ↵
ACCESS LOCAL RESOURCES FROM REMOTE MACHINES [YES]? (Y, N, OR NL)     ↵
CHANGE PASSWORD [YES]? (Y, N, OR NL)     ↵
UNLIMITED SONS [NO]? (Y, N, OR NL)     ↵
SONS [1] CHANGE? (Y OR NL)     Y ↵
NEW (0-255):     3 ↵
CHANGE PRIORITY [NO]? (Y, N, OR NL)     ↵
CHANGE TYPE [NO]? (Y, N, OR NL)     ↵
CHANGE USERNAME [NO]? (Y, N, OR NL)     ↵
ACCESS DEVICES [NO]? (Y, N, OR NL)     ↵
SUPERUSER [NO]? (Y, N, OR NL)     ↵
SUPERPROCESS [NO]? (Y, N, OR NL)     ↵
BECOME INFOS [NO] (Y OR NL)     ↵
MEMORY [32] CHANGE (Y OR NL)     ↵
PRIORITY [2] CHANGE (Y OR NL)     ↵
MAX QPRIORITY [0] CHANGE (Y OR NL)     ↵
DISK QUOTA [500] CHANGE (Y OR NL)     Y ↵
NEW (0-2147483647):     15000 ↵
USER COMMENT [] CHANGE? (Y OR NL)?     Y ↵
NEW (0-79 CHARS):     General User ↵
COMMAND:
```

This tailors PREDITOR's !DEFAULT! profile for general use. Now, when you create each profile, the new defaults you gave will appear; this allows you to default more questions and saves time. The original *[default]* values will return when PREDITOR terminates — so you should do all the profiles you need during this session.

Creating the Profiles

For each profile, the only values you *must* enter (and cannot default) are username and password.

Each username must be unique among usernames. As a username, you might use a person's first name and initial (if needed) (If a user doesn't like his/her username, you can rename the profile via PREDITOR later). For the password, use the username and tell the user to change it when he or she logs on (described later).

The following dialog shows how you might set up a profile for someone named JACK. It also explains some of the issues involved.

COMMAND: C ↵
USERNAME: JACK ↵ (type username)
PASSWORD CHANGE? (Y OR NL) Y ↵
NEW (3-15 CHARS): JACK ↵ (type password)

INITIAL IPC FILE [:UDD:GENERAL:LOGON.CLI] CHANGE? (Y OR NL)

For IPC, change the default to UDD:username:LOGON.CLI or the desired file pathname. For example,

Y ↵
NEW (0-63 CHARS): :UDD:JACK:LOGON.CLI ↵

PROGRAM [CLI.PR] CHANGE? (Y OR NL)

For *PROGRAM,* default to CLI.PR unless you want this user to come up in BASIC or some other program — in which case, type Y↵ and the full pathname, with .PR suffix, of the program you want. For BASIC, there is often a BASIC directory off the root or :UTIL. If this will be true, you'd answer Y↵, then :BASIC.PR↵ or :UTIL:BASIC.PR↵ to have the user come up in BASIC.

The CLI is a good general-purpose choice because it allows users to access text editors and write programs in *all* DG languages; it also allows users to execute other programs like BASIC. Unless you know that you want this user to come up in a program other than the CLI, select the default by pressing

↵

CREATE WITHOUT BLOCK [NO]? (Y, N, OR NL)

CREATE WITHOUT BLOCK means that the user can have at least two processes running concurrently. By default, the creating (father) process is blocked when it executes the son; this means that the father is eligible to be swapped, which may speed up the system. But if the user will be using DG's SWAT™ debugger (for FORTRAN 77, PL/I, or COBOL programs), he or she must have the CREATE WITHOUT BLOCK privilege. So, for such users, you must answer Y↵. Otherwise, take the default, which is NO:

↵

USE IPC [NO]? (Y, N, OR NL) ↵

IPC means InterProcess Communications calls, available in assembly language and some higher-level languages. IPC privileges are needed wherever two or more active processes must communicate. For IPC usage to work, a user must also have CREATE WITHOUT BLOCK, because at least two of his/her processes must remain active if they are to use IPC. For most people, take the default:

)

USE CONSOLE [YES]? (Y, N, OR NL))
USE BATCH [YES]? (Y, N, OR NL))
MODEM [NO]? (Y, N, OR NL)

If you want this user to be able to log on via a modem, type Y). Generally, SUPERUSERs should not be able to use a modem, because the two privileges allow the user to explore the entire system from his own home or wherever a remote console is placed. In most cases, press

)

USE VIRTUAL CONSOLE [YES]? (Y, N, OR NL)

This question, and the next, are meaningful only if your system will run DG's XODIAC™ networking software. The default answer allows this user to log onto your system from a remote host system. Generally, unless you don't want the user to do this, press

)

ACCESS LOCAL RESOURCES FROM REMOTE MACHINES [YES]? (Y, N, OR NL))

The default answer allows a remote user to access files and devices like tapes and printers on your system. This is different from being able to log on, as covered in the previous question. Again, unless you don't want the user to do this from a remote system, press

)

CHANGE PASSWORD [YES]? (Y, N, OR NL)

In general, users should be able to change their own passwords, per the default. But if you are setting up a GUEST profile, to allow guests to use your system, the password must be public; and you should answer N) to prevent a guest from changing the password and barring other guests from the system. Generally, press

)

UNLIMITED SONS [NO]? (Y, N, OR NL)

A user who can create unlimited son processes has the potential for hobbling the system. Each process requires some CPU time, and disk I/O to the SWAP.SWAP file. So far as possible, it's a good idea to minimize the number of processes. So, in most cases, press

)

SONS [3] CHANGE? (Y OR NL)

BASIC and clerical data entry users can get along with fewer sons: 0 or 1. CLI users who want to execute a nonCLI son process from within a son (instead of going back to the CLI to do it), need more sons: at least five. For these people, you might want to say Y). PREDITOR will say NEW(0-255) and you will type the new number and).

For other users, three sons is a good general-purpose number. This will allow a user to develop programs and use the SWAT debugger; it's a minimum for serious application programmers who will use FORTRAN 77, PL/I, or COBOL. So, for this group, take the default by pressing

)

BECOME INFOS [NO] CHANGE (Y, N, OR NL)

Generally, user processes don't need this privilege. So, unless you know that this user will need to start INFOS, press).

MEMORY [32] CHANGE (Y OR NL)

Generally, each user should have the maximum of 32 memory pages, so press ↓.

CHANGE PRIORITY [NO]? (Y, N, OR NL)

Processes compete for CPU time, and processes of the same type with higher priority (closer to 0) get preference. But it's simpler and often better to have processes of the same type all have the same priority. So, unless you know that a process *must* be able to change its priority, press

↓

CHANGE TYPE [NO]? (Y, N, OR NL)

Processes can run as one of three types: resident (always in main memory), pre-emptible (generally in main memory, but swappable if blocked), and swappable. Resident and pre-emptible always have priority over swappable. Swappable is the most common type and is the default for user processes. Resident is quite rare — used only for the AOS peripheral manager and the system itself. If a process can change type, it can become resident, and perhaps hobble the system. So, unless you know that a process must be able to change its type, answer NO by pressing

↓

CHANGE USERNAME [NO]? (Y, N, OR NL)

A process that can change its username can assume the name of OP or another privileged user — giving it access to the entire system. Again, unless you know that the process must be able to change its username, say NO by pressing

↓

ACCESS DEVICES [NO]? (Y, N, OR NL)

This privilege allows a process to bypass operating system safeguards, and access devices directly in machine language. You should never give it unless the user is a systems programmer who needs it to write or debug device drivers. So, in nearly all cases, press

↓

SUPERUSER [NO]? (Y, N, OR NL)

This privilege allows a user process to bypass all file access controls and *execute, read, modify, or delete any file on the system.* SUPERUSERs can run PREDITOR to change their own profiles; and they can find other users' usernames and passwords in their profile files. The master CLI needs SUPERUSER to control the system; but most users (other than OP) do not need it, and they shouldn't have it. In most cases, say NO by pressing

↓

SUPERPROCESS [NO]? (Y, N, OR NL)

This privilege allows a user process to terminate any process — including the master CLI, which would bring down the entire system. Unless you know that a process needs SUPERPROCESS, say NO by pressing

↓

PRIORITY [2] CHANGE (Y OR NL)

This user process will be type swappable. Swappable processes can have priority 1, 2, or 3. Generally, you will want to maintain process equality, so press

↓

MAX QPRIORITY [0] CHANGE (Y OR NL)

When the multiuser environment is up and running, users will use Q-series commands to print files and submit batch jobs. Priority 0 is the default and highest priority. If you give the default to all users, they will receive equal treatment on their Q-series jobs. Generally — unless you want to favor or discourage a user's Q-series requests — press

```
    ↵
```

DISK QUOTA [15000] CHANGE (Y OR NL)

This sets the limit on the size of the user directory that PREDITOR will create and the system maintain for this user process.

The default — 15000 512-byte blocks — is a good general-purpose amount of disk space. If this user process is for guests or other casual users, then you might want to specify less space (perhaps the original default, 500). If this user process will be used by many people (perhaps data entry clerks or students), you might want to specify a larger figure (e.g., 100000). If this user process will deal with a large database and its directory will contain the database(s), you might want to allot an entire single- or multiple-disk LDU to it. A model 6061 disk contains about 370000 blocks (depending on whether a system is installed on it); a model 6122 disk contains about 540000 blocks. If you want to change the space quota, type Y↵. PREDITOR will then say *NEW(0-2147483647)*: and you will type the new quota.

To accept the default, press

```
    ↵
```

USER COMMENT [GENERAL USER] CHANGE? (Y OR NL)?

As described earlier, you can use this for text comment about the user: full name, date, etc. Type Y↵, then the desired command. For example,

```
                        Y ↵
NEW (0-79 CHARS):       JACK   ARMSTRONG   30   APR   85 ↵
```

COMMAND:

You have finished this profile. PREDITOR has written it into its profile directory :UPD, as a file with the filename of USERNAME; e.g., JACK for a username of JACK. The profile is ready for use.

To do another profile, type C↵ and run through this section again. (Remember that a user must have a profile to log onto the system. Many people can have a common profile — usable through a common username and password — but, aside from using the system console, a person who doesn't know a valid username and password cannot use the system.)

When you have finished all the profiles you want, PREDITOR will be asking for a COMMAND. Leave PREDITOR and return to the CLI by typing

```
    BYE ↵
)
```

Proceed to the next section.

Initializing EXEC and Its Queues

With the user profiles done, you can initialize the EXEC process. This involves executing EXEC and creating and opening EXEC queues; generally you need to do it only once.

Via your primary tool, the master CLI, set your search list and turn on SUPERUSER:

```
) SEARCH   :UTIL ↵
) SUPERU   ON ↵
*)
```

Now start up EXEC:

```
*) PROCESS/DIR=@/DEF/NAME=EXEC   EXEC ⏎
PID 3
FROM PID 3 : (EXEC) EXEC REV n READY
FROM PID 3 : (EXEC) time
*)
```

The PROCESS command creates a new process — just as XEQ does — but it is more versatile. The switches make the EXEC process' home directory :PER (@ means :PER), give it default privileges (/DEF), and make its process name EXEC.

The Pid messages indicate that EXEC is running as a process with Process ID 3 (PID 3); and give its revision and the time.

Now create the batch and spool queues via the following commands to EXEC.

```
*) CONTROL   @EXEC   CREATE   PRINT   LPT ⏎
FROM PID 3 : (EXEC) ...
*) CONTROL   @EXEC   OPEN   BATCH_INPUT ⏎
FROM PID 3 : (EXEC) ...
*) CONTROL   @EXEC   OPEN   BATCH_OUTPUT ⏎
FROM PID 3 : (EXEC) ...
*) CONTROL   @EXEC   OPEN   BATCH_LIST ⏎
FROM PID 3 : (EXEC) ...
*) CONTROL   @EXEC   OPEN   LPT ⏎
FROM PID 3 : (EXEC) ...
*)
```

The CONTROL @EXEC directs the command through the CLI to EXEC. The commands created EXEC's permanent spool queues in a file named :QUEUE:Q. As long as this file exists, you'll never need to repeat these commands.

Now, go to the line printer and make sure power is on, paper is aligned, and that it is ON LINE. The printer must be on line for EXEC to start it.

Associate the batch output and batch list file with the line printer queue, and start the queue, by typing

```
*) CONTROL   @EXEC   CONTINUE   1 ⏎
FROM PID 3 : (EXEC) STREAM 1 CONTINUING
FROM PID 3 : (EXEC) STREAM 1 IDLE

*) CONTROL   @EXEC   START   BATCH_OUTPUT   @LPB [UPPER] ⏎
FROM PID 3 : (EXEC) @LPB CO-OPERATIVE INITIATED
FROM PID 3 : (EXEC) @LPB PAUSED

*) CONTROL   @EXEC   START   BATCH_LIST   @LPB [UPPER] ⏎
FROM PID 3 : (EXEC) ...

*) CONTROL   @EXEC   START   LPT   @LPB ⏎
FROM PID 3 : (EXEC) ...

*) CONTROL   @EXEC   CONTINUE   @LPB ⏎
FROM PID 3 : (EXEC) @LPB CONTINUING
```

For an uppercase-only type LPA, LPB, or LPC printer, include the argument UPPER in the EXEC START command; this tells EXEC to change lowercase characters to uppercase for printing. For a type LPA (LPC) printer, use the device name LPA (LPC) instead of LPB.

For a type LPD printer, use device name LPD instead of LPB. And for an LPD printer that is a laser document printer, use the following variation of the EXEC START command:

```
*) CONTROL @EXEC START/NL LPT @LPD ⏎
```

This tells EXEC to correct each NEW LINE (⏎) character it prints as needed for the laser printer.

You have continued the batch streams, and started the batch queues and printer queue on the printer. Users can now issue Q-series commands. There are four batch streams, and we have activated (with CONTINUE) only number 1, but this is enough to start. These commands are part of a CLI macro named UP.CLI, so they will not need to be typed individually.

Letter-Quality Printer Queues

If your system has one or more letter-quality printers (connected on console lines), create and open a letter-quality printer queue for each. For example, if you have two letter-quality printers:

```
*) CONTROL @EXEC CREATE PRINT LQP ↵
*) CONTROL @EXEC CREATE PRINT LQP1 ↵
*) CONTROL @EXEC OPEN LQP ↵
*) CONTROL @EXEC OPEN LQP1 ↵
```

*

This sequence of commands readies the batch and printer queues for use; users can now use Q-series commands. There are four batch streams, and we have activated (with CONTINUE) only number 1, but this is enough to start. These commands are part of a CLI macro named UP.CLI, so they will not need to be typed individually.

Second Line Printer and Plotter Queues

If you have a second line printer, initialize a queue for it.

```
*) CONTROL   @EXEC   CREATE   PRINT   LPT1 ↵
*) CONTROL   @EXEC   OPEN   LPT1 ↵
*)
```

And if you have a digital plotter, do its queue.

```
*) CONTROL   @EXEC   CREATE   PLOT   PLT ↵
*) CONTROL   @EXEC   OPEN   PLT ↵
*)
```

Enabling a User Console

Now to enable a user console. The user console names were determined at AOSGEN and — if the consoles haven't been labeled, now is a good time to label them. Using white tape or tape labels, label each with its name: @CON2, @CON3, etc.

Choose a CRT (for example, @CON2), turn it on, and place it on line. Then enable it via EXEC:

*) CONTROL   @EXEC   ENABLE   @CON2 ↵
FROM PID 3 : (EXEC) ENABLED CONSOLE, @CON2

@CON2 is ready for user log on. If you want to try enabling *all* consoles, type CONTROL @EXEC ENABLE/ALL ↵. This tells EXEC to try and enable all consoles specified to AOSGEN (files of type CON in directory :PER).

If EXEC does not report an error, skip the next section.

If EXEC Fails to Enable a Console

If you get an error message from the EXEC ENABLE command, issue it again. If the error persists, type DIR @↵, then F/S↵ to list the entries in directory @ (:PER, the peripherals directory). All the device and console names you specified during AOSGEN should be in this directory. The console names here should match the names you pasted on the consoles. If not, you may have made a mistake during AOSGEN, or the console line connections may be wrong.

Try readying another console whose name appears in directory @ (:PER); and issue the CONTROL @EXEC ENABLE command with the other console's name.

Logging on as a User

Having enabled a console, walk over to it. The screen (or paper, on a hard-copy console) should say

AOS REV n / TYPE NEW-LINE TO START LOGGING ON /

Now, log on with the OP username and password you created for the operator with PREDITOR. Do it as follows:

↓

AOS EXEC REV n date time
USERNAME: OP ↓
PASSWORD: (type the password; e.g., OPR ↓. It doesn't echo.) *AOS CLI REV n date time*
)

You have now logged on as a user (user OP) and your user process is running a CLI process for you. This is your own CLI, independent of the CLI on the system console.

Now you know that EXEC's logon function works, and that the hardware and software configuration of your system are probably correct.

Try some QPRINT commands:

) QPRINT :UTIL:ERMES.SR ↓
QUEUED, SEQ = n
) QPRIN :UTIL:AOS.PANICS.SR ↓
QUEUED, SEQ = n
)

The line printer should now print the text of these files, each preceded by a header sheet that gives your username, file pathname, and date, among other things. You needn't read these files now — they served only to test the printer queue. Take the printed copy to the system console for later use.

Try a batch command:

) QBATCH Write Hello ↓
QUEUED, SEQ = n
)

The batch output file is the line printer, so the text string Hello should appear there, preceded by a printed header and logon information. If so, EXEC's batch function is okay.

To get a sense of the CLI's help facility, type

) HELP ↓
.
. (CLI displays HELP topics)
.
)

The entire help facility is available to any user from any CLI. Use it whenever you have doubts or questions on a topic or command.

Things look good for the multiuser environment. You can terminate the user process by typing

) BYE ↓

AOS CLI TERMINATING
PROCESS CONNECT TIME ...

AOS REV n / TYPE NEW LINE TO BEGIN LOGGING ON /

and return to the system console.

Changing a Password

By default, any user can change his or her password at log on time. The user types username↵ as usual, then types the password but hits the END PAGE key (or the ERASE page key, or types CTRL-L) instead of ↵.

EXEC will then ask for the new password. When the user types it and ↵, EXEC logs the user on. From that point on, the new password will be in effect. Note that usernames are public information, but passwords — including the OP password — should be private.

Bringing down EXEC

Back at the system console, pause the device queues, and bring down EXEC.

) CONTROL @EXEC PAUSE 1 ↵
FROM PID 3 : (EXEC) STREAM 1 WILL PAUSE...
FROM PID 3 : (EXEC) STREAM 1 PAUSED

) CONT @EXEC PAUSE @LPB ↵
FROM PID 3 : (EXEC) @LPB PAUSED

) TERM OP:EXEC ↵
*)

The CLI command TERMINATE (TERM) brings down a process and all its sons. Here, it terminated EXEC.

Editing the UP and DOWN Macros

As you saw, there are a lot of commands involved in bringing EXEC up and down. To make this easier, DG supplied CLI macros named UP.CLI and DOWN.CLI in directory :UTIL. Because systems vary, these macros are not directly executable; you must edit them for *your* system before you can use them. When you've done this, you'll need only to type UP↵ to bring up EXEC and the multiuser environment; and DOWN↵ to bring them down.

To edit the macros, you'll need to know how to use a text editor — specifically, the text editor named SED.

Using the SED Text Editor

SED is a good text editor, with informative error messages and its own Help facility. It has many commands and features. But to do the editing you need now, you'll require only a few commands:

HELP	Gives Help on SED commands or features.
APPEND	Adds text to the end of the file.
BREAK/ESC or ESC key	Terminates an APPEND or INSERT.
INSERT	Inserts lines of text before the current line.
MODIFY	Edits lines of text one by one.
→ ← ↑ ↓	Moves cursor. These cursor control keys are on the keypad to the right of the main keypad. They move the cursor to the right, left, a line up, or a line down.
CTRL-E	Begins a character insert or ends a character insert, on a line.
DEL key	Deletes the previous character.
LIST	Displays a range of lines.
FIND	Finds a text string.
DELETE	Deletes one or more lines of text.
BYE	Leaves SED and returns to the CLI.

The SED prompt is an asterisk (*). As with CLI commands, you can abbreviate SED commands to their shortest unique parts; e.g., MOD for MODIFY.

SED is a line-oriented editor, dealing with text a line at a time. It is also screen-oriented, depending heavily on cursor control keys. Thus you should use it on a CRT console if you can. If the system console is a printing console, bring up EXEC again with the PROCESS command, do a CONTROL @EXEC ENABLE @CONn (where @CONn is a CRT), log onto the CRT as OP, turn SUPERUSER ON↵, and create/edit the macros. When you've finished editing the macros, go back to the system console and TERM OP:EXEC↵; then test the macros from the system console.

The first steps are to get into directory :UTIL, where all the macros are, and turn SUPERUSER on so you can create new files. Use the DIRECTORY and SUPERUSER commands for this.

```
) DIR   :UTIL ↵
) SUPERU   ON ↵
```

Note that the following sections tell you to create a number of macros before you edit UP.CLI. If any of these macros exist, don't recreate them; simply try them to see if they work. You can see if a macro exists by typing F/AS name↵, where name is the macro filename. If the F/AS command returns the name, the macro exists.

Editing Macro CX.CLI

Now create a macro called CX.CLI. This will allow you to issue EXEC commands as simply "CX" command, instead of the tedious sequence "CONTROL @EXEC" command. Type

*) XEQ   SED   CX.CLI ↵

SED REV n Input file - :UTIL:CX.CLI
Do you want CX.CLI to be created? YES ↵

* APPEND ↵	(Append text to the new file.)
1 CONTOL @EXEC %-% ↵	(Append CONTOL @EXEC %-%.)
2 ESC	(Press BREAK/ESC or ESC key)
* MOD 1 ↵	(Modify line 1.)
1 *CONTOL @EXEC %-%*	(SED displays line 1.)
Press → 4 times, CTRL-E, type R, then, ↵, producing	(Use cursor control characters to correct typo.)

CONTROL   @EXEC   %-%
* BYE ↵
Output File - :UTIL:CX.CLI
*)

This little macro will save a lot of time in the future. (The line numbers — 1 and 2 above — are displayed by SED for your editing convenience. They are not part of the file.)

Editing Macro SED.CLI

Another useful macro which may not exist is SED.CLI. It allows you to skip the "XEQ" when you want to use SED. To create it, type

```
*) XEQ   SED   SED.CLI ↵
SED Rev n Input File :UTIL:SED.CLI
Do you want SED.CLI to be created?      Y ↵
* APP ↵                                 (Append text.)
1     XRQSED%/%   %-% ↵                 Append XRQSED%/% %-%.)
2     ESC                               (Press BREAK/ESC or ESC key.)
* MOD   1 ↵                             (Modify line 1.)
1          XRQSED%/% %-%                (SED displays line 1.)
```

Press →, type E to correct the R, press →, press CTRL-E, press the the space bar, then press ↵ — changing the bad line to XEQ SED%/% %-%

(Use control characters and space bar to correct the bad line.)

```
* BYE ↵
Output File - :UTIL:SED.CLI
*)
```

Now you can run SED by simply typing SED filename↵.

Editing Macro ?.CLI

A very useful macro which may not exist is ?.CLI, which tells you about all processes running on your system. To create a simple, yet workable version of ?.CLI, type

```
*) X   SED   ?.CLI ↵
Input file — :UTIL:?.CLI
Do you want ?.CLI to be created?      Y ↵
* AP ↵
1     WHO/2=IGNORE   <,1,2,3,4,5,6,7,8,9><0,1,2,3,4,5,6,7,8,9> ↵
2     WRITE   You are the following process: ↵
3     WHO ↵
4     ESC            (Press BREAK/ESC or ESC key.)
* BYE ↵
Output file — :UTIL:?.CLI
*)
```

This macro displays every process on the system with a number 1 through 99. You can execute it by typing ?↵.

To make macros SED.CLI and ?.CLI available to *users*, type the following CLI commands:

```
) ACL   SED.CLI   OP,OWARE   +,RE ↵
) ACL   ?.CLI   OP,OWARE   +,RE ↵
```

For the following few macros, we show only the text. Try the SED commands for yourself.

Macro FF.CLI

In many sites, the fold on which line printer paper falls is important. If people use the line printer controls to output the last sheets of their printing jobs, and they press TOP OF FORM an odd number of times, then the paper fold will be reversed — which can be a nuisance. Even worse, people may forget to put the printer back on line, preventing requests from being processed until someone puts it back on line.

Instead of having people operate the printer controls, you can create a form feed macro, perhaps named FF.CLI. People can then type FF ↵ after their QPRINT commands, and the printer will automatically output enough paper to allow them to tear off their jobs. You can cover the printer controls with a label that says "Use FF macro."

There are two files involved. Get into the directory you want to hold the files — say :UTIL — and type

*) CREATE/I RECYCLE_ME ↵	(Or whatever header you want.)
*)) ↵	(Insert a NEW LINE.)
*)) CTRL-L	(Insert a form feed.)
*))) ↵	(End text insert mode.)

Now create the form feed macro — let's say FF.CLI. The text is

```
QPRINT/DESTINATION=RECYCLE_ME   RECYCLE_ME
```

After creating the two files, make them accessible to users by typing

```
*) ACL   RECYCLE_ME   OP,OWARE   +,RE ↵
*) ACL   FF.CLI   OP,OWARE   +,RE ↵
```

You can use whatever header text you want on the throw-away sheets, instead of RECYCLE_ME. If you have a second line printer, you can create a second form feed macro with the name FF1.CLI. FF1.CLI would be the same as FF.CLI, but with the queue name LPT1.

Macros BATCH.CLI and CHEK.CLI

By default, when users submit batch jobs (usually with the QBATCH command), the output and list files are the first line printer queue, LPT. To check the results of the batch jobs, users must walk to the line printer. This may discourage them from using batch. The following macros allow any user to do a whole batch job without leaving his or her console.

Macro BATCH.CLI queues a batch job with output and list files in a user's initial directory; macro CHEK.CLI types and deletes these files. (It's named CHEK.CLI to distinguish it from the CLI command CHECKTERMS.)

Macro BATCH.CLI is self-documenting. It explains itself if someone types its name without an argument (e.g., BATCH ↵). This is a good idea for your own user-oriented macros: if they explain themselves, then you don't have to explain them.

The text of BATCH.CLI is

```
[!EQUAL, %1% , ]
   WRITE  CLI macro %0% queues a batch job, with multiple
   WRITE  arguments. It writes the batch output file and batch
   WRITE  list file to your initial working directory -- instead
   WRITE  of the line printer queue -- so that you needn't go to
   WRITE  the printer each time you use batch. Do not use it to
   WRITE  stack multiple batch jobs -- wait for one job to complete
   WRITE  before using it to queue the next job. The format for
   WRITE  using this macro is ,, BATCH , normal-command-line , NEW-LINE
   WRITE  ,, BATCH , normal-command-line , NEW-LINE ,,, For example,
   WRITE  ,, BATCH , XEQ , MASM , PROG1 , PROG2 NEW-LINE
[!ELSE]
   DELETE/2=IGNORE   :UDD:[!USERNAME]:(LAST_BATCH.<OUT,LIST>)
   CREATE   :UDD:[!USERNAME]:(LAST_BATCH.<OUT,LIST>)
   QBATCH/NOTIFY/QOUTPUT=:UDD:[!USERNAME]:LAST_BATCH.OUT&
/QLIST=:UDD:[!USERNAME]:LAST_BATCH.LIST %-%
   WRITE  When this batch job is done your console will
   WRITE  show 'STREAM COMPLETED' then beep. Type CHEK NEW-LINE.
   WRITE  Macro CHEK types the batch output file -- allowing
   WRITE  you to check for errors. Then it allows you to
   WRITE  delete or save the batch output and list files.
[!END]
```

The text of CHEK.CLI is

```
[!EQ, comment,] This macro prints batch output file after a
batch job created by macro BATCH.CLI completes -- [!end]

  TYPE :UDD:[!USERNAME]:LAST_BATCH.OUT
  WRITE To delete batch output and empty batch list files press NEW-LINE.
  STRING [!READ To save them type S NEW-LINE. ]
  [!EQUAL,[!STRING],S]
    WRITE Saving output and list files
    WRITE :UDD:[!USERNAME]:LAST_BATCH.<OUT LIST>
  [!ELSE]
    DELETE/V :UDD:[!USERNAME]:LAST_BATCH.OUT
    [!EQUAL [!SIZE :UDD:[!USERNAME]:LAST_BATCH.LIST] ,0]
      DELETE/V :UDD:[!USERNAME]:LAST_BATCH.LIST
    [!ELSE]
      WRITE List file - :UDD:[!USERNAME]:LAST_BATCH.LIST
      WRITE is NOT empty. Saving.
    [!END]
  [!END]
```

Make the macros accessible to users by typing

```
) ACL  BATCH.CLI  OP,OWARE  +,RE ↲
) ACL  CHEK.CLI  OP,OWARE  +,RE ↲
```

Now, to post a batch job, all a user needs to do is type BATCH command-line ↲; for example

```
) BATCH  X  MASM  MYFILE ↲
```

The BATCH macro will advise the user of what to do next; and the prompt will return to his console. When the job is done, the user types

```
) CHEK ↲
```

which displays the batch output file — showing all errors — with an option to delete both files. These macros offer a fast, simple and effective way for users to post batch jobs.

Macro REMEMBER.CLI

As people work on your system, they may want to post reminders to themselves — for example, about meetings or deadlines. You can make it easy for anyone to post one or more such personal reminders by creating macro REMEMBER.CLI (naturally, you can give it any name you want).

The text of REMEMBER.CLI is

```
[!eq, %2%, ]
  WRITE This macro -- %0% -- reminds you of a future engagement.
  WRITE It enqueues a batch job to run at the time you specify.
  WRITE The batch job consists of SENDing the specified message
  WRITE to your console.
  WRITE
  WRITE To execute the macro use the form
  WRITE ,,, %0% ,,,, date:time ,,,, message ,,,, NEW-LINE
  WRITE
  WRITE "date:time" can be a specific date and/or time -- for example
  WRITE "23-AUG-85:20" means "August 23 85 8pm". Or "date/time" can be
  WRITE relative -- for example "+1" means "an hour from now". Try the
  WRITE HELP *AFTER ,,, topic for more information on "date/time".
[!ELSE]
  STRING @[!CONSOLE]
  QBATCH%/%/after=%1%/qlist=@null/qoutput=@null &
  SEND [!STRING] [!asc 215] [!asc 207] [!asc 216] REMEMBER: [!asc 217] %2-%
[!END].
```

Make the REMEMBER macro available to users by typing

) ACL REMEMBER.CLI OP,OWARE +,RE ↵

Now, for example, if user Sally wants to be reminded of a meeting on August 23, 1985, at 4:00 pm (16:00 on a 24-hour clock), she would type

) REMEMBER 23-AUG-85:16 Meeting ↵

Then, when the system calendar and clock showed the specified time, Sally's REMEMBER batch job would run; her console would beep and display REMEMBER blinking with the message text. (A REMEMBER batch job would be delayed if all batch streams were busy throughout the delay interval.)

Macros ON.CLI and OFF.CLI

You — and other people who run the system — will often need to turn SUPERUSER ON and OFF, and it's a nuisance to have to type the whole string to do it. Here is the text of macros ON.CLI and OFF.CLI — which turn SUPERUSER or SUPERPROCESS ON and OFF. We don't show the SED commands; try them for yourself.

Text of ON.CLI:

```
[!EQ,%0/%,/P]
SUPERPROCESS ON
[!ELSE]
SUPERUSER ON
[!END]
```

Text of OFF.CLI:

```
[!EQ,%0/%,/P]
SUPERPROCESS OFF
[!ELSE]
SUPERUSER OFF
[!END]
```

With these macros in :UTIL, you can turn SUPERUSER ON by typing ON↵; and you can turn SUPERPROCESS ON by typing ON/P↵. To turn them off, type OFF↵ or OFF/P↵.

Macro BROADCAST.CLI

In the multiuser environment, the person at the system console will often want to SEND messages to all users. The SEND @CON- message↵ command can be used for this, but it can produce a lot of messy (but harmless) error messages. The text of a macro called BROADCAST.CLI follows; it sends a message with time and (beep) to each user terminal, *without* error messages.

Text of Macro BROADCAST.CLI

```
SEND/1=IGNORE/2=IGNORE (@CON-\@CONSOLE)&
  [!ASCII 207][!USERNAME] at [!TIME] [!ASCII 276][!ASCII 240] %1-%
```

The operator can use this macro to send messages to other users by typing

) BROADCAST message ↵

If you want *users* to be able to use BROADCAST, type the following CLI command:

) ACL BROADCAST.CLI OP,OWARE +,RE ↵

Editing Macro UP.CLI

After writing some macros with SED, you're ready to tackle UP.CLI. To start, type

```
*) SED   UP.CLI ↵
SED Rev n - Input file :UTIL:UP.CLI
*
```

Type LIST ALL↵ to see the entire prototype macro. As you can see, there are instructions at the end on how to make it executable.

As time passes, and your system grows and changes, you'll be editing this macro to reflect the changes. In this session, we'll simply give some general-purpose answers that will make UP.CLI executable — without explaining the other lines in the macro.

The first line to edit is line 1. Line 1 makes the macro nonexecutable — it says, "if 1 equals 2, execute all lines up to the next !ELSE or !END." So, using cursor controls, position on the line and make the two numbers equal. For example,

```
[!EQUAL,1,1] ↵
```

Now find the PROCESS command that starts EXEC:

```
* FIND   "PROCESS/" ↵
n      PROCESS/DEFAULT/DIRECTORY=...
```

Before the line containing the PROCESS command that starts EXEC, insert the following lines to run the queue compacter program, QCMP:

```
WRITE   Running QCMP now...
XEQ   QCMP/YES
WRITE   QCMP is done.
```

The QCMP program helps keep directory QUEUE neat by deleting unused files in it. QCMP must run before EXEC does. You have just made sure it will do so.

Enabling User Consoles

The UP.CLI macro contains an EXEC ENABLE/ALL command, so you do not need to insert ENABLE commands.

But, after EXEC enables a console, it "owns" that console. Some other DG programs need to "own" consoles to run on them. Among these are data entry programs (Idea, TPMS, DATAPREP®) and IBM emulator programs (RCX70, DG/SNA). If you plan to run one or more of these other programs, decide which consoles you want them to use. Then, in the UP macro, execute the other programs and have them take charge of the consoles they need *before* the CONTROL @EXEC ENABLE/ALL command. The other program(s) will get use of the consoles, and there will simply be a *DEVICE ALREADY IN USE* error message when EXEC tries to enable them.

Alternatively, you can replace the ENABLE/ALL command with specific enable lines of the form

```
CONTROL   @EXEC ENABLE   @CON(n,n,n ... )
PAUSE   10
```

For example,

```
CONTROL   @EXEC ENABLE   @CON(2,3,4,5,6,7,8,9,10,11,12)
PAUSE   10
```

This would allow you to specifically enable consoles and avoid the EXEC *DEVICE ALREADY*... error messages. You can deal with this issue later, when you know more about your applications software. But it's important to know how to handle the console enabling issue. (The parentheses iterate the CONTROL @EXEC ENABLE @CON command on each number within them. Parentheses work the same way with other commands to the CLI.)

(If you're wondering why we didn't simply say "CX" instead of CONTROL @EXEC, you should know that it's good practice to use full commands within a macro. This helps keeps macros simple, versatile, and independent of one another.)

After making sure that the consoles you want will be enabled, proceed.

Starting the Second Line Printer and Plotter Queues

The UP macro already contains commands to start the first line printer queue and continue the printer. If your printer is an upper- and lowercase type LPA, LPB, or LPC printer, you can leave these commands alone. But if your primary printer is an uppercase-only LPA, LPB, or LPC, change the CONTROL @EXEC START... command to CONTROL @EXEC START ... UPPER. For a type LPD (LPA, LPC) printer, change the name @LPB to @LPD (@LPA, @LPC) in both the CONTROL @EXEC START and CONTROL @EXEC CONTINUE commands. For a laser document LPD-type printer, change CONTROL @EXEC START... to CONTROL @EXEC START/NL....

The EXEC START and CONTINUE commands for batch stream 1 and the first printer are just before the command POP in the prototype UP macro.

If you have a second line printer and/or plotter, start and continue its queue *after* the POP command in the macro. Depending on the printer type (LPB or LPD, uppercase only or laser document), insert the following commands after POP.

CONTROL @EXEC START *[/NL]* LPT1 @LPx *[UPPER]*

(Insert the */NL* for a laser printer, type LPD. Insert the */UPPER* for an uppercase only printer. x is B1 for the second LPB printer, D for the first LPD printer, or D1 for the second LPD printer.)

CONTROL @EXEC CONTINUE @LPx

For a digital plotter, insert the following commands:

```
CONTROL @EXEC START PLT @PLA
CONTROL @EXEC CONTINUE @PLA
```

This will allow users to post printing requests to the second printer via QPRINT/QUEUE=LPT1...commands; and/or post plotting requests via QPLOT commands.

With two or more line printers, label each printer (perhaps using a sticky-backed mag tape label) with its queue and device name; for example "Device LPB, queue LPT" for the first printer. Having the names clearly visible will make operations easier later on.

Starting Letter-Quality Printer Queues

If you created one or more letter-quality printer queues earlier, insert commands to start and continue them. For example, if you have two letter-quality printers, connected to console lines 13 and 14, and you created queues named LQP and LQP1 for them, you'd insert the following commands:

```
CONTROL @EXEC START LQP @CON15
CONTROL @EXEC START LQP1 @CON16
CONTROL @EXEC CONTINUE @CON15
CONTROL @EXEC CONTINUE @CON16
```

This will allow CLI users to post printing requests to the first letter-quality printer using QPRINT/QUEUE=LQP ... commands or to the second printer using QPRINT/QUEUE=LQP1... commands. CEO users can access the printers by CEO-defined names, from within CEO. (For CEO users, you will also want to enable binary mode — EXEC command BINARY — before continuing the letter-quality printers in the UP macro. The CEO.PRINTER macro does this automatically. For more information on CEO, see *Managing the CEO® System.*

As with line printers, you may want to label your letter-quality printers — with queuename, device name, and perhaps CEO name.

Finishing Up

Having made the macro executable, enabled user consoles, and (perhaps) started the second printer and plotter queues, you're done with the UP macro (for the time being).

Leave SED and save the old file as backup via the following commands:

```
* BYE ↵
Do you want to save the original file as a backup file?      YES ↵
Output file - :UTIL:UP.CLI
Backup file - :UTIL:UP.CLI.BU

*)
```

Copy the macro to the root directory with the MOVE command:

```
*) MOVE/V/R  :  UP.CLI ↵
UP.CLI
*)
```

The command lines in UP.CLI are all CLI commands and directives called *pseudomacros* (pseudomacros start with "[!"). You can see that UP.CLI's PROCESS and EXEC commands are similar to the CLI commands you typed earlier.

Later on, you might want UP.CLI to CONTINUE batch streams other than 1, initialize other LDUs (if any), create data management processes, use the SYSID command to create a custom banner for display on user consoles, etc. And, you might want it to start the system log, via macros described in Chapter 9.

Editing DOWN.CLI

Bringing the multiuser environment down is simply a matter of bringing down EXEC. This will also bring down interactive user processes — so whoever uses the DOWN macro will need to make sure that all users are alerted to the impending shutdown so they won't lose work.

Type

```
*) SED  DOWN.CLI ↵
```

As with UP.CLI, the two numbers in line 1 must be the same. For example you could edit line 1 to be

```
[!EQUAL,1,1]
```

After editing line 1, you might want to insert some text about pausing queues and warning users. To do so, type

```
* FI  "TERM" ↵

n     TERMINATE OP:EXEC
```

Insert the following lines before the TERMINATE command.

```
WRITE   EXEC  queues should be paused and users warned about
WRITE   EXEC  shutdown. If not: type CTRL-C CTRL-A; then PAUSE queues
WRITE   [!READ  and send warning to users. If so: press NEW LINE.]
WRITE   Terminating EXEC.
```

This gives information and allows the person to stop the macro if needed. For completeness, you might add some other information near the end of the macro.

```
* FIND  "!ELSE" ↵

n     [!ELSE]
```

Insert the following lines before this !ELSE.

```
WRITE  EXEC   and all its son processes are terminated.
WRITE  To check for other processes use ? macro. To shut
WRITE  down system: type BYE NEWLINE then Y NEWLINE.
```

Then leave SED:

* BYE ↵
Do you want to save the original file as a backup file? YES ↵

Output file - :UTIL:DOWN.CLI
Backup file is - :UTIL:DOWN.CLI.BU

*)

As with UP.CLI, copy DOWN.CLI to the root directory:

*) MOVE/V/R : DOWN.CLI ↵
DOWN.CLI
*)

Testing UP and DOWN

Having edited the macros, try them in sequence. First, make sure all consoles you want to enable are turned ON and are ON LINE. Make sure the line printer(s) are ON LINE.

Then try UP:

*) UP ↵

Running QCMP now.
...(QCMP types messages)...
QCMP done.

PID 3
FROM PID 3 : EXEC REV n READY

FROM PID 3 : ENABLING ALL CONSOLES

.
.
.

FROM PID 3 : ALL CONSOLES ENABLED

FROM PID 3 : (EXEC) STREAM 1 CONTINUING
FROM PID 3 : (EXEC) @LPB CO-OPERATIVE STARTED
FROM PID 3 : (EXEC) @LPB CO-OPERATIVE CONTINUING

.
.
.
)

One of the last lines executed is EXECUTE CLI, which creates a son process under the master CLI. So the master CLI, PID 2, is no longer running on the system console. Check with WHO↵ or ?↵.

Go and check the consoles. Every console that is enabled for user log on will have a *TYPE NEW LINE TO START LOGGING ON* message on it. If any consoles that you expected to be enabled do not show this message, note their numbers; then return to the system console.

If the system console shows error messages like *ILLEGAL DEVICE OR CONSOLENAME FORMAT,* you probably made a syntax error in an ENABLE command. Type BYE↵ to get back to the master CLI, use SED to fix UP in :UTIL, and move it to the root. Then, terminate EXEC by issuing the DOWN macro, and try UP↵ again.

If an error message is *FILE DOES NOT EXIST,* the console was not defined via AOSGEN. If the console exists, you may need to re-execute AOSGEN. If the console doesn't exist, type TERM OP:EXEC↵, then fix UP as in the previous paragraph.

When the UP macro runs without errors and all the consoles you want are enabled, try bringing the multiuser environment DOWN.

The down sequence generally goes as follows

1. Warn users that EXEC is coming down, so they can get out of text editors or take other appropriate action. The CLI SEND command or BROADCAST macro is useful for this. You can use ?.CLI (created above) to check all user processes.
2. Pause EXEC's queues with the CX PAUSE command.
3. When all users have logged off or have only CLI.PR running on their consoles, get back to the OP CLI by typing BYE↵.
4. Run DOWN ↵ to terminate EXEC.

To do it, type

*) SEND @CON- System coming down NOW! ↵
FROM PID n : (OP) System coming down NOW!

(console displays NOT A CONSOLE DEVICE errors messages...
this is why we recommend BROADCAST instead of SEND)

*) ? ↵

PID: 1	*PMGR*	*PMGR*	*:PMGR.PR*
PID: 2	*OP*	*OP*	*:CLI.PR*
PID: 3	*OP*	*EXEC*	*:UTIL:EXEC.PR*
PID: n	*OP*	*n*	*:CLI.PR*

You are the following process:

PID: n	*OP*	*n*	*:CLI.PR*

*)

Each WHO command in ?.CLI returns the process ID, its username, its process name, and the pathname of the program it is running. There are no other processes now because no users are logged on. As you can see, ?.CLI is quite handy.

*) CX PAUSE 1 ↵
FROM PID 3 : (EXEC) STREAM 1 PAUSED

*) CX PAUSE @LPB ↵
FROM PID 3 : (EXEC) @LPB PAUSED

To bring down the system, you need to get back to the master CLI, PID 2. Do it with BYE↵ and check again:

*) BYE ↵
AOS CLI TERMINATING ...
*) ? ↵

You are the following process:

PID: 2 OP OP :CLI.PR
*)

As PID 2, with no users logged on, you can proceed with DOWN:

*) DOWN ↵

EXEC queues should be paused and users warned about.
EXEC shutdown. If not: type CTRL-C CTRL-A; then PAUSE queues
and send warning to users. If so: press NEW LINE.

Press ↵ to proceed:

Terminating EXEC.

PROCESS TERMINATION, PID: 3
ABORT
TERMINATED BY A SUPERIOR PROCESS

EXEC and all its son processes are terminated.
To check for other processes use ? macro. To shut down system, type BYE then Y.

time

*)

Type ?↵ to check on the processes. There should be only two: the peripheral manager and master CLI. The DOWN macro is simple, so the only error messages you get should be the result of syntax errors, fixable with SED. If you get errors, fix them with SED, then type UP↵, then DOWN↵ again.

You can, if you wish, make the CX PAUSE commands part of the DOWN macro. But this will cost something in versatility: there may be times when you want to issue EXEC commands other than PAUSE to the queues (as described in Chapter 8, "EXEC and User Processes").

Special Printing Forms — Creating the FORMS Directory

EXEC can handle not only general-purpose printing jobs, but special forms like invoices or paychecks. For these special printing jobs, users build format control commands into a disk file with the FCU (Format Control Utility). Then they submit the job for printing by typing a QPRINT/FORMS=form-file↵ command; the appropriate person puts the pertinent forms in the printer; and EXEC prints the forms with the format control specified in the form-file.

The form files must be in a directory named FORMS, beneath directory :UTIL. Even if you will not use special forms, it can't hurt to create this directory — you may want it later. So type

```
*) CREATE/DIR   :UTIL:FORMS ↵
*) ACL   :UTIL:FORMS   OP,OWARE   +,RE ↵
*)
```

This creates the directory named FORMS in UTIL. Then it sets the access control list (ACL) to allow username OP all privileges and to allow all users to read and execute (but not delete or modify) files in FORMS. FORMS is empty at this point — later, users may create forms files that you want to put in it. But you will not need to recreate FORMS.

PREDITOR and EXEC Summary

With user profiles, EXEC, and the macros done, the multiuser environment is practically complete.

If you wish to bring the system down, type BYE↵, then Y↵ — later, you'll need to bootstrap to bring it up. Assuming power stays on to your CPU and disks, the sequence from startup to normal shutdown will go

1. Bootstrap from disk, bring up AOS.
2. Type UP ↵ to CLI.
3. Multiuser environment runs; users log on and off.
4. Warn users, pause queues, type BYE ↵ and DOWN ↵.
5. Type BYE ↵, then Y ↵.

Actually, you can type BYE↵ at any time, with other processes running, from the PID 2 CLI. If you confirm, everything will be shut down. But this risks killing user processes prematurely and causing users to lose work.

Bootstrapping and normal (and abnormal) shutdown are detailed in the next chapter.

There are a number of issues and tools to learn about (described next). And there are other processes and commands you will eventually want to make part of your UP macro. But with user profiles, EXEC, and the ?, UP, and DOWN macros, the backbone of the multiuser environment stands. Congratulations.

Other DG Software

Along with AOS, you may have acquired other DG products, like the communications/networking products, Comprehensive Electronic Office (CEO) the COBOL, FORTRAN 77, Pascal, and/or PL/I languages, SWAT debugger, INFOS II or DG/DBMS data management systems, or others.

Each of these products comes on its own mag tape, with its own documentation. Instructions for loading and using it appear in the documentation and/or in the product Release Notice.

Some products, like XODIAC and CEO, have a specific home directory; others, like languages, can be placed where you wish: in their own directories, perhaps under :UTIL, or in :UTIL itself. Putting a product in its own directory keeps it in one place; and may allow faster access. But putting a product in :UTIL makes it easier for users to access because their search lists need include only :UTIL.

Some products have programs that must be executed with the PROCESS command; some have commands that the master CLI process or users can issue. Many products, like XODIAC and CEO, have UP macros of their own to ease operation. CEO may require a one-line edit of your UP.CLI macro to include the CEO directory in the searchlist, described in *Managing the CEO® System.*

For products other than XODIAC and CEO, after some experience, you may decide to place the pertinent PROCESS and other commands in the UP.CLI macro. This macro is a tool that summarizes the components of your entire system; you will find it, and perhaps different variations of it, very useful.

Error Handling — The ERMES File

DG strives to have a unique error code for every kind of error that can occur in every process running under AOS. Each code is a number. The system translates each code to a text error message via a file called ERMES.

The CLI uses ERMES to describe its own errors. And when any program executed from the CLI (with PROCESS or XEQ command) hits a fatal error, the program process terminates and returns control to its parent CLI. The process may describe the error before it terminates, or it may simply return a numeric error code to the CLI.

When the CLI receives an error code, it looks for the text definition in file ERMES, in the root directory. If ERMES defines the code, the CLI will find and display the text error message. But if the code is not defined in ERMES, the CLI will report UNKNOWN ERROR CODE n, where n is the code.

The ERMES file supplied with AOS contains error message text needed by AOS programs, like the CLI, EXEC, SED, the SPEED editor, the macroassembler, and Link. But the supplied ERMES does not define error codes for other DG software, like high-level languages (e.g., FORTRAN and COBOL), data management software (e.g., INFOS II, DG/DBMS and Sort/Merge), or communications/networking software.

After you have loaded all your DG-supplied software, someone must create an ERMES with text definitions for all its errors. (If this isn't done, users will often get only the numeric error codes, which won't help productivity.)

Creating the tailored ERMES involves editing a macro called NEWERMES to contain all the product names, executing NEWERMES, and copying the new ERMES into the root directory.

For example, assume you have acquired COBOL, FORTRAN 77, AOS BASIC, SORT/MERGE, and XODIAC networking. After loading these onto the system, you might create a new ERMES as follows.

First, turn SUPERUSER on and make the working directory :UTIL.

```
) SUPERU   ON ↵
*) DIR   :UTIL ↵
```

Now, with a text editor (SED or SPEED), edit file NEWERMES.CLI. The following text string in this file defines all the codes for ERMES:

```
<SYS,Z,SPEED,PL1,DEB,SYS_M,CLI_M,PASC_M,... >ermes %-%
```

From this text, you can see that the PL/I messages are already part of the system ERMES. Anywhere between the angle angle brackets, you'd add the proper strings. They are COB for COBOL, F77 for FORTRAN 77, BAS for BASIC, SORT for Sort/Merge, and NET for Networking. If all the text won't fit on one line, use the & continuation character to continue the line; then delete the existing first angle bracket. For example, change the original line to

```
<COB,F77,BAS,SORT,NET&
  SYS,Z,SPEED,PL1,DEB,SYS_M,CLI_M,PASC_M,...>ermes %-%
```

Then leave the editor. Execute the new NEWERMES by typing

```
*) NEWERMES ↵
```

... (Link message) ...

```
*) MOVE/V/R   :   ERMES ↵
```

ERMES

```
*)
```

Now users will get text explanations of COBOL, FORTRAN 77, BASIC, SORT, and XODIAC network runtime errors. (The /R switch copies the file only if it is more Recent than a file with the existing name in the root directory.)

If you get *FILE DOES NOT EXIST* messages from the NEWERMES macro command, MOVE copies of the pertinent .OB files from their own directories to :UTIL; then try NEWERMES ↵ again. The filenames, in this case, are COBERMES.OB, F77ERMES.OB, BASERMES.OB, and SORTERMES.OB, and NETERMES.OB.

ERMES instructions usually appear on the Release Notices of each relevant software product. Also, file :UTIL:ERMES.SR describes the format of the ERMES file.

Making Life Easier for Users

This section tells you how to create log-on messages and give and get Help.

User Log-On Files

When you specified the "INITIAL IPC FILE" to PREDITOR, you gave a filename (example, :UDD:username:LOGON.CLI) that would be executed for users when they logged on. There will be no error message if this file doesn't exist, but it serves as a useful information and control tool for users — so you might as well create it and see how it works.

Use a text editor (SED or SPEED) to create the file. Be sure to create the file in the same directory and with the same name that was specified to PREDITOR as the initial IPC file (shown earlier as :UTIL:LOGON.CLI). Sample text for the initial IPC file follows.

```
WRITE  Welcome to this new AOS Operating System.
WRITE  The file you are reading is a CLI macro in
WRITE  your own directory. You can edit it as you wish:
WRITE  to set searchlist and/or default access control list
WRITE  - DEFACL - for your files and/or reminders, etc.
SEARCHLIST  :UTIL
WRITE  This macro set your searchlist to [!SEARCHLIST]
WRITE  Type HELP NEW LINE for Help.
```

After writing the LOGON.CLI macro, test it by typing

```
*) LOGON ↵
```

You should see the text you typed after each WRITE command, and the current searchlist set by the SEARCHLIST command. If there are error messages, they probably result from syntax errors. The system reads only the first 512 characters in this file, so you shouldn't make it more than 512 characters long.

When the log-on macro runs without error messages, give everyone all access to the macro and move a copy of it to everyone's directory by typing the following commands.

```
*) MOVE/V  :UDD  LOGON.CLI ↵
LOGON.CLI
*) DIR  :UDD ↵
*) ACL  LOGON.CLI  +,OWARE ↵
*) SPACE  +  50 ↵
*) MOVE/V  ([!FILENAMES  +])  LOGON.CLI ↵
...(CLI verifies each file move)...
```

This sequence initializes each user directory and moves a copy of LOGON.CLI into each one — including OP.

Note that the macro text applies only to CLI users (users whose PROGRAM, specified to PREDITOR, is CLI.PR.) If a user's PROGRAM is BASIC.PR, LOGON.CLI must be rewritten in BASIC to PRINT the log-on message you want. A BASIC program cannot use CLI commands directly. CEO, however disregards the IPC file text, so you need not rewrite it for a CEO user.

LOGON.MESSAGE File

If you create a file called LOGON.MESSAGE in :UTIL, EXEC will display the contents of this file to every user who logs on — in whatever program.

LOGON.MESSAGE is quite useful for general system information, like planned shutdowns, new features, and so on. The system manager or operator can add information to this file as needed — any day or any hour. If you wish, you can give the file an access control list that allows any CLI user to edit it — adding messages that he or she feels would be of general interest. As with the initial IPC file, only the first 512 characters of LOGON.MESSAGE are read (but people can TYPE LOGON.MESSAGE↵ to see the whole thing, if it exceeds 512 characters).

Create and/or edit file LOGON.MESSAGE as you would any file. It must be in the same directory as EXEC (usually directory :UTIL). Sample text is

```
This is the wonderful world of AOS.
If you have questions, please see system operator.
```

After creating both the user log-on and LOGON.MESSAGE files, bring everything UP↵, then log on as OP on a user console. You will see the messages just as any user will see them.

Giving and Getting Help

AOS has a Help mechanism that can help inexperienced people to use the system. You may have used part of this with AOSGEN, PREDITOR, and SED; you can also use it with the CLI, EXEC, and other DG products. For example, type the following commands.

*) HELP ↵

...(CLI displays list of HELP topics)...

*) HELP *COMMANDS ↵

...(CLI displays list of its commands)...

*) HELP/V ACL ↵

...(CLI describes ACL command)...

*) XHELP ↵

...(EXEC displays all its commands)...

*) XHELP ENABLE ↵

...(EXEC describes its ENABLE command)...

You can see that HELP provides quick, pertinent information, when people need it. All help messages, for all programs, are in directory HELP. This directory was created, and its files loaded, when you first brought up the starter system.

Each help file begins with a character string that identifies it to the system. For example

The contents of a HELP file named	Will be displayed by the CLI command
CLI.TPC.TOPICS	HELP ↵
CLI.TPC.string	HELP *string ↵
CLI.CMD.string	HELP string ↵
CLI.PSM.string	HELP !string ↵
EXEC.string	XHELP string ↵

Additional help files in :HELP, accessible from nonCLI programs, are

Filename	Accessible from
SED.string	SED text editor.
AOSGEN.string	AOSGEN program.

By default, every user can read these files (via appropriate HELP or TYPE pathname commands), and the system will automatically find and display them.

Generally, if you want to create one or more help files and have the system type its contents, use the filename form

CLI.TPC.string

The system will then display the string in proper alphabetical order when a user types HELP↵. It will display the file's contents when a user types HELP *string↵.

If a help message can tell the whole story by itself, you can simply leave it as a help topic. But if the explanation is quite long, you can use the topic file to tell the user what files to TYPE for more help.

All HELP files you create should have an Access Control List (ACL) of at least +,R so the users will be able to read them. You can set ACLs as shown in the example below.

Even if you decide not to create your own HELP messages now, all files from nonAOS products that begin with "CLI.TPC." should be in directory :HELP. This will allow users to see the topic when they type HELP⤶.

You can find all CLI.TPC.- filenames via a filename *template* (character that matches all or parts of filenames). The character + (plus sign) matches all characters. So, in directory "dir", you can see which filenames begin with CLI.TPC. — and sort the filenames — by typing

```
*) DIR   dir ⤶
*) F/S   CLI.TPC.+ ⤶
.
.  (system lists all filenames that begin with CLI.TPC. alphabetically)
.
*)
```

HELP File Example

In directory :HELP, use a text editor to create a file named

CLI.TPC.ABOUTSYSTEM

contents of this file can be any help message you want. For example:

```
ABOUTSYSTEM — This is a DG S/280 CPU running the
AOS operating system. TYPE any of the following
files for more information: ⤶

:HELP:COMPILERS       Describes compilers and how to use them.
:HELP:BATCH           Describes using Batch.
:HELP:DAY_RUN         Describes schedule of application programs.
:HELP:DAY_DUMP        Describes daily schedule of Dump/Backup runs.
```

After leaving the text editor, give everyone read access to the help file:

```
*) ACL/V   CLI.TPC.ABOUTSYSTEM     +,R ⤶
```

CLI.TPC.ABOUTSYSTEM

```
*)
```

This shows the simplest approach to help. When any user types HELP⤶ from the CLI, ABOUTSYSTEM will appear as a topic. When the user types HELP *ABOUTSYSTEM ⤶, the file text will be displayed; the user can then TYPE any of the pathnames described within for more information. You'll be able to think up more sophisticated ways to use HELP later on. But the main point is to know about it: it can be a great boost to productivity.

Overview of Your System File Structure

Your multiuser environment is ready for users. During the whole process, there were a number of directories and files created — some via the system tape, some by the tailored system, some by PREDITOR and EXEC, and some by you.

Figure 5-1 shows the directory structure, with some pertinent files, and describes how and when the files were created. An oval indicates one or more directory files; a box indicates one or more individual files.

What Next?

This chapter has given you the essentials on creating the multiuser environment.

It has shown you how to create good general-purpose user profiles with the PREDITOR profile editor; how to initialize EXEC, the multiuser environment "manager"; given you some pointers on other DG software and the ERMES error file; shown you how to make life easier for users with initial user IPC files and the log-on message file; and offered a picture of your finished system's directory structure.

This chapter ends the "cookbook", blank-disk to finished-system portion of the book. This portion has introduced you to the DG hardware and software in Chapter 1; told you how to format LDUs, install, and bring up the starter system in Chapter 2 or 3; how to generate, test, patch, and optionally install a tailored system in Chapter 4; and how to create the multiuser environment in this chapter.

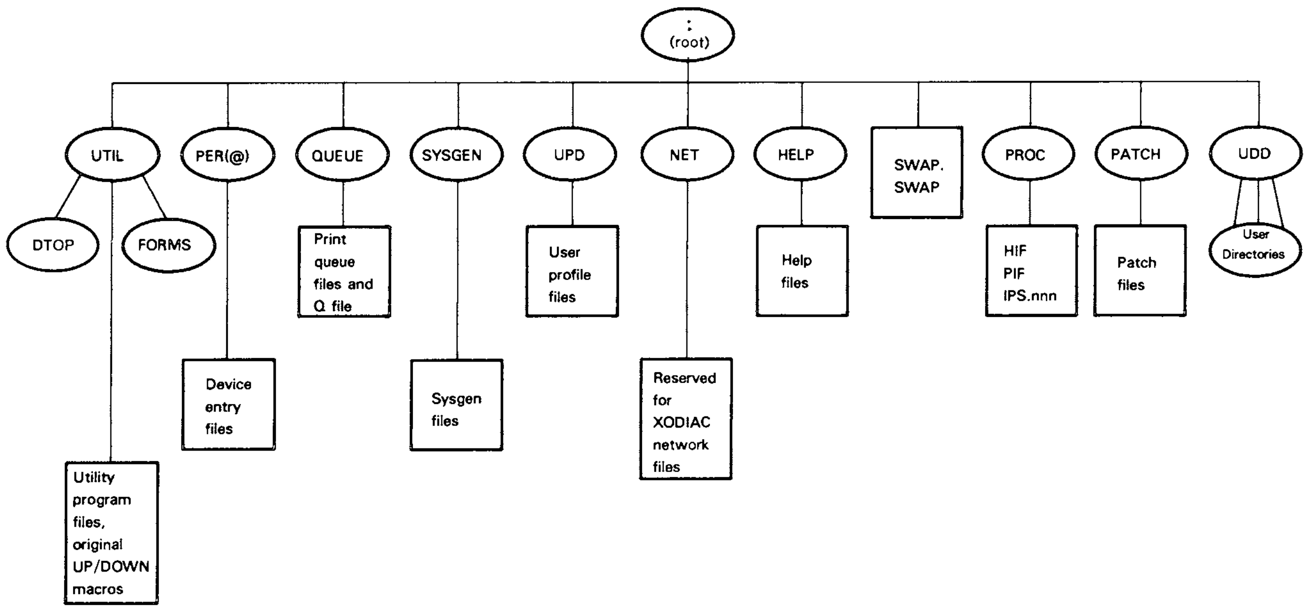
Whoever you are — DG engineer, system manager, DP manager, MIS manager, or nonadministrative person — you have done a tremendous job. Not only have you had to execute many steps, but you've had to learn a lot. Your tailored, multiuser system is up and running. You may have to execute parts of preceding chapters again — but you will rarely, if ever, need to do it all again for your installation.

The rest of this book gives the details — in a form designed for reference — of programs to help generate and run an AOS system. The next chapter details startup and shutdown, both normal and abnormal. Read it if you want to learn about bootstrapping, and/or handling abnormal shutdown. Chapters 7 and 8 give the *details* on PREDITOR and EXEC — batch, spooling, and tape mounting — read them for the inside story on these important programs. Chapter 9, "Other Runtime Tools", describes general-purpose operator tools, like the PED environment display program, the SYSLOG/REPORT utilities, and file backup.

Chapter 10 tells you how to handle unusual system conditions; Chapter 11 details the Disk Formatter; and Chapter 12 details the Installer. Chapter 13 gives some summary pointers and cautions.

Chapter 14 examines some system management issues and tries to show you how to make the best decisions for your installation.

Depending on your interest — go to the appropriate chapter, or check the index for a specific topic or product.



From the top, left to right, the directories are as follows

Directory name (pathname)	Description
ROOT (:)	This is the root directory. It contains all other directories. It was created by the Disk Formatter. Its nondirectory files include the GHOST, peripheral manager, stand-alone FIXUP disk fixer, and DFMTR Disk Formatter, the CLI program files, the edited UP.CLI and DOWN.CLI (which you moved there), and the error message file, ERMES.
UTIL (:UTIL)	This is the utilities directory. It contains most AOS utility program files: the SED text editor, macroassembler, Link, debugger, etc. UTIL also has operator tools like PREDITOR, EXEC, and PED.
DTOP (:UTIL:DTOP)	This is the DTOP directory, included in SYSGEN AOS (model 30505). It includes the entire AOS system, designed for those who want to create PREGEN diskettes for DESKTOP GENERATION systems. See Chapter 4.
FORMS (:UTIL:FORMS)	This is the FORMS directory that you created to hold special printing form directive files; it will be empty until someone places files in it.
PER or @ (:PER OR @)	This is the Peripherals directory, created by the AOS system each time it comes up. PER contains a device entry file for each device (not controller) generated to be part of the operating system (e.g., MTB0, DPF0, CONn, LPB. The @ is convenient shorthand for PER; for example @MTB0 is easier to type than :PER:MTB0. PER is also the home directory of EXEC (although the EXEC program and overlay files are in :UTIL). When AOS starts up, it creates device entries in PER. When EXEC starts up, it creates batch and device queues in PER. Other software products, like networking or communications products, also create entries in PER when they are started up. When a system is shut down normally, it deletes PER, so don't place user files or directories there.

DG-26900

Figure 5-1. Directory Structure in the Finished (AOS) System (continues)

QUEUE (:QUEUE)	This is directory QUEUE, created by EXEC the first time EXEC starts up. QUEUE contains a file named Q, which has queue information for EXEC, and temporary print queue files. The QCMP queue compactor may delete all files in Q, so don't place user files here.
SYSGEN (:SYSGEN)	This is the system-generation directory. It was created by the starter system from the system tape during the initial load procedure. It contains needed AOSGEN and library files for system generation.
UPD (:UPD)	This is the user profile directory, created by PREDITOR the first time someone runs PREDITOR on the master LDU. The user profile file for each user -- also created by PREDITOR -- lives here. EXEC checks this directory for a matching profile file before allowing a user to log on. And, after each user has logged on, the system itself enforces the limits set within the profile file.
NET (:NET)	This is the network directory, created by the operating system the first time the system comes up. XODIAC network files are put here automatically by XODIAC software. Generally, don't put user files here.
HELP (:HELP)	This is the HELP directory, created by the starter system from tape during the initial load procedure. It contains topic help files, command and pseudo-macro help files, and AOSGEN and EXEC help files. You can also place your own help files here.
SWAP.SWAP (:SWAP.SWAP)	The SWAP.SWAP file is part of AOS' memory management mechanism. AOS creates it at startup and deletes it at shutdown. Its size is specified at AOSGEN, but you can override this if you OVERRIDE DEFAULT SPECS when you bootstrap.
PROC (:PROC)	AOS uses the PROC directory to keep track of running processes. It is created at startup and deleted at shutdown. Don't place user files here.
PATCH (:PATCH)	This is the patch directory, created with patch files and an update notice, by the first AOS update tape loaded. Subsequent update tapes update the contents. Generally, restrict this directory to patch and update files; users should not store files here.
UDD (:UDD)	This is the user directory directory, created by PREDITOR the first time PREDITOR is run on this master LDU. In this directory, PREDITOR creates a user directory for each user given a profile. This directory has the name given as a username to PREDITOR; for example, JACK, F77, or INFOS. The user directory becomes the user's working directory when the user logs on; within it, the user can create files and subordinate directories.

DG-26900

Figure 5-1. Directory Structure in the Finished (AOS) System (concluded)

End of Chapter

Chapter 6
Startup and Shutdown

Read this chapter

- when you want to start up your computer system;
- when you want to bootstrap a program like an AOS system or the FIXUP Disk Fixer;
- when you want to shut an AOS system down;
- if AOS stops with a FATAL ERROR or HARD ERROR message;
- when power returns after a power failure.

This chapter gives the details on system startup, normal shutdown, and abnormal shutdown. It has the following major sections:

- CPU Front Panel
- Startup
- Normal Shutdown
- Abnormal Shutdown
- The FIXUP Disk Fixer
- Power Failures
- What Next?

CPU Front Panel

The switches on your computer's front panel affect how you bootstrap, the break sequence, power fail, and many other things. Generally, there are two types of computer front panel. They are

- Computers with a *programmed console*, such as the S/20, S/120, S/140, S/280, MV/8000, and DESKTOP GENERATION systems. On these computers, the front panel has its own memory, microcoded with a loader program that loads other programs. The front panel has three rocker switches (except MV/8000).

  With a programmed front console, you can bootstrap AOS by typing characters on the system console (CRT display or printer terminal). After you create your first AOS system, you can also bootstrap simply by turning computer power on with the LOCK switch in LOCK (except on MV/8000s).

- Computers with hardware data switches, such as all models not mentioned above. On these computers, the front panel has hardware data switches, numbered 0 through 15 (or X4/0 through 15), and a number of other switches. You set the data switches, then use the RESET switch and PR LOAD (PROG LOAD) switches to program load.

Programmed Console Break Sequence

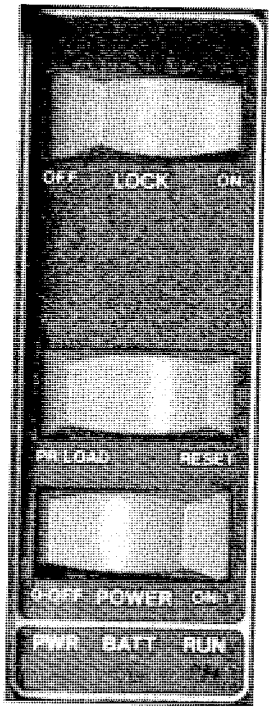
The break sequence stops the computer, freezing all system activity (except MV/8000), and gives control of the system console to front panel microcode. Generally, you will want to avoid the break sequence unless AOS is deadlocked.

The break sequence is a sequence of one or more keys typed on the system console. On a DASHER D/200 or D/400 console, the keys are CMD and BREAK/ESC (the two keys pressed simultaneously). On a DASHER printing console, the key is BRK. On a DASHER D2, the key is BREAK.

If you type the break sequence at the system console while the computer is unlocked, the computer suspends any program that is running (such as AOS), stops, and displays the *!* prompt (S/20, S/120, S/140, S/280, and DESKTOP GENERATION systems) or the *SCP-CLI>* prompt (MV/8000). To continue the program and resume normal system activity, type P ↵.

To avoid accidental break sequences, make sure the computer LOCK switch (if any) is in the LOCK position.

S/280 Panel Switches and Lights



PH-0691

Figure 6-1. ECLIPSE S/280 Computer Front Panel

S/280 computers — shown in Figure 6-1 — have a programmed front panel with three rocker switches: LOCK, PR LOAD/RESET, and POWER. The front panel has three lights that indicate faults by blinking.

The *panel* switches work as follows:

- LOCK switch. In the ON position, the LOCK switch disables the PR LOAD/RESET switch, the POWER OFF switch, and the break sequence. LOCK ON enables transfer to the backup battery (if any) if outside power goes down. LOCK also tells the hardware to program load from the preselected device code (if any) when power is turned on. You must turn LOCK off to turn power off with the rocker switch, to enable the break sequence, or to bootstrap from a device other than the one preselected. *Keep LOCK in the ON position unless you want to do one of these things.*

  If LOCK is off when you power on, a microcoded console loader program (*!* prompt) gets control. You must type nnL, where nn is the device code, to bootstrap from a device. For example, to bootstrap from tape, type 22L.

- PR LOAD/RESET switch. If the console loader program (*!* prompt) has control, pressing PR LOAD tells the hardware to load from the preselected device code. PR LOAD has no effect if the CPU is running (as when AOS is running). Pressing RESET resets the computer, if unlocked. *Don't press RESET if AOS is running.*
- POWER switch. This switch should be at the ON-1 position, unless you want to cut CPU power.

The lights on the S/280 panel work as follows:

PWR light	On when dc power is normal; off when power is off or the computer is under partial battery backup.
BATT light	On when computer has transferred from normal power to backup battery (full or partial backup).
RUN light	On when the computer is executing instructions (AOS, DTOS, etc.); off when the computer is halted.

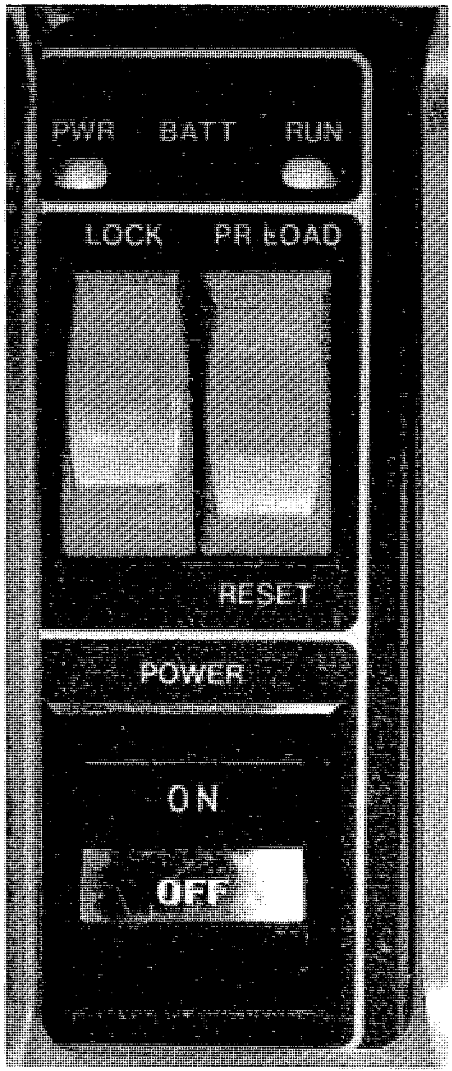
When the computer is running AOS on normal power, the PWR and RUN lamps are on. When the POWER switch is OFF, all lamps are off. When one or more lights *blink*, there is a power supply fault.

Table A-3 in Appendix A explains all the power supply error codes, what they mean, and what to do to cope with them.

In the normal course of system operations, you don't need any panel switches. To cut CPU power, press LOCK OFF and rocker switch POWER OFF. To power up, press LOCK ON and rocker switch POWER ON. Otherwise, avoid touching the switches.

S/140, S/120, and S/20 computers — shown in Figure 6-2 — have a programmed front panel with three lights: PWR, BATT, and RUN, and three switches. The lights work as follows:

PWR light	On when dc power is normal; off when power is off or the computer is under partial battery backup.
BATT light	On when computer has transferred from normal power to backup battery.
RUN light	On when the computer is executing instructions (AOS, etc.); off when the computer is halted.



PH-689

Figure 6-2. S/140, S/120, or S/20 Computer Front Panel

When the computer is running AOS on normal power, the PWR and RUN lamps are on. When the POWER switch is OFF, all lamps are off.

The three panel switches are LOCK, RESET/PR LOAD, and POWER. The switches work as follows:

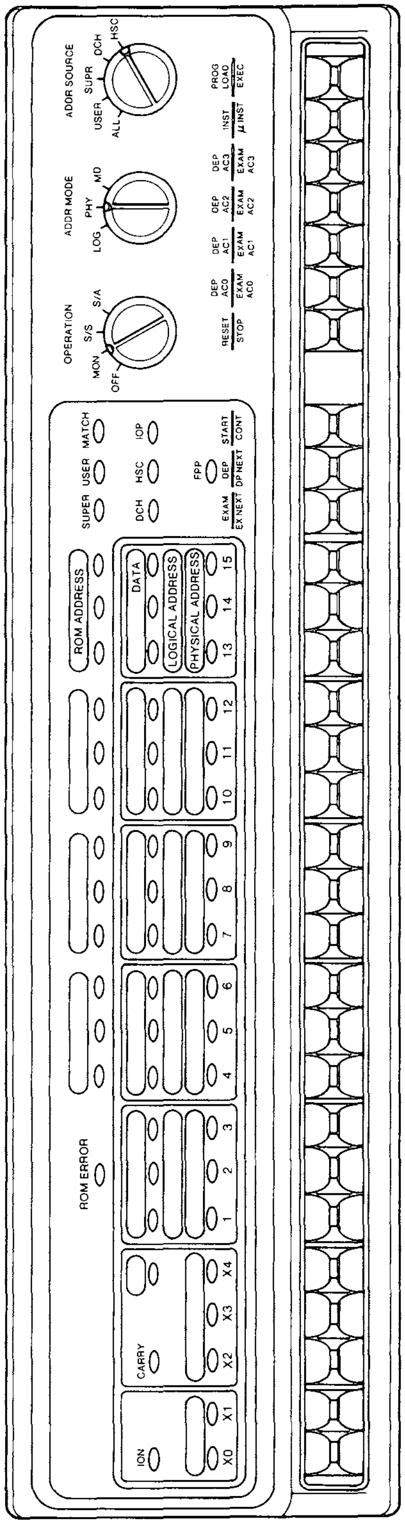
- LOCK switch. In the LOCK position, the LOCK switch disables the RESET/PR LOAD switch and POWER OFF switch. LOCK also tells the hardware to program load from the preselected device code (often 27 or 33) when power is turned on. You must unlock LOCK to turn power OFF or to bootstrap from a device code other the one preselected. *We recommend that you keep the computer locked.*
- RESET/PR LOAD SWITCH. Press RESET to reset the computer if it is unlocked. *Don't press RESET if AOS is running.*
- POWER. This should be ON, unless you want to cut CPU power.

Computers with Data Switches

ECLIPSE computers with hardware data switches include C/350, C/150, M/600, S/230, S/130, and all S/200 and C/300 series machines. The data switches are numbered 0 through 15 (or X4/0 through 15). To program load from a disk or tape, you set switch 0 up (to indicate a high-speed device), and then set switches 10-15 according to the device code you want. Then, you use the RESET switch and PR LOAD (PROG LOAD) switches to program load.

Switches other than data switches, like RESET/STOP and START/CONTINUE are three-position. To reset the CPU, lift the RESET switch. To stop the CPU press the RESET switch down.

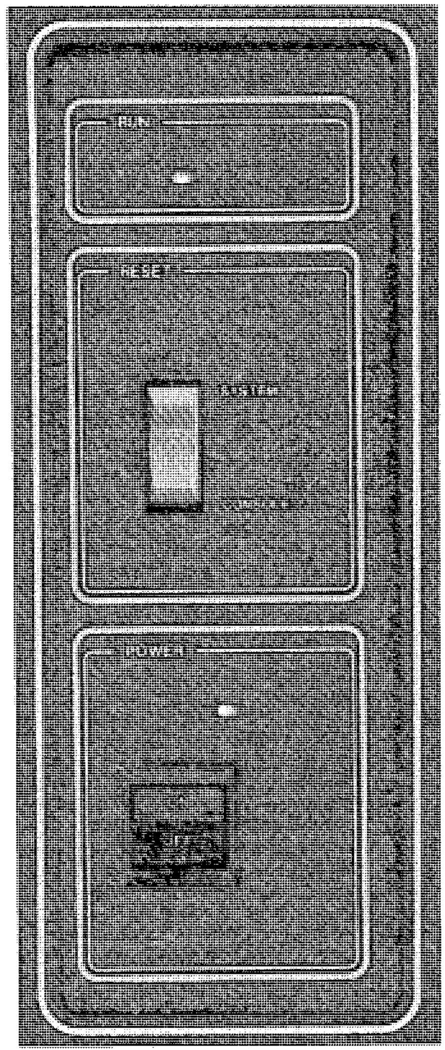
The power switch is a button (CPU PWR) or a key (POWER). You can lock the CPU by turning the LOCK (or POWER) key to the LOCK position. In LOCK, power is locked on and other switches are disabled. When you cold start, the computer must be unlocked; then turn power on and use the RESET and PR LOAD (PROG LOAD) switches to bootstrap. Later, when AOS is running, you should lock the computer. For powerfail to work (if you have a backup battery), and to help safeguard the system, the computer *must* be locked.



SD-00971

Figure 6-3. Example (Composite) Front Panel, Computer with Data Switches

ECLIPSE MV/8000 Panel Switches and SCP



PH-688

Figure 6-4. ECLIPSE MV/8000 Computer Front Panel Switches

As Figure 6-4 shows, MV/8000 computers have a programmed front panel (run by a microNOVA computer called the SCP). The panel has one light (on when power is on) and two front panel switches: RESET and POWER. The switches work as follows:

- RESET switch. Pressing this to SYSTEM resets the CPU; *don't do it if AOS is running.* Pressing this to CONSOLE has the same effect as a break sequence.
- POWER switch. This switch should be ON, unless you want to cut CPU power. Generally, avoid touching the MV/8000 front panel switches.

Typing the break sequence (CMD and BREAK keys, or BRK key, or BREAK key, depending on console type) gives the SCP control. Avoid typing the break sequence unless you must (as for an AOS deadlock). You can disable break with the SCP CLI FLAGS command. It will stay disabled while power stays up or until you re-enable it. If you really need the SCP CLI, you can get to it by pressing the RESET switch to CONSOLE.

If you find the *SCP-CLI>* prompt on the console when AOS is running, you can return control to AOS by typing TTY). For example,

SCP-CLI> TTY)

)

SCP-CLI Commands (MV/8000 only)

The SCP operating system, with its CLI, runs in the SCP processor (which is a microNOVA computer). The SCP system offers many commands, detailed in the manual *ECLIPSE MV/8000 System Control Processor*, 014-000649. For your convenience, here are a few of the most useful commands.

Command	What it Does
BOOT n	Program loads from the first device on device code n. Be sure the main process is RESET) before using. BOOT can be abbreviated B. An example is BOOT 27).
CONTINUE	Tells the main processor to continue running a halted program. Useful when you want to rerun a stand-alone program like the Disk Formatter without rebooting. Can be abbreviated CO.
FLAGS arg {YES / NO}	Sets or clears an SCP flag. The most common flags are SCOPE and LOCK. SCOPE tells the SCP to erase characters when you press the DEL key; it's useful if the system console is a CRT. LOCK disables the break sequence, which can be useful. An example is FLAGS SCOPE Y).
HELP *[name]*	Gives a general help message (without *name*) or gives help on command *name*.
HALT	Halts the main processor CPU after it has completed the current instruction. Can be abbreviated HA.
RESET	Halts and resets the main processor CPU. Use RESET before booting or running Emergency Shutdown (ESD). Can be abbreviated R. (See Chapter 6 for information on running ESD.)
START addr	Starts the main processor CPU at addr. The CPU must be halted. This command is useful under unusual circumstances for emergency AOS shutdown. An example is START 14).
TTY	Gives the main CPU program control of the system console.
(period)	Displays the status of main processor accumulators, program counter, carry, and map. This is useful whenever you want to check these or see if the SCP-CLI is active.

Startup

Startup includes turning power on (if off), initializing an AOS operating system or other stand-alone program, and bringing up the multiuser environment. It assumes that AOS was shut down normally (not by a fatal error or power fail). If abnormal shutdown occurred, see "Abnormal Shutdown".

A warm start is different from a cold start (power turned off to CPU).

Warm Start, CPU Running

You can *warm start* your system if AOS was shut down normally and power remained on to the CPU.

1. Make sure all disks are on, write enabled, and ready.

2. Reset the main processor and boot from the disk as follows:

2a. On an S/280, S/120, S/20, or DESKTOP GENERATION system, next to the *!* prompt, type nnH, where nn is the device code. For example:

```
!  27H        (S/280, S/120, or S/20)
```

or

```
!  26H        (DESKTOP GENERATION systems)
```

and go to step 14 under "Cold Start, CPU Power Off".

2b. On an S/140, next to the *!* prompt, type 11A. The system displays a number, which you can ignore. After this number, type 1000nn ↓, where nn is the device code. Then after the *!* prompt, type 1000nnL. For example:

```
!  11A   xxxxxx   100033 ↓      (S/140)
!  100033L
```

and go to step 14 under "Cold Start, CPU Power Off".

2c. On an MV/8000, next to the *SCP-CLI>* prompt, type RESET ↓. After the *!* prompt, type BOOT nn ↓, where nn is the device code. For example:

```
SCP-CLI>  RESET ↓        (MV/8000)
SCP-CLI>  BOOT  27 ↓
```

and go to step 14 under "Cold Start, CPU Power Off".

2d. With hardware data switches, unlock the computer (if locked). Make sure the data switches are set to 1000nn, where nn is the device code. Lift the RESET switch, lift the PROG LOAD (PR LOAD) switch, lock the computer, and go to step 14 under "Cold Start, CPU Power Off."

Cold Start, CPU Power Off

1. Make sure the system console is on and on line.

2. If your system is a DESKTOP GENERATION, S/20, or ECLIPSE C/30 (any microECLIPSE) computer, continue with the following steps. On a different computer, skip to step 6.

3. If your system has a tape unit, make sure unit power is on. If it's a model 10/SP with printer, make sure the printer is turned on and is on line.

4. Turn computer unit power on. The system console may display test characters. Wait for it to display the *!* prompt.

5. After the *!* prompt, type 26H — for example:

```
!  26H
```

Skip to step 14.

6. If you have an S/280, S/140, S/120, or ECLIPSE MV/8000, or an ECLIPSE computer with data switches, make sure the disk(s) are on, started (for removable disks), write enabled, and ready.

7. On the computer panel, make sure that the LOCK switch is at the locked position. (If the computer has a lock *key*, skip this step.)

8. Press or turn the POWER switch to "ON."

   On a computer with data switches, skip to step 13a.
   On an MV/8000, skip to step 12a.
   On any other computer, the system console will display either

   SPECIFY EACH DISK IN THE LDU
   DISK UNIT NAME?

   or

   !

9. If you see the *SPECIFY* message, skip to step 14. If you see *!* prompt, program load as follows.

10. On an S/280, or S/120, S/20, or DESKTOP GENERATION system, type nnH, where nn is the device code. The code is 27 for DPF disks, 24 for DPJ disks, 33 for DPI disks, or 26 for DPN or DESKTOP GENERATION disks. For example:

    ! 27H

    Skip to step 14.

11. On an S/140, type 11A. The system will display a 6-digit number, after which you type 1000nn ↵, where nn is the device code (27, 24, or 33). Then, type 1000nnL. For example:

    ! 11A *xxxxxx* 100027 ↵ (on S/140)
    ! 100027L

    Skip to to step 14.

12a. On an MV/8000, EPROM does some power-up tests. When these succeed, the system console displays

    CONSOLE READY

    MV/8000 SYSTEM CONTROL PROGRAM REV n
    TYPE HELP<CR> FOR HELP

    CHECK-SUM OK
    COPYRIGHT (C) DATA GENERAL ...
    ALL RIGHTS RESERVED.

    STARTING POWER UP SEQUENCE

    ENTER DATE (MO DAY YR)

    (If one or more power up tests fail, the console will show only a partial ***CONSOLE READY*** message, or none at all; and the SCP octal debugger prompt (*!*) will appear. Turn power off and on again. If the problem recurs, see Appendix A, "Power Up Error Codes.")

12b. Enter the date in numeric form. For example, for April 15, 1985, type

    4 15 85 ↵

    ENTER TIME (HR MIN)

12c. Enter the time using a 24-hour format. For example, for 2:30 p.m., type

    14 30 ↵

    MICROCODE (1= STD, 2= C350/MMPU[1])?

12d. You want to load the C/350 microcode, so type

2 ↓

LOADING FROM FILE MVC350 REV. n

.
. (microcode status messages appear here)
.

BEGIN SYSTEM INITIALIZATION
 # OF 256 KB MEMORY MODULES - n
END SYSTEM INITIALIZATION

SCP-CLI>

The SCP and microcode load takes a little more than a minute. If you see an error message, make sure the diskette is inserted properly; turn the power off and on again. If the error recurs, try another diskette, or refer to Appendix A.

12e. Type RESET; then boot from the device code of your master LDU. For example:

SCP-CLI> RESET ↓
SCP-CLI> BOOT 27 ↓ (or 33 ↓ or other code)

Skip to step 14.

13a. With hardware data switches, unlock the computer (if locked).

13b. Make sure data switches are set to 1000nn, where nn is the device code. For tape (device code 22), set switches 0, 11, and 14 in the up position. For a DPF-type disk (device code 24), set switches 0, 11, 13, 14, and 15 in the up position. For a DPJ-type disk (device code 24), set switches 0, 11, and 13 in the up position. For a fixed-head disk or model 6102, 6105, 6222, or 6271 (device code 26), set switches 0, 11, 13, and 14 in the up position. For any other disk (device code 33), set switches 0, 11, 12, 14, and 15 in the up position.

13c. Lift the RESET switch, lift the PROG LOAD (PR LOAD) switch, and lock the computer.

14. The system console displays

SPECIFY EACH DISK IN THE LDU
DISK UNIT NAME?

If this *SPECIFY...* message doesn't appear, type the break sequence and the reset sequence. Then repeat the program load step before this step.

If on an MV/8000 the console hangs, you probably loaded the wrong microcode. Either power down and cold start (step 6), or perform the following steps to load the correct microcode:

SCP-CLI> XEQ MCODE ↓

SCP-MCODE> LOAD MVC350 ↓

LOADING FROM FILE MVC350 REV. n

.
. (microcode status messages appear here)
.

SCP-MCODE> CLI ↓

SCP-CLI> RESET ↓

SCP-CLI> BOOT 27 ↓

15. Type the unit name of the disk. For device code 26, this is DPK0 or DPN0. For device code 27, it's DPF0. For code 24, it's DPJ0. For code 33, try DPI0. For example:

    DPN0 ↵ (or DPF0 ↵, or DPJ0 ↵)

    DEVICE CODE?

16. Press ↵ for the default.

    ↵

    SYSTEM PATHNAME?

17. At this point, you can specify an AOS system or stand-alone program like the Disk Formatter (pathname DFMTR). We'll assume you want an AOS system. Type the pathname (usually :SYSGEN:sysname.SY ↵); or, for the installed system, press ↵. For example:

    :SYSGEN:SYS_7.00.SY ↵

    AOS REV n

    DATE (MM/DD/YY)?

18. Enter the date as numbers for month, day, and year. Spaces or slashes can separate each number. For example, for April 15, 1985, type

    4 15 85 ↵

    TIME (HH:MM:SS)?

19. Enter the time, based on a 24-hour format, in hours, minutes, and seconds. (Minutes and seconds are optional. If you omit them, the system sets each to 00.) Use spaces or colons to separate each number pair. For example, for 2:35 p.m., type

    14 35 ↵

    OVERRIDE DEFAULT SPECS [N]?

20. SPECS means the parameters in the system specification file created during AOSGEN. These parameters include the following:

    - NUMBER OF BUFFERS IN CACHE (number of system buffers)
    - SWAP FILE DEFINITION (swap file size or device)
    - INITIAL LOAD (used when you are building a new AOS system on this LDU).

    If you wish to override any DEFAULT SPEC, answer Y ↵; the system will then ask about each one and you can take the default, or specify a new value. In most cases, though, you will not want to override the DEFAULT SPECS, so you will press

    ↵

    (pause)

    AOS CLI REV n
)

21. Make sure your line printers are on line, with paper aligned.

22.) UP ↵ (Bring up multiuser environment.)

    .
    . (messages from EXEC and other processes)
    .
)

EXEC and the multiuser environment are up; users can log on; and you can bring up other processes and/or issue EXEC and CLI commands (described in Chapter 8, "EXEC and User Processes" and Chapter 9, "Other Runtime Tools") as needed.

Now, with a Model 10/SP, you might try to type a lowercase letter (for example, q) just to make sure the system console has full capability. Make sure that the ALPHA LOCK status light above the keyboard is off (if it's on, press the ALPHA LOCK key once). Then type the letter Q. If the screen displays q, this means that the system console has full capability and everything is okay. If the screen displays Q, press the ALPHA LOCK key and retype Q. If the screen still displays Q, the terminal emulator isn't operating. You should copy the appropriate emulator to the hard disk as described in Chapter 4.

If, with a Model 10/SP, you want to enable the 7-bit mode to run user programs that require the 7-bit character set, use the macro ENABLE7BIT.CLI in :UTIL (type :UTIL:ENABLE7BIT ↵). Later, after all the programs that require the 7-bit set have run, restore normal 8-bit handling, using DISABLE7BIT (also in :UTIL).

Normal Shutdown

Normal shutdown means orderly shutdown from an active multiuser system to the SYSTEM SHUTDOWN message, and, optionally, to turning off power to devices.

These shutdown steps assume that EXEC is running and that multiple users are logged onto the system.

1. Send a message to all users indicating that the multiuser environment will be coming down. You could use the BROADCAST macro (Chapter 5) for this.

) SEND/2=I @CON- System coming down in 5 min. Please logoff. ↵
   or
) BROADCAST System coming down in 5 min. Please logoff. ↵

2. Use the EXEC PAUSE command to pause batch streams and spool queues; for example:

) CX PAUSE ↵
   .
) CX PAUSE @LPB ↵
   .
)

3. Use the ? macro (Chapter 5) to verify that users have logged off. Use SEND (or BROADCAST) and ? until all users who stand to lose work are logged off. CLI users will not lose anything when you bring EXEC down (although they might be annoyed if they are not notified). People using text editors — and perhaps user processes running application programs — *will* lose work if you terminate EXEC, so you should try to get them out of their editors or programs before you do it. For example, type

) ? ↵

PID: 1 PMGR	*PMGR*	*:PMGR.PR*
PID: 2 OP	*OP*	*:CLI.PR*
PID: 3 OP	*EXEC*	*:UTIL:EXEC.PR*
PID: 4 JACK	*004*	*:CLI.PR*
PID: 5 SALLY	*005*	*:CLI.PR*
PID: 6 SALLY	*006*	*:UTIL:SED.PR*
PID 20 OP	*020*	*:CLI.PR*

```
) SEND    @CON-   System coming down in 2 min. Please Logoff now. ↵
 or
) BROADCAST   System coming down in 2 min. Please Logoff now. ↵
) ? ↵
```

PID: 1 PMGR	*PMGR*	*:PMGR.PR*
PID: 2 OP	*OP*	*:CLI.PR*
PID: 3 OP	*EXEC*	*:UTIL:EXEC.PR*
PID 20 OP	*020*	*:CLI.PR*

)

4. Eventually, all users will have logged off or be in the CLI. Now, you need to get back to PID 2, the master CLI. Type

) BYE ↵

then

) WHO ↵

If the answer is PID 2, go to step 5.

4a. If you are running a locked LOCK_CLI (a lockable CLI, described in the "Other Runtime Tools" chapter), LOCK_CLI will remain an active PID on the system console: the BYE command does not affect it. To terminate LOCK_CLI, you must type UNLOCK↵, the password, and BYE↵. For example:

) WHO ↵	
PID 20 OP LOCK_CLI.PR	(Running LOCK_CLI)
) UNLOCK ↵	(Start to UNLOCK it.)
PASSWORD ↵	(Password doesn't echo)
BYE ↵	(Get back to master CLI.)
AOS CLI TERMINATING date time	
) WHO ↵	
PID 2 OP :CLI.PR	(Back in the master CLI.)
)	

5. Run the DOWN macro to bring EXEC down.

) DOWN ↵

...(messages, etc., if you put them in DOWN.CLI)...

PROCESS TERMINATION, PID: 3

ABORT

TERMINATED BY A SUPERIOR PROCESS

...(more messages, if you put them in DOWN.CLI)...

6. With EXEC terminated, check the processes again with ?. There may be only two processes left: the peripheral manager and the master CLI.

If processes like CEO, XODIAC, and INFOS II are still running, terminate these normally (for CEO, type CEO.SYSTEM STOP↵; for XODIAC, type DOWN.NETWORK↵, and so on). If there are any other processes (e.g., application-based processes), terminate *these* normally. Eventually, you may want to put all the process-terminating commands needed in the DOWN.CLI macro.

When ready, start shutdown by typing

) BYE ↵

DO YOU REALLY WANT TO SHUT THE SYSTEM DOWN?

7. This message gives you a chance to change your mind. To keep it running, type N↵; to shut it down, type

Y ↵

STARTING SYSTEM SHUTDOWN date time

SYSTEM SHUTDOWN

With programmed console computers, the system prints

nnnnnn
! (or *SCP-CLI>* on MV/8000)

AOS has shut down and the CPU has halted. If AOS's last message is *ABNORMAL SHUTDOWN,* see the next section.)

Note that PID 2, the master CLI, can always shut the system down directly via BYE↵. If so, AOS will say

YOU HAVE SONS. DO YOU WANT TO TERMINATE?

If you answer Y ↵, all processes below PID 2 will be terminated, the system will ask for confirmation again, and then shut down. If any processes other than EXEC are running, this can be a very dangerous way to shut the system down. For example, certain processes that use databases (like CEO) depend on normal shutdown to close the database properly. If shut down improperly, the structure and/or the integrity of the database may be compromised. So, use this quick method only if processes that would not be jeopardized (like CLI processes) are running.

After shutdown, you can boot an AOS system as described under "Warm Start" above; or you can bootstrap another system or stand-alone program. Or you can leave everything as is; or you can cut power to any or all devices.

To power down your disk(s), press disk switches STOP and DC POWER OFF for removable-pack disk units; or press READY for sealed disk units. To power down the system console, press the knob near the lower right, or for a hardcopy console, the switch under the keyboard to the right. To power down tape units, press POWER OFF.

If you power down the computer, you will have to reload later. Thus the next start will be a cold start. If you decide to cut power to the computer, press the LOCK switch (if any) to OFF or unlock, and press the rocker POWER switch to OFF.

Figure 6-5 summarizes startup and normal shutdown.

For a warm start, make sure all disks are on, write enabled, and ready.

- On an S/280, S/120, S/20, or DESKTOP GENERATION system, skip to step 10.
- On an S/140, skip to step 11.
- On an MV/8000, skip to step 12e.
- On a computer with hardware data switches, skip to step 13a.

For a cold start, start with step 1.

1. Make sure the system console is on and on line.
2. If your system is a DESKTOP GENERATION, S/20, or ECLIPSE C/30 (any microECLIPSE), continue with the following steps. On a different computer, skip to step 6.
3. If your system has a tape unit, make sure unit power is on. If it's a model 10/SP with a printer, make sure the printer is turned on and is on line.
4. Turn computer unit power on. The system console may display test characters. Wait for it to display the *!* prompt.
5. After the *!* prompt, type 26H and skip to step 14.
6. If you have an S/280, S/140, S/120, or ECLIPSE MV/8000, or an ECLIPSE computer with data switches, make sure the disk(s) are on, started (for removable disks), write enabled, and ready.
7. On the computer panel, make sure that the LOCK switch is at the locked position. (If computer has a lock *key*, skip this step.)
8. Press or turn the POWER switch to "ON."

   On a computer with data switches, skip to step 13a.
   On an MV/8000, skip to step 12a.
   On any other computer, the system console will display either

   SPECIFY EACH DISK IN THE LDU
   DISK UNIT NAME?

   or

   !

9. If you see the *SPECIFY* message, skip to step 14. If you see *!* prompt, program load as follows.
10. On an S/280, or S/120, S/20, or DESKTOP GENERATION system, type nnH, where nn is the device code. The code is 27 for DPF disks, 24 for DPJ disks, 33 for DPI disks, or 26 for DPN or DESKTOP GENERATION disks. Then, skip to step 14.
11. On an S/140, type 11A. The system will display a 6-digit number, after which you type 1000nn ↓, where nn is the device code (27, 24, or 33). Next, type 1000nnL. Then, skip to step 14.

12a. On an MV/8000, EPROM does some power-up tests. When these succeed, the system console displays

CONSOLE READY

.
.
.

DG-26901T

Figure 6-5. Startup-Shutdown Summary (continues)

12b. *ENTER DATE (MO DAY YR)* 4 15 85 ⤶ (Enter correct date.)

12c. *ENTER TIME (HR MIN)* 14 30 ⤶ (Enter time.)

12d. *MICROCODE (1= STD, 2= C350/MMPU[1])?* 2 ⤶

.
.
.

12e. *SCP-CLI>* RESET ⤶
SCP-CLI> BOOT 27 ⤶ (or 33 ⤶, or other code)
Skip to step 14.

13a. With hardware data switches, unlock the computer (if locked).

13b. Make sure data switches are set to 1000nn, where nn is the device code. For tape (device code 22), set switches 0, 11, and 14 in the up position. For a DPF-type disk (device code 24), set switches 0, 11, 13, 14, and 15 in the up position. For a DPJ-type disk (device code 24), set switches 0, 11, and 13 in the up position. For a fixed-head disk or model 6102, 6105, 6222, or 6271 (device code 26), set switches 0, 11, 13, and 14 in the up position. For any other disk (device code 33), set switches 0, 11, 12, 14, and 15 in the up position.

13c. Lift the RESET switch, lift the PROG LOAD (PR LOAD) switch, and lock the computer.

14. The system console displays
SPECIFY EACH DISK IN THE LDU

15. *DISK UNIT NAME?* DPF0 ⤶ (or DPN0 ⤶ or other master LDU name)

16. *DEVICE CODE?* ⤶

17. *SYSTEM PATHNAME?* :SYSGEN:sysname.SY ⤶ (pathname or ⤶ to load installed system)

AOS REV n

18. *DATE (MM/DD/YY)?* 4 15 85 ⤶

19. *TIME (HH:MM:SS)?* 14 35 ⤶

20. *OVERRIDE DEFAULT SPECS [N]?* ⤶
AOS CLI REV n
)

21. Make sure line printers are on line, with paper aligned.

22.) UP ⤶ (Bring up multiuser environment.)

.
. (messages from EXEC and other processes)
.
)

DG-26901T

Figure 6-5. Startup-Shutdown Summary (continued)

Normal operation proceeds here; then, you want to shut the system down. Follow these steps.

1.) BROADCAST System coming down in 5 min. Please logoff. ↲

   or

) SEND/2=1 @CON- System coming down in 5 min. Please logoff. ↲

2.) CX PAUSE ↲

   ...

) CX PAUSE @LPB ↲

   ...

)

3. Use the ? macro and SEND/BROADCAST until all text editor and other users who stand to lose work are out of their programs.

4.) BYE ↲

) WHO ↲

   If answer is PID 2, go to step 5. Else

4a. Type UNLOCK ↲, password, BYE ↲ for LOCK_CLI.

5.) DOWN ↲

   ...

   PROCESS TERMINATION, PID: 3
   ABORT
   TERMINATED BY A SUPERIOR PROCESS

   ...

6.) BYE ↲

7. *DO YOU REALLY WANT TO SHUT THE SYSTEM DOWN?* Y ↲

   STARTING SYSTEM SHUTDOWN date time
   SYSTEM SHUTDOWN
   CPU HALTED

   Power down devices if desired; for CPU, lock switch to OFF or unlock first.

DG-26901T

Figure 6-5. Startup-Shutdown Summary (concluded)

Abnormal Shutdown

An abnormal shutdown is any shutdown not executed by the BYE↵ command from the master CLI, PID 2, on the system console. If you get an *ABNORMAL SHUTDOWN* message during a normal shutdown then it — too — is an abnormal shutdown.

An abnormal shutdown can result from a deadlock (hang), panic (FATAL AOS ERROR), hardware failure (which may cause a panic), or power failure.

If you have an MV/8000 and you experience a hang or a panic, perform the following steps:

1. Verify the microcode and the scratch pad, by typing the following commands to the SCP:

   SCP-CLI> HALT ↵

   SCP-CLI> XEQ MCODE ↵

   SCP-MCODE> VERIFY MVC350 ↵

   VERIFYING MCODE FROM FILE MVC350 REV. nn

   (This will take about a minute.)

   VERIFICATION COMPLETE

   SCP-MCODE> VSPAD MVC350 ↵

   (This will take a few seconds.)

   VERIFICATION COMPLETE

   SCP-MCODE> CLI ↵

   SCP-CLI>

2. Examine the results of step 1 to make sure that the disk image matches the microcode or scratch-pad locations. Differences between the microcode or scratch-pad locations and the disk image indicate probable hardware problems, except for the following cases:
   - location 147 (octal) of the scratch pad contains the CPUID and probably does not match the contents of the disk. You can ignore this error.
   - the location in the disk file contains a 0, but the scratch pad location contains a nonzero value. This type of error is normal, and is simply a result of normal scratch-pad initialization. You can ignore this difference.
3. Attach the results of verifying the microcode and the scratch pad to the Software Trouble Report.

No matter what type of computer you have, there are several software tools to help you handle and recover from abnormal shutdown situations. They are

- the Memory Dump routine, which copies main processor memory to tape for later analysis;
- the AOS Emergency Shutdown (ESD) routine, which can turn the abnormal shutdown into a normal shutdown;
- the AOS disk fixer, FIXUP. FIXUP corrects disk inconsistencies and allows you to restart AOS. It takes much longer than ESD. You will usually run FIXUP only if ESD fails.
- Diagnostic Operating System (DTOS) tests (MV/8000 only). Usually, only DG field engineers use DTOS. However, under some circumstances, you may want to run DTOS Field Replaceable Unit (FRU) tests yourself. In this case, consult you field engineer for more information on DTOS.

System Deadlocks

If AOS seems to be denying service, it may be hung in a deadlock. Deadlocks can result from high-priority and/or resident processes that loop, monopolizing CPU time. Deadlocks can also occur if three or four batch streams are running with heavy traffic along with many interactive processes.

The primary symptom of a deadlock is long response time. Another symptom is a process (like a text editor) that is not aborted by the abort sequence CTRL-C CTRL-B. Users may complain that nothing is happening on their consoles; and/or there may be little or no response to commands given at the system console. (If this console shows a FATAL AOS ERROR message, a panic has occurred; see the next section.)

If the system is very sluggish, XEQ PED (the Process Environment Display program, described in Chapter 9, "Other Runtime Tools") to see if any process is getting too much CPU time. If PED reveals such a process, TERMINATE it from the master CLI; this may resolve the deadlock.

If no process seems guilty, or the deadlock persists, check the batch queue status with the CLI command QDISPLAY/V or the EXEC command CX STATUS. If three or four batch streams are active with big jobs, you might try to PAUSE, then FLUSH a stream or two with the EXEC CX PAUSE n and CX FLUSH n commands. This may clear up the deadlock.

If the deadlock persists, and/or the system console is not accepting input, you must force a shutdown and bring up AOS again. (But if you want to report this deadlock to DG in a Software Trouble Report (STR), there are additional steps; for these, see the STR section near the end of Chapter 10). To simply force a shutdown, follow these steps:

- Unlock the computer using the CPU LOCK switch (if any).
- For machines with programmed consoles, type the break sequence on the system console: CMD and BREAK keys, BRK key, or BREAK key. (If the software FLAGS lock is set on an MV/8000, press the CPU RESET switch to CONSOLE.)

  ! (or *SCP-CLI>*)

- Start the CPU at location 14 octal (the address of the Emergency Shutdown routine) as follows:

  On any 16-bit machine with programmed console: Type 14R (If R does not do an I/O reset on your computer, then you must type I before the 14R.)

  With hardware data switches: Set switches to 000014 (switches 12 and 13 up, the rest down). Press the STOP switch. Lift and release the RESET switch and then lift and release the START switch.

  On MV/8000, type RESET ⏎. Then, type START 14 ⏎.

  This should abort processing and start a shutdown. (If nothing happens, type the break sequence and try the previous step again.

  The system console says

  AOS PROCESSING ABORTED
  STRIKE "D" FOR AOS DUMP, "S" FOR SHUTDOWN, OR "R" FOR
  OPEN FILE REPORT
  ?

- To shut down, press the S key; and the Emergency Shutdown routine will run as described in "Running ESD," below. To do a Memory Dump, press the D key and proceed as described under "Doing a Memory Dump", below.

Panics

A panic (crash) results from an error that AOS recognizes but cannot rectify. It may involve hardware or software. On a panic, the system console displays the message

FATAL AOS ERROR - xxxyyy

value1 value2 value3 value4 value5 value6 value7 value8

AOS SYSTEM DUMP *LOAD TAPE ON MTA0* *STRIKE D WHEN READY:*	or	*AOS SYSTEM DUMP* *LOAD DISKETTE FOR DUMPING* *STRIKE ANY KEY WHEN READY*

xxx identifies the panic category with an octal code in bits 0 through 6.

yyy identifies the particular type of panic with an octal code in bits 7 through 15.

value1-8 identifies registers and important locations; note them if you want help from DG with this panic.

File :UTIL:AOS.PANICS.SR tells you how to interpret the panic values. You should keep a printout of the *current revision* of this file near the system console. The panic codes appear after the first few pages in the file (after the .DUSR definitions).

Generally, you should log each panic in a system log book kept near the system console. Note the time, revision of AOS system, any unusual conditions (like new software or hardware) that may have helped cause the panic, and the panic values. The written record is especially important if your system console is a CRT. Panic records can be very important to DG personnel whom you may call on for assistance.

The panic may have originated in hardware, as described in file :UTIL:AOS.PANICS.SR and in the next section. In any case, try to do a Memory Dump and run ESD, described below.

Hardware Errors

AOS relies on hardware to run. Inconsistencies in the CPU (including components like the map, buses, or multibit memory errors), disk, or other devices (like MCAs or DCUs) can cause it to panic. If you suspect a CPU error, or if a panic message indicates one, call your DG support organization. Some computers (like the S/280 with power supply error lights and MV/8000 with power-up checking and SCP) can diagnose CPU problems; if so, check the appropriate table in Appendix A.

If abnormal shutdowns recur, you should run the system log (SYSLOG) and, on MV/8000, ERRLOG (described in the SCP manual).

Disk and Tape Errors

A hardware error on disk from which AOS cannot recover, either with or without retries, is a *hard error*. An error from which AOS can recover is a *soft error*. If AOS encounters a hard error, it may panic. In this case, it will *not* write an error message to the system console. However, if AOS encounters either a hard error that does not cause a panic, or a soft error, it will write an error message to the system console. Then, it will continue.

The error message is

$$\left\{\begin{matrix} \mathit{SOFT} \\ \mathit{HARD} \end{matrix}\right\} \quad \mathit{ERROR,\ DEVICE\ nn,\ UNIT\ n,\ RETRIES\ r,\ STATUSES\colon s,\ s,\ s,\ s}$$

nn is the device code; e.g., 67.

n is the unit number; e.g., 0.

r is the number of retries AOS made before it gave up and signaled an error. Usually, it will retry 15 times. But on certain errors (for example, if a disk goes off line), it can't retry at all.

s is the hardware status word, described in the *Peripherals* manual named in the Preface. The system can display up to 4 status words. The code may indicate a bad sector (bad disk block), ECC, or other error.

If AOS is still running, and the disk is nonessential (a nonmaster LDU), try to ready and write-enable the disk. Then, try to RELEASE the disk, using its LDU name as assigned by the Disk Formatter. (The RELEASE will fail if the system cannot write to the disk.

If AOS is still running and the disk *is* essential, warn users to log off immediately, get back to the master CLI, and shut the system down with BYE) as shown in "Normal Shutdown". If AOS panics, note the panic code and try to run ESD.

When AOS is down, check the disk for obvious problems. For example, it may have gone off line; you can correct this with disk switches. If you fix the problem, and ESD succeeds, you can warm start AOS. If the status code *s* indicates a bad sector (bad block), or if you suspect a new bad block, run a Disk Formatter Partial format on the LDU, changing nothing; and specify read-only surface analysis. If the Formatter finds one or several new bad blocks, answer Y) when it asks *UPDATE BAD BLOCK TABLE*. You may then need to run FIXUP on the LDU. Identifying new bad blocks via the Formatter will correct the problem and make the LDU usable as usual. But if the Formatter finds *many* new bad blocks (say 20 or more), do not let it update the bad block table; the problem may well be head alignment. Turn disk power off and call your DG support organization.

If you cannot identify and solve the problem, turn disk power off and call your DG support organization. If possible, run without the disk until it is fixed.

If a hard error occurs on a tape unit, AOS will usually stay up. Try another tape. If the hard error recurs, call your DG support organization, and run without the tape unit until it is fixed.

While AOS is running, if system logging is on, it will try to record hardware errors in the system log file, SYSLOG. Using SYSLOG is described in Chapter 9, "Other Runtime Tools."

Doing a Memory Dump

AOS always prompts for a Memory Dump after a panic. This dump is required if you want to submit an STR. (An STR requires more than a Memory Dump. If you want to submit an STR, see Chapter 10. You *can* skip the Memory Dump by forcing a shutdown described in "Forcing ESD", below. Skip the Memory Dump only if you know that you don't need the Memory Dump information.)

To do a Memory Dump, follow these steps:

- For tapes, get a scratch tape, 800 feet or more, with write-enable ring in. Mount the tape on unit 0 on the first controller. If the unit has a density switch, choose DENSITY HIGH.

  For diskette, get a diskette (which need not have been formatted with the Disk Formatter) and make sure it is write-enabled. Insert the diskette in the primary unit.

- The system console is displaying

  AOS SYSTEM DUMP
  LOAD ...

  STRIKE ... WHEN READY:

  Strike the D key.

- The memory dump routine now copies main memory to tape or diskette. If it prompts for another diskette, remove the full diskette, insert another, and press a key. When done, the routine says

  AOS SYSTEM DUMP COMPLETED
  STRIKE "S" FOR AOS SHUTDOWN, OTHER KEY TO HALT
  ?

Dismount the tape or diskette, — and, if you want to save the dump, label the tape or diskette(s), and store in a safe place. Continue to the next section.

If the Memory Dump routine hits an error, the message *FATAL ERROR* appears on your system console. Mount a new tape or diskette. Then, type CONTINUE ↵ (MV/8000), P (other programmed consoles), or press the CONT switch (data switches). If this fails, notify your DG support organization.

Running ESD

Now you should run ESD. ESD is a routine that tries to restart AOS and force a normal shutdown, by writing system buffers to disk and closing open files. ESD is not perfect: it can't cope with certain system errors, and it can't verify the accuracy of all system databases that the panic may have affected. But it offers a good way to handle panics.

To run ESD,

- Simply press the S key in response to the Memory Dump or panic prompt.

  If you took a Memory Dump, it says, and you respond

  STRIKE "S" FOR AOS SHUTDOWN, OTHER KEY TO HALT
  ?   S

  If you didn't do a Memory Dump, but you want to run ESD, it says, and you respond

  AOS PROCESSING ABORTED.
  STRIKE "D" FOR AOS DUMP, "S" FOR SHUTDOWN, OR "R" FOR OPEN
  FILE REPORT.   S

  If you didn't do a Memory Dump, it says also

  DUMPS MUST BE TAKEN BEFORE ESD/REPORT. DO YOU WISH TO
  TAKE A DUMP? [N]

  This message warns you that a Memory Dump must be taken before ESD or Open File report. If not, ESD or REPORT will overwrite material needed for the dump. To do a dump, proceed as described earlier. To skip the dump, press ↵.

- Then ESD takes over:

  FILE SYSTEM RESTART
  NOW RESTARTING DEVICE nn UNIT n

  FLUSHING BUFFERS

  OPEN FILE PROCESSING

  SYSTEM SHUTDOWN

The main processor is halted. You can bring up AOS as described under "Warm Start" above.

If AOS comes up, but you cannot initialize any nonmaster LDUs that were initialized when the error occurred, then ESD was only partially successful. It closed the master LDU, but not the other(s). To handle this, execute either stand-among FIXUP from AOS and run it on the inaccessible LDU(s), or shut down and run stand-alone FIXUP on the inaccessible LDU(s).

If AOS will not come up after ESD runs, get an Open File Report, then run stand-alone FIXUP on all LDUs that have open files.

If ESD fails, it issues either a FATAL ERROR message of its own or an ABNORMAL SHUTDOWN message. Take a Memory Dump and submit it with its own Software Trouble Report. When you submit the Software Trouble Report, be sure to list the previous panic. Then, type S (if you get the chance) or force ESD.

If ESD fails again, it cannot deal with the error. Run Open File Report, then reset the CPU, boot, and run FIXUP on all LDUs that have open files.

Forcing ESD

How you force ESD depends on what type of computer you have:

- On any 16-bit machine with programmed console: Type the break sequence: CMD and BREAK keys, or BRK key, or BREAK key, depending on the type of system console. Then type I and 14R (on an S/140 you can skip I).
- With hardware data switches: Set switches to 000014 (switches 12 and 13 up). Press the STOP switch, lift the RESET switch, and then lift the START switch.
- On an MV/8000, type the break sequence, then type RESET↵ and START 14↵.

The system displays

AOS PROCESSING ABORTED
STRIKE "D" FOR AOS DUMP, "S" FOR SHUTDOWN, OR "R" FOR OPEN FILE REPORT
?

Normally, you should type S or D to answer. If ESD succeeds, you do not need an open file report.

Open File Report

An open file report can save time if your system has one or more nonmaster LDUs initialized when it fails. It saves no time if only one LDU (the master LDU) is running. So you should read this section only if more than one LDU was initialized when AOS came down.

If ESD panics or displays an *ABNORMAL SYSTEM SHUTDOWN* message, this means that ESD could not close files on one or more disks. However, if the *ABNORMAL SYSTEM SHUTDOWN* message displays when you try to shut down the system normally, then the system could not close all of the open files on one or more disks. In either case, you must run FIXUP on these disks to close the files and finish the shutdown. The Open File Report option describes exactly *which* disks require FIXUP. If the Report says that no file is open on all disks in an LDU, you need not run FIXUP on that LDU. In a multiple-LDU system, this can shorten the FIXUP run significantly.

The best time to specify an Open File Report is after ESD has failed. (You don't need it if ESD succeeds.) You can take the routine whenever you see the prompt

STRIKE "D" FOR AOS DUMP, "S" FOR SHUTDOWN, OR "R" FOR OPEN FILE REPORT
?

After running ESD, you can get this prompt by typing CONTINUE↵ or by pressing the CONTINUE switch.

To get a report, type R. (If you have not taken a Memory Dump or attempted an ESD shutdown, you will now get a chance to take a Memory Dump, but can proceed by pressing ↵.)

The report will then be displayed on the system; in the form

OPEN FILE REPORT

FILES OPEN ON THE FOLLOWING UNITS:

DEVICE CODE	*UNIT #*	*OPEN COUNT*
device-code	*unit*	*number-of-open-files*
device-code	*unit*	*number-of-open-files*

The Open File Report lists the device code and unit number of the first unit in each LDU on which you must run FIXUP. Although you must run FIXUP on every disk in each LDU that the Open File Report lists, you do not need to run FIXUP on any other LDUs.

The following example shows a panic — Dump — failed ESD — report sequence for a two-LDU system. The master LDU is a single-disk LDU; the other LDU is a two-disk LDU, including units DPF10 and DPF11.

FATAL AOS ERROR: ....

...

AOS SYSTEM DUMP.
LOAD TAPE ON MTA0.
STRIKE "D" WHEN READY: D

Mount tape on unit 0, and strike D to do a Memory Dump.

STRIKE "S" FOR SHUTDOWN, OTHER KEY TO HALT
? S

FILE SYSTEM RESTART

NOW RESTARTING DEVICE 27 UNIT 0

NOW RESTARTING DEVICE 67 UNIT 0

NOW RESTARTING DEVICE 67 UNIT 1

FLUSHING BUFFERS

OPEN FILE PROCESSING

FATAL AOS ERROR ...

.
.
.

AOS SYSTEM DUMP.
LOAD TAPE ON MTA0.
STRIKE "D" WHEN READY: D

Either type D to take the Memory Dump of the new panic and force ESD, or you can just force ESD. (See "Forcing ESD.") When you force ESD, the message is:

AOS PROCESSING ABORTED
STRIKE "D" FOR AOS DUMP, "S" FOR SHUTDOWN, OR "R" FOR OPEN FILE REPORT
? R
OPEN FILE REPORT

FILES OPEN ON THE FOLLOWING UNITS:

DEVICE CODE	*UNIT #*	*OPEN COUNT*
000027	*0*	*000002*

Here, the operator needs to run FIXUP on only the LDU on device code 27, the master LDU.

Device code 67, unit 0, is not listed in the Open File Report, because there are no open files on the LDU for which DPF10 is the first disk. The Open File Report would not mention device code 67, unit 1, even if the multiple-disk LDU consisting of units DPF10 and DPF11 had open files, because the Open File Report lists a multiple-disk LDU by only the device code and unit number of the first disk in the LDU.

The FIXUP Disk Fixer

Abnormal shutdown leaves the master LDU (and other initialized LDUs, if any) in an unpredictable state, with open files that may not have been updated. ESD, if it succeeds, writes system buffers to update open files, closes the files, and restores disk integrity.

But if ESD fails, then you must run FIXUP to fix the pertinent LDU(s). If an LDU does not need fixing, FIXUP will tell you that fixing is not necessary, and allow you to skip the fix. You may want to do the fix anyhow if you suspect errors in the LDU.

Even if you are not forced to run FIXUP, you might run it periodically to clean up and verify the file structure on your LDUs.

A *hard error* on an LDU may not cause abnormal shutdown. But a hard error often means that part of the LDU is inaccessible — perhaps with an unreadable bad block in the middle of a file. After a hard error, you should run a Disk Formatter Partial format on the LDU(s) to check for new bad blocks and enter any in the bad block table. Then, if the Formatter gives a *MUST RUN FIXUP* message, you must run FIXUP to correct the file structure. FIXUP may be able to rebuild — thus save — part of the pertinent file.

You can run FIXUP as a stand-alone program or as a stand-among program under AOS control. Stand-among FIXUP works only for nonmaster LDUs. To start stand-among FIXUP, type

```
) XEQ    :UTIL:FIXUP ↵
```

Stand-alone FIXUP allows you fix any LDU, including the master LDU. Stand-among FIXUP allows you to run AOS while fixing LDUs, although it takes longer to run than it would take to run stand-alone FIXUP on the same LDU. To fix the master LDU, you must use the stand-alone FIXUP.

If You Make a Mistake

If you make a typing mistake, before pressing NEW LINE, press the DEL key to erase characters one-by-one; or enter CTRL-U to erase the entire line.

If you have already pressed NEW LINE to enter an incorrect answer, FIXUP may recognize your error and repeat the question. If so, type the desired answer.

If you decide to abort stand-alone FIXUP, type the break sequence and RESET ↵, then reboot FIXUP. To abort and restart stand-among FIXUP, enter CTRL-C CTRL-B and type XEQ :UTIL:FIXUP ↵.

On a CRT display console, you can suspend display by typing CTRL-S and resume display by typing CTRL-Q.

Running FIXUP

You can boot and run either FIXUP directly from disk. (Stand-alone FIXUP is also in file 1 of your AOS system tape and on system diskette 1 — if for some reason it won't boot from disk.) The console you use must be an upper- and lowercase console, because FIXUP console dialog is upper- and lowercase.

To bootstrap and run FIXUP, follow these steps.

1. Make sure that all the disks you want to fix are mounted in their units (if removable), write-enabled, and ready.

   With AOS running (stand-among FIXUP), type XEQ :UTIL:FIXUP ↵ next to the CLI prompt; and go to step 4.

2. Start the system (a warm start if power has stayed on to the CPU; otherwise a cold start). When you program load from the master LDU, the system console says

****WARNING*** DISK IS IN USE - FIXUP MUST BE RUN ON LDU*
SYSTEM PATHNAME?

(If there is no ****WARNING**** message, then the master LDU was closed normally and you need not run FIXUP on it.)

3. The system console is asking for a pathname. Reply with FIXUP:

SYSTEM PATHNAME? FIXUP ↵

4. The previous steps read FIXUP into memory. It says

AOS Disk Fixer Rev. n

Verbosity [1] ?

5. FIXUP can write messages to a log file, covered in the next question. Your answer to this question sets the amount of detail in each message.

0 ↵	tells FIXUP not to log messages.
↵ or 1 ↵	tells FIXUP to log the message and file pathname for each file that it deletes, rebuilds, renames, closes, or finds incomplete. If you choose certain options later in the dialog, FIXUP will also log messages that pertain to these options. This is a good general-purpose answer.
2 ↵	tells FIXUP to log the following for each disk error: message, file pathname, LDU unique ID, file address, index level, disk unit, and logical and physical disk addresses of the error. FIXUP reports all multiply-allocated blocks, invalid pointers, and empty chain anchor blocks.
3 ↵	tells FIXUP to log everything it does: all actions included in verbosity 2, plus replacement of directory bitmaps, updates of file lengths, and deallocation of empty directory blocks, and rebuilding FNB and FIB chains.

A verbosity of 0 produces the fastest fix, but yields no error information; so you should answer 0 ↵ only when you don't care about error information. If you answer 0 ↵, go to step 8. Verbosity 1 gives user-oriented messages, on which you can act. In most cases, you will want verbosity 1 and a line printer log.

Verbosities 2 and 3 are intended for DG personnel. If you want to submit a Software Trouble Report to DG, please specify 3 ↵.

If you choose a verbosity other than 0 ↵, FIXUP asks

Error log file [console] ?

6. This determines the error log file. For stand-alone FIXUP, the most common log file is the line printer (device names LPA, LPA1, LPB, LPB1, LPC, LPC1, LPD, or LPD1). A slower option is the console (default) if this is a hardcopy device. For stand-among FIXUP, you can specify @CONSOLE, @LPT, or a disk file (which FIXUP will create if it doesn't exist or append to if it does exist). Generally, answer

LPB ↵ (or @LPT ↵)

If you specified a line-printer device for the logfile, stand-alone FIXUP will ask:

Device code [default] ?

Unless you know that the line printer is connected to a nonstandard device code, press ↵ to select the default.

7. *Should I report closing files and deleting transients [n]?*

Reporting on each open and transient file adds time to the fix — especially if the log file is a hardcopy console. (Transients are files that would normally have been deleted by other programs; FIXUP normally deletes them in the course of cleaning up.) The "closing" and "deleting" information is not usually needed for diagnostic purposes. If you want FIXUP to report open files and deleting transients, type Y ↵. But unless you really need to know which files were open, press

* ↵

8. *May I fix it [n] ?*

This question determines whether FIXUP will fix the LDU(s) or simply report errors. Unless you type Y ↵, the LDU(s) will remain inaccessible to AOS. If you *do* type Y ↵, FIXUP will act to correct the LDU; a disk integrity problem may involve the loss of files.

If you answer ↵ (No), FIXUP will not write to the LDU. You will see the same messages as if you said Yes, except that FIXUP will not try to rebuild files and won't display *FILE REBUILT* messages. You might answer ↵ if you suspected disk controller hardware problems, and didn't want your LDU(s) fixed on the basis of hardware problems. If you answer ↵, FIXUP will say *DONE, BUT NOT FIXED!* after it runs on the LDU.

Generally, answer

Y ↵

9. *Would you like to select options [n] ?*

* Options include: confirmation if fixing will cause deletions; deleting temporary files, cancellation of queued requests on serious errors, and output to the console in addition to the log file.

If you decline options, and have told FIXUP to fix it, FIXUP will correct the LDU even if fixing will cause deletions; it will delete temporary files (usually a productive thing to do); it will shrink directories; it will proceed with queued requests regardless of errors; and it will send messages to the log file only (but abort messages will go to the console in any case).

Options add time and steps to the fix; but under some circumstances, you might want to select an option like "cancel queueing on serious errors". Any option you choose will be carried over into the next request, if you queue multiple fix requests.

If you don't want to select any options, press ↵ and go to step 10.

To select one or more options, answer Y ↵. FIXUP then asks one or more of the following questions.

9a. *Confirm fixing if an error will cause deletions [n] ?*

FIXUP asks this only if you chose verbosity 2 or 3 and said Y ↵ to *May I fix it?*. If you answer Y ↵, when FIXUP finds an error where fixing would cause a deletion, it reports the error, stops, and asks

May I continue fixing [n]?

To have FIXUP continue, which will mean the loss of at least part of a file and may mean the loss of multiple files, you will need to type Y ↵. FIXUP will then delete something and continue fixing until it finds another error that requires deletion to fix.

If you answer Y ↵ to this *Confirm fixing...* question, FIXUP will send messages to the console as well as the log file.

If you choose to say no by pressing ⤶, FIXUP will stop fixing the LDU and proceed as if you had said No to *May I fix it?*. The LDU will remain inaccessible to AOS. *

9b. *Delete ?+.TMP files [y] ?*

Temporary files, whose names have the form "?name.TMP", are created and used by DG utility programs; the utility programs delete them after completing their jobs. But if a utility is interrupted before finishing, its temporary files remain on the LDU.

Normally, unless you know that the LDU has important files of the form ?name.TMP, answer yes by pressing ⤶. If you want to keep all ?+.TMP files, type N ⤶.

9c. *Should I cancel queueing on serious errors? [n]*

This is asked only if you told FIXUP to fix it.

By default, FIXUP continues fixing queued LDU requests, regardless of errors on the current LDU, even if it aborts on the current LDU. If you answer Y ⤶, FIXUP will fix the LDUs in sequence; and if it finds a serious error on one LDU, it will finish fixing that LDU, then ask

May I continue fixing [n]

If you see this question, you will know that the current LDU has been fixed (unless FIXUP aborted). If you say no by pressing ⤶, FIXUP will cancel the remaining requests and terminate; the remaining LDUs will remain inaccessible to AOS. If you type Y ⤶, FIXUP will continue with the next queued request; if it encounters another serious error, it will ask for confirmation again as above.

A "serious" error is any read/write error (even if retries succeeded), deletion of any whole file or a multiply-allocated block, or an invalid pointer in an index block.

The option to cancel queueing on serious errors requires your presence at the console to confirm — eliminating the main advantage of queueing. But, under some circumstances, you may want to choose this option. If so, type Y ⤶. Generally, say no by pressing ⤶.

9d. *Send output to console in addition to logfile [n] ?*

FIXUP asks this only if you specified a log file other than the console and said no to *Confirm fixing....*

If you chose a log file other than the console, and want error messages to appear on the console as well as the log file, type Y ⤶. Otherwise, say no by pressing ⤶.

10. *Specify each disk unit in the Logical Disk*
Enter disk unit name :

Now, you need to identify each disk in the LDU. The order in which you give the unit names doesn't matter, but FIXUP will insist that you specify each disk. With stand-alone FIXUP, for the master LDU, the disk unit name is often DPF0 or DPI0. With stand-among FIXUP, under AOS, the name(s) can be unit names (e.g., @DPF10); or, if the LDU was mounted under EXEC in a user MOUNT request, the linkname used in the MOUNT command. For example,

DPF0 ⤶ (or DPI0 ⤶)

11. *Device code [default] ?*

This is asked only for stand-alone FIXUP. Unless you know that the LDU is connected to a nonstandard device code, select the default by pressing

↵

FIXUP cycles the *UNIT NAME* and *DEVICE CODE* questions until you specify all disks in one LDU.

12. FIXUP now checks the LDU you identified in steps 10 and 11. If the LDU does not *need* fixing, FIXUP says

*** *This LDU was not in use, fixing is not necessary* ***

This means that the LDU was closed normally by AOS, and is accessible to AOS; you need not run FIXUP to use the LDU. However, there may be other errors that FIXUP will find if allowed to fix the LDU.

13. Then, for each LDU, FIXUP asks

Would you like to cancel this request [n]

To cancel this request (perhaps because FIXUP said fixing wasn't needed, or for any reason), type Y ↵. To have FIXUP run on the LDU, press

↵

14. FIXUP asks

Would you like to queue up another request [n] ?

FIXUP allows you to fix as many as 16 LDUs, using the same verbosity, logfile, and options for each LDU. If you want to specify different verbosities, logfiles, or options for different LDUs, you must run FIXUP multiple times.

If you want to specify another LDU fix, answer Y ↵; FIXUP then asks about the LDU as in steps 10-13. If you have specified all the LDUs you want, press ↵.

15. After you confirm the last request, FIXUP runs on the LDU(s). For an average 190-megabyte LDU, a typical fix takes about 10 minutes. You need not stay at the console, unless you chose options that require confirmation. When FIXUP has finished each LDU, it says

DONE! (or *DONE, BUT NOT FIXED!* if you told FIXUP not to fix the LDU)

When FIXUP has run on the last LDU specified, control returns to either the SCP-CLI> prompt (MV/8000) or the ! prompt (other programmed consoles) or — for stand-among FIXUP — the CLI.

To rerun stand-alone FIXUP, type CONTINUE ↵ MV/8000, P (other programmed consoles), or press the CONTINUE switch — or, for stand-among FIXUP, XEQ :UTIL:FIXUP ↵ — and return to step 4. Otherwise, you're done with FIXUP. You can warm start AOS and/or continue with system operations.

If FIXUP hits a disk error that it can't correct, it aborts. Make sure the disk is write enabled and try again. If the abort recurs, the cause is usually hardware. See the *DISK ERROR* message in Table 6-1.

If you suspect disk alignment problems or surface damage, *don't put any other packs in the unit* (if this applies). FIXUP verbosity 1 error messages, and all abort error messages, are shown in Table 6-1.

FIXUP Examples

A FIXUP example with the simplest dialog follows in Figure 6-6. A FIXUP example with multiple requests and options follows in Figure 6-7.

(Bootstrap FIXUP or XEQ :UTIL:FIXUP ⤶.)

AOS Disk Fixer, Rev 7.00

Verbosity [1] ? ⤶
Error logfile [console] ? LPB ⤶ (@LPT ⤶ for stand-among FIXUP)
Device code [17]? ⤶ (For stand-alone FIXUP)
Should I report closing files and deleting transients [n] ? ⤶
May I fix it [n] ? Y ⤶
Would you like to select any options [n] ? ⤶
Enter disk unit name: DPF0 ⤶ (@DPF0 ⤶ for stand-among FIXUP)
Device code [27] ? ⤶ (Omitted for stand-among FIXUP)

Would you like to cancel this request [n] ? ⤶

Would you like to queue up another request [n]? ⤶

.
. (time passes as FIXUP fixes LDU)
.

DONE!

The line printer listing from this FIXUP session might show something like this:

REQUEST 1 (DPF0), FIXING LDU 'ROOT1' NOW...

REPAIR IN FILE :XDIR:TEST -- PART OF FILE MAY BE MISSING
DONE!

DG-26902T

Figure 6-6. Simple FIXUP Dialog Example

(Bootstrap FIXUP or XEQ :UTIL:FIXUP ⤶

AOS Disk Fixer, Rev 7.00

Verbosity [1] ? ⤶
Error logfile [console] ? LPB ⤶ (@LPT ⤶ for stand-among FIXUP)
Device code [17]? ⤶ (Omitted for stand-among FIXUP)
Should I report closing files and deleting transients [n] ? ⤶
May I fix it [n] ? Y ⤶
Would you like to select any options [n] ? Y ⤶

Confirm fixing if an error will cause deletion(s) [n] ? ⤶
Delete ?+.TMP files [y] ? ⤶
Should I cancel queueing on serious errors [n] ? Y ⤶
Send output to console in addition to logfile [n] ? Y ⤶

--- REQUEST 1 ---

DG-26903T

Figure 6-7. Multiple-Request FIXUP Dialog, with Options (continues)

Specify each disk unit in the Logical Disk

Enter disk unit name: DPF0 ↲ (@DPF0 ↲ for stand-among FIXUP)
Device code [27] ? ↲ (Omitted for stand-among FIXUP)

Would you like to cancel this request [n] ? ↲
Would you like to queue up another request [n]? Y ↲
--- REQUEST 2 ---

Enter disk unit name: DPF21 ↲ (@DPF21 ↲ for stand-among FIXUP)
Device code [no default] ? 44 ↲ (Omitted for stand-among FIXUP)

**** This LDU was not in use, fixing is not necessary ****

Would you like to cancel this request [n] ? Y ↲
--- REQUEST 2 CANCELLED ---

Would you like to queue up another request [n] ? Y ↲

-- REQUEST 2 --

Specify each disk unit in the Logical Disk
Enter disk unit name: DPF11 ↲ (@DPF11 ↲ for stand-among FIXUP)
Device code [67] ? ↲ (Omitted for stand-among FIXUP)

Would you like to cancel this request [n] ? ↲
Would you like to queue up another request [n] ? ↲

--- 2 REQUESTS QUEUED ---

REQUEST 1 (DPF0), FIXING LDU 'ROOT1' NOW...

REPAIR IN FILE :XDIR:TEST -- PART OF FILE MAY BE MISSING

DONE!

**** QUEUEING CANCELLED - FIXING OF REMAINING*

*REQUESTS MUST BE CONFIRMED **** (FIXUP says this because the "Cancel queueing" option was chosen and a serious error occurred.)

May I continue fixing [n] ? Y ↲

REQUEST 2 (DPF11), FIXING LDU 'STRS' NOW...
DONE!

The line printer listing from this FIXUP session might show something like this:

--- 2 REQUESTS QUEUED ---

REQUEST 1 (DPF0), FIXING LDU 'ROOT1' NOW...

REPAIR IN FILE :XDIR:TEST -- PART OF FILE MAY BE MISSING DONE!

**** QUEUEING CANCELLED - FIXING OF REMAINING*
*REQUESTS MUST BE CONFIRMED ****

REQUEST 2 (DPF11), FIXING LDU 'STRS' NOW...
DONE!

DG-26903T

Figure 6-7. Multiple-Request FIXUP Dialog, with Options (concluded)

What FIXUP Does

If you tell FIXUP to "fix it," it performs the following tasks on an LDU.

- Closes all open files.
- Checks for multiple-allocation of disk blocks. For each multiply-allocated block, FIXUP deletes all but the first use of the block. Only the block itself (or the file element, if the block is part of a data file) is deleted.
- Checks that each directory data block and each index element is correct; and corrects it if needed.
- Checks that the directory data blocks in each directory are correctly linked; and rebuilds directory data block chains if needed;
- Shrinks directory files if it finds unused blocks in the directory file.
- Deletes temporary files in every directory. Temporary files are used by system utilities and are normally deleted when the utilities finish their jobs. Their names have the form ?name.TMP
- Deletes all "transient" files from the peripherals directory (:PER or @), if :PER is on the LDU. These include device entry files, user tape volumes (if any), names of nonmaster LDUs initialized (if any), and IPC files created by other DG software (if any). The next AOS system will recreate the device entry files when it comes up. The tape volume files represented outstanding user tape MOUNT requests: users will need to reissue any outstanding tape MOUNT requests when EXEC is running. The nonmaster LDUs will need to be initialized (INITIALIZE command) when AOS is running; this is often done in the UP macro.
- Recomputes the count of subordinate directories for every directory, the length of every file, and the current size for every control point directory.
- Builds an updated bit map for the LDU.

All this action cleans up the file structure and frees disk space. (When AOS is running, you can use the CLI command SPACE :) to check space in the master LDU.)

Interpreting FIXUP Messages

FIXUP verbosity 1 messages are designed for user action; they are described in Table 6-1.

FIXUP verbosity 2 and 3 messages concern directory-file structures: chains, disk addresses, multiply allocated blocks, and invalid or incorrect file block entries (e.g., FIB, FNB). These messages are really intended for DG personnel; you can't do anything about them, so they are not included in Table 6-1.

Table 6-1. FIXUP Error Messages, Verbosity 1

Message	Description and Action
ABORT message	This FIXUP abort message precedes one of several other messages shown in this table. Find "message" and take the action described.
ACL DELETED (FILE filename)	The access control list (ACL) information for this file has been deleted because FIXUP found inconsistencies. FIXUP may rebuild the file, with a *FILE REBUILT (FILE filename)* message, with a new ACL. If FIXUP doesn't rebuild the file, it will be left with a null ACL. Whether or not FIXUP rebuilds the file, you can assign the desired ACL later, from AOS, if you want.
CAN'T DELETE ROOT DIRECTORY, ...	The first index block for the root directory contains invalid addresses. FIXUP will not delete this block, because doing so would effectively destroy the LDU directory structure. Run FIXUP again. If this message recurs, then this LDU cannot be fixed *in this unit*. There may be alignment or other hardware problems. DTOS hardware diagnostics are needed, on a scratch pack if possible. If there is a correctable hardware problem, the LDU file structure may be intact. If diagnostics show the hardware is okay, the LDU can't be fixed. It may need hardware formatting; in any case, a Disk Formatter Full format must be run on the pertinent disk(s) to recreate the LDU.
DEVICE ALREADY IN USE	The disk unit you specified is in use by AOS. Abort FIXUP and release the unit (command RELEASE ldu-name ↵); or, for the master LDU, shut down AOS. Then, if you are running FIXUP after a hard error, run a Disk Formatter Partial format on the LDU, and update the bad block table. Then run FIXUP again.
DISK ERROR, DEVICE d STATUS s, RETRIES r, LDU ID = id, PHYSICAL ADDRESS= n	FIXUP encountered a disk error. It retries up to 20 times (24 octal) before aborting. You might want to note the disk status code *s* for later reference in the Peripherals manual (see Preface). Even if FIXUP can correct the error and continue, there may be a potential bad block at physical address *n;* you may want to run a Disk Formatter Partial format later to check. (Numbers *s, r,* and *n* are octal.) If FIXUP cannot correct the error in 20 retries, it aborts. Make sure the disk unit is write enabled; if not, write enable the unit and rerun FIXUP. If the unit *is* write enabled, see the *FATAL DISK ERROR* message, next, for recovery action.
FATAL DISK ERROR	If this message follows a "DISK ERROR" message, it means that FIXUP hit a new bad block (or that the disk unit was not write enabled). If not write enabled, write enable the unit and run FIXUP again. Otherwise, run a Disk Formatter Partial format, with read-only surface analysis, on the LDU. If the Formatter finds a new bad block, answer Y ↵ to the *UPDATE BAD TABLE* question. Then, run FIXUP again. If this *FATAL* message does NOT follow a *DISK ERROR* message, it means that FIXUP couldn't read the same block twice. This means hardware problems in the disk controller or unit. Try FIXUP again. If it fails the same way, DTOS diagnostics may be needed on the disk unit (with a scratch pack in the unit, if possible); contact your DG support organization.

(continues)

Table 6-1. **FIXUP Error Messages, Verbosity 1**

Message	Description and Action
FILE file IS NOT A DISK UNIT	The name you entered is not a disk unit name. Respecify.
FILE ACCESS DENIED	You do not have write access to the disk unit you specified. Abort FIXUP, turn SUPERUSER on, and try again.
FILE CLOSED (FILE pathname)	FIXUP found this file open and has closed it.
FILE DELETED (FILE filename)	FIXUP deleted this file. Possible reasons: the File Information Block data was inconsistent; "delete on close" was specified when the file was opened; the file was a link entry with bad directory information; or a data block in the file's parent directory was unreadable. If you want the file, load it from backup media when AOS is running.
FILE DOES NOT EXIST	The file does not exist. Try adding the @ prefix to the devicename. Perhaps the unit was not identified to AOSGEN (stand-among FIXUP only).
FILE REBUILT (FILE filename)	FIXUP rebuilt this file, which means that FIXUP tried to rejoin the file with its File Name Block, ACL block, or other descriptor block. If *filename* is a normal AOS filename, then FIXUP gave the file its original name. But if the association between filename and file was bad, and FIXUP could not let the file keep its original name, then FIXUP assigned a *filename* of the form ?AAAAAAAAAC The next file rebuilt in this directory could be renamed ?AAAAAAAAAD, the next ?AAAAAAAAAE, and so on. If the FIXUP log shows one or more *FILE REBUILT (FILE ?AAAA...)* messages, you should check the files from AOS as follows. Get into the pertinent directory, and type) FILES/AS/S ?AAA- ↵ If only one ?AAA- filename appears, check the FIXUP log for an *INVALID FILENAME DELETED* message before the *FILE REBUILT* message, in the same directory. The invalid filename, if shown, is usually the original filename. If so, you can rename the ?AAAAAAAAAC file to the invalid filename. If more than one ?AAA- filename appears from the FILES command, check the FIXUP log for multiple *INVALID FILENAME DELETED* messages. The ?AAA- original filenames are usually the invalid filenames. You can rename the ?AAA- files to the invalid filenames, as appropriate. Don't worry if you see duplicate ?AAA- filenames after the FILES command; just rename each ?AAA- file until you have renamed them all. (FIXUP might have needed to create duplicate ?AAA- names, but renaming the files will make them accessible as usual.) Any file to which FIXUP assigns a new name retains its original type, creation date, and creation time. Lastly, a file mentioned in a *FILE REBUILT* message may be missing its ACL or User Data Area. Check and correct if needed from AOS.

(continued)

Table 6-1. FIXUP Error Messages, Verbosity 1

Message	Description and Action
FILE(S) MAY BE MISSING	FIXUP found one or more multiply-allocated blocks in this directory, and deleted these; so filenames that FIXUP cannot know about may have been deleted. When AOS is up, check for missing files in this directory (perhaps using the last dump listing); load the missing file(s) from backup media.
FIXUP CHECKSUM ERROR	Microcode is bad or the FIXUP disk file is unstable. Try reloading microcode via a cold start. If the error recurs, you must load FIXUP from tape. Get the AOS system tape, program load from it (code 22), and choose tape file 1↵. For diskette, go to Chapter 3 and execute steps 2 through 9; type FIXUP↵; and execute step 27 if needed. Then run FIXUP on the LDU.
FIXUP INTERNAL ERROR -- message	Run FIXUP again. If this message recurs on an MV/8000, reload microcode (power off and on) and try again. On any machine, if FIXUP fails again, contact your DG support organization.
INCORRECT DISK FORMAT REVISION NUMBER OF n, FIXUP'S REVISION NUMBER IS 3	If *n* is not 3, this message means that the Disk Information Block (DIB) is bad, and this disk cannot be fixed. A Disk Formatter Full format is required to make the disk usable by AOS.
INCORRECT FORMAT FOR DISK INFORMATION BLOCK	The Disk Information Block (DIB) for one of the disks in the LDU is bad. A Disk Formatter Full format is required to make the disk usable by AOS.
INSUFFICIENT MEMORY FOR BIT MAP, FIXUP NEEDS nK WORDS MORE.	There is not enough main memory for FIXUP to rebuild the LDU bit map. Retry. If this message recurs, call your DG support organization.
INVALID FILE NAME DELETED (FILE filename)	FIXUP found inconsistencies in the filename-file structure; and it has dissociated the filename from the file. The filename is stored outside the file, so the file itself has *not* been deleted. Later, within this directory, FIXUP will either reassign *filename* to the original file, or it will assign a filename of the form ?AAAAAAAAAC to the file. If the latter, this message will be followed by a *FILE REBUILT* message. Proceed as described under the *FILE REBUILT* message, above.
INVALID FIRST LOGICAL ADDRESS, FILE EMPTIED	The starting logical address for this file is too great for the LDU. FIXUP zeros the logical address, effectively emptying the file. Later, from AOS, load the file (or the contents of the directory, if the file named in the message is a directory) from backup media.

(continued)

Table 6-1. FIXUP Error Messages, Verbosity 1

Message	Description and Action
NAME BLOCK ADDR. =n *ACL BLOCK ADDR. =n* *SYSBOOT ADDR. =n,..* *BITMAP AREA ADDR. =n* *OVERLAY AREA ADDR. =n* *unit REMAP ... ADDR.=n.* . . . . followed by one of the these messages: *The NAME BLOCK message* *The ACL BLOCK message* *SYSBOOT message* *The BITMAP area message* *The OVERLAY area message* *unit REMAP area message*	This abort sequence of messages means that one of the LDU disk's Disk Information Block (DIB) has bad information. (All *n* numbers are octal.) If one of the messages suggests moving the BITMAP, OVERLAY, or REMAP areas, note all addresses, then run a Disk Formatter Partial format and move the area to a free space on the LDU (or, for REMAP area, to a free space on the disk). Then run FIXUP again. If FIXUP succeeds, you should dump all LDU material and run a Disk Formatter Full format on the LDU; then reload the LDU material. If a Disk Formatter Partial format cannot move the offending area, then the LDU cannot be fixed; a Full format is needed. If the addresses and sizes that FIXUP reports seem reasonable, try FIXUP again. If it fails again on MV/8000, reload microcode (described under *FIXUP INTERNAL ERROR* message) and try FIXUP again. If it fails again, contact your DG support organization.
NEW SIZE = #n BLOCKS, *#n BLOCKS RECOVERED*	FIXUP has calculated a new size for the LDU. If you chose the option to shrink directories, FIXUP also reports the total number of blocks reclaimed on the LDU.
NO ROOM TO REBUILD FILE, *CHECK FOR A FILE MISSING*	FIXUP could not find a valid filename for this file, and tried to create a new filename. But FIXUP could not find an unused "slot" for the new name, so it had to delete the file. If FIXUP reported an *INVALID FILE NAME DELETED (FILE filename)* message for this directory, the invalid filename may be the file FIXUP deleted. Check from AOS; then load the missing file from backup media.
PART OF FILE MAY BE *MISSING*	FIXUP found one or more multiply-allocated file elements in this file, and has deleted them. From AOS, see if the file has the correct length and content. If it is not intact, load it from backup media.
QUEUEING CANCELLED - *FIXING OF REMAINING* *REQUESTS MUST BE* *CONFIRMED* *May I continue* *fixing [n] ?*	The last LDU you specified has been fixed, (unless FIXUP aborted); but FIXUP encountered a noteworthy error on it. If the log shows that only a few errors (like recoverable disk errors or one or two file deletes) occurred, type Y ↵ to have FIXUP fix the next LDU(s). If there are many serious errors, there may be disk controller hardware problems; and you may want to have FIXUP stop by pressing ↵.
RENAMED TO newfilename *(FILE oldfilename)*	The file named in *oldfilename* didn't hash correctly. FIXUP renamed it to *newfilename,* of the form ?AAAAAAAAC. You can rename the file to its original name later, from AOS. (If, from AOS, you see duplicate ?AAA- filenames in this directory, rename the FIXUP-renamed files as described in the *FILE REBUILT* message, above.)
REPAIR IN DIR pathname *-- message*	This message precedes one of several other messages shown in this table. Find *message* and take the action described.
REPAIR IN FILE pathname *-- message*	This message precedes one of several other messages shown in this table. Find *message* and take the action described.
The xxxx BLOCK ...	See the *NAME BLOCK* message.

(continued)

Table 6-1. FIXUP Error Messages, Verbosity 1

Message	Description and Action
TOO MANY DISKS IN USE	An LDU cannot include more than eight disks, but you have entered more than eight disk unit names, and FIXUP has verified that each disk has the same LDU ID. Retry FIXUP, making sure that you enter only the correct disk unit names for the LDU. If this message recurs, this LDU cannot be fixed; a Disk Formatter Full format is needed.
TRANSIENT FILE (FILE filename DELETED)	FIXUP has deleted transient file "filename". A transient file is a file that would have been deleted anyway in the course of normal system operations (further defined in section "What FIXUP Does").
USER DATA AREA DELETED FILE filename)	FIXUP found inconsistencies in this file's User Data Area (often used for printer formatting) and it has deleted the User Data Area. From AOS, you can either LOAD/DELETE the file from backup media or recreate a printer formatting User Data Area with the FCU utility (in directory :UTIL:FORMS).

(concluded)

Power Failures

Unless your computer has a backup battery, it will lose the contents of volatile memory on a power fail. On a brownout (serious power drop) or overtemperature condition, the computer may cut its own power. In either case, without battery backup, someone must run FIXUP on all initialized LDUs when full power returns.

With battery backup, AOS will try to restart all critical devices (like disks) when power returns; if this succeeds, AOS will continue running as before the power fail. For battery backup to work, all the following conditions must be true:

- The CPU must have a full backup battery;
- power must return before the battery is exhausted;
- battery backup must have been chosen at AOSGEN;
- the CPU LOCK switch (if any) must have been in the ON or LOCK position when power went down. This enables transfer to the battery.

Power Fail Recovery

When power returns, look at the system console.

- If the system console shows messages like

THERE HAS BEEN A POWERFAIL

NOW RESTARTING DEVICE n UNIT n

.

.

.

AUTO RESTART HAS COMPLETED

)   (Returns to AOS CLI prompt or prompt of other program running under CLI.)

This means that AOS has fully recovered. The power outage is recorded in the SYSLOG log file, if logging was on at the outage.

If AOS cannot restart a device, it will tell you so. If the device is a disk, you will probably need to run FIXUP on it.

Any power outage, regardless of recovery, has the following effects:

1. It eliminates vacuum to mag tape units; so if type MTA or MTB units were active on your system, someone must press BOT on them to recreate the vacuum, then press ON LINE.
2. It invalidates any tape write that was occurring when power went down; the write must be restarted from the beginning.
3. It takes line printers off line; and someone must put them back on line.
4. It takes the system clock off line for the duration of the outage; so someone must update the system time, using the TIME command from the master CLI process, PID 2.

- If the system console is displaying messages like

SPECIFY EACH DISK IN THE LDU
DISK UNIT NAME?

or

STARTING POWER UP SEQUENCE
ENTER DATE (MO DAY YR)

or

!

or nothing at all

this means that power has stopped to the CPU and that contents of volatile memory have gone away. You must cold start the system; then, run FIXUP on all initialized LDUs as described under "Running FIXUP". Tape units and line printers will need to be put back on line, and any aborted tape writes restarted.

What Next?

This chapter described CPU panels and switches, system startup, and normal shutdown. It also covered abnormal shutdown: deadlocks, panics, and hardware errors and power fails and it showed how to handle these with ESD and FIXUP. In short, this chapter covered the nitty-gritty steps to start, run, stop, and restart the system.

At first light, startup and shutdown seem complex — but they are really not: program load steps, disk unit name, enter the pathname, date/time, and default everything else. On abnormal shutdown, do a dump strike S and ⟩; if ESD fails, get an open file report and boot run FIXUP on LDUs with nonzero open file counts. On power restore, proceed as usual (with full battery backup), or run FIXUP (no backup or battery exhausted).

Now that you know how to start up and shut down your AOS system, you may want to learn more about PREDITOR, EXEC, other runtime tools, or System Management, described in later chapters.

End of Chapter

Chapter 7
About the PREDITOR Profile Editor

Read this chapter

- when you want to be more specific than in Chapter 5 about privileges for users on your system;
- when you want to understand the privileges that you can give or withhold from a user.

PREDITOR profiles, along with EXEC, provide the base for the multiuser environment. Profiles allow only authorized persons to use the system, without trespassing on other persons' or system files.

If you created your system's multiuser environment (described in Chapter 5), you already have some experience with PREDITOR. Chapter 5 described how to create a privileged operator profile and many general-purpose user profiles. As a base for the user profiles, the reader edited PREDITOR's default profile, then used the new default values as a basis for each user profile.

This chapter describes all PREDITOR commands — alphabetically. These commands allow you to create, edit, delete, and rename profiles, among other things. For ease of use, this chapter duplicates some of the material in Chapter 5. The major sections proceed as follows:

- About PREDITOR
- BYE Command
- CREATE and EDIT Commands, with Comments on Disk Space Control
- DELETE Command
- HELP Command
- LIST Command
- QUESTIONS Command
- RENAME Command
- USE Command
- PREDITOR Error Messages
- What Next?

About PREDITOR

The PREDITOR program file, PREDITOR.PR, is in directory :UTIL. It is a privileged program: only users with SUPERUSER privilege can run it. (You need not turn SUPERUSER on to run it; but you must have SUPERUSER privilege.)

For every user profile you create, PREDITOR creates a user directory in :UDD and a profile file in :UPD. The user directory and profile file have the username you give to PREDITOR. When you delete, edit, or rename a profile, PREDITOR makes the appropriate changes in :UDD or :UPD.

When a person tries to log onto AOS, or submits a batch job, EXEC checks the profile before it creates a user process to serve that person. If there is no valid profile, EXEC rejects the user or job. If there is a valid profile, EXEC creates a user process with the privileges defined in the profile. While the user process runs, it can exercise only these privileges. So, as you can see, profiles are central to multiuser system operation.

User Access Control List

When you create a new profile or rename an old one, PREDITOR sets the Access Control List (ACL) of the user directory to username,OWARE. This ACL gives the user *all* access privileges to the directory. No one else, except a SUPERUSER, can access it at all. This gives the user privacy, security, and unique access to all files and directories that he or she may create.

Executing PREDITOR

PREDITOR is in directory :UTIL, so, unless the working directory is :UTIL, your searchlist must include :UTIL. (The UP macro does this for the master CLI and its sons, so you need not do it if the multiuser environment is up.)

To execute PREDITOR, type

) XEQ PREDITOR ↵

AOS USER PROFILE EDITOR REV n date time

COMMAND:

PREDITOR is ready for a command.

If you want to change an answer that you previously gave to PREDITOR, type ↑ (press SHIFT and 6 keys) until you reach the bad entry. Then type the desired answer and proceed. For example

USE IPC [NO] (Y,N,OR NL)	Y ↵	Type answer.
USE CONSOLE: ^		Type ^ (SHIFT-6) to correct previous entry.
USE IPC [YES] (Y,N,OR NL)	N ↵	PREDITOR backs up and you change it.

If — at any point — you want to return to the *first* PREDITOR question, enter CTRL-C CTRL-A.

BYE Command — Leave PREDITOR

BYE↵ terminates PREDITOR and returns you to the CLI. It works only when PREDITOR is asking for a command. (As with any program, you can exit by aborting with CTRL-C CTRL-B — but a better solution is to use CTRL-C CTRL-A to return to the *COMMAND:* question.)

For example

COMMAND: BYE ↵

TERMINATING date time

)

CREATE and EDIT Commands — Create or Edit a Profile

The CREATE and EDIT commands allow you to create a new profile and edit an existing one. After either command, PREDITOR asks for answers to a series of parameter questions. You can suppress questions temporarily with the QUESTION command.

PREDITOR displays *[default]* values and valid answers (in parentheses) after each question. For the CREATE command, *[default]* is the value in PREDITOR's default profile, unless you told it to use another profile with a previous USE command. For the EDIT command, *[default]* is the current value in the profile you're editing. For example, it might say

INITIAL IPC FILE [:UDD:SAM:LOGON.CLI] CHANGE? (Y OR NL)

which means that the default IPC file for this profile is :UDD:SAM:LOGON.CLI and that you can answer Y↵ to change it or press ↵ (NEW LINE) to take the default.

When you EDIT a profile and change a value, the new value becomes effective the next time the user logs on. In other words, if you change a value while a user is logged on the system, the new value will not take effect until the user logs off and logs on again.

CREATE/EDIT Questions

When PREDITOR asks for a command, you can answer CREATE↵ (or abbreviation) or EDIT↵ (or abbreviation). Then PREDITOR asks the following series of questions.

USERNAME:

You must type a username; there is no default. Valid usernames are 1 to 15 valid filename characters long.

As part of itself, PREDITOR contains a default profile under the username of !DEFAULT!. You can edit !DEFAULT! just as you can any profile to change default values. This can speed up the creation of real user profiles by allowing you to answer most questions with ↵. The new default values you place in !DEFAULT! remain only until PREDITOR terminates; then, the original defaults return. !DEFAULT! is simply a convenience; no user profile or directory is associated with it.

When you want to edit a profile and can't remember the username, back up to the *COMMAND:* question with ^ (SHIFT-6). Then type BYE↵ to exit, and display the names of all users by typing

```
) SUPERU    ON ↵
*) DIR   :UDD ↵
*) F/S ↵
```

...(CLI displays sorted names of all user directories)...
*)

PASSWORD CHANGE? (Y OR NL)

When you create a profile, you must give a password. When you edit one, you can type Y↵ and enter the new password or press ↵ to retain the old one.

Each user must know his or her password to log on. So if you change it, be sure to let the user know. If a user forgets his/her password, you can simply enter a new one; you need not know the old password to change it with PREDITOR. A password can be 3 to 15 filename characters long.

By default, a user can change his or her password at log on, by pressing the ERASE PAGE key instead of ↵ after typing the old password.

INITIAL IPC FILE [default] CHANGE? (Y OR NL)

IPC means InterProcess Communication. The IPC file is a file the system will execute when this user logs on. It exists to communicate with the user's initial program and is not required. It usually contains a sequence of CLI commands — which may execute another program like BASIC.

The IPC filename may be the same for all users, since the file can reside in each user's directory, pathname :UDD:username:filename. Users can create or edit the IPC file as desired, to set searchlists, default access control lists, and so on. Only the first 512 characters in the file are read, so it shouldn't exceed 512 characters. The initial IPC filename should be memorable so that users won't forget it. Chapter 5, section "Making Life Easier for Users," shows how to create a sample IPC file and put it in user directories.

You can use an IPC to have a user log on and come up in CEO. Type Y ↓, and then :UTIL:CEO_DIR:CEO.STARTUP.CLI ↓. This will bring up the CLI, set the search list to include :UTIL:CEO_DIR, set the terminal's characteristics, and then overwrite the CLI program with the CEO Control Program, CEO_CP.PR. Similarly, if you have CEO Word Processing — Independent, you can have the user log on and come up in the Word Processor by typing :UTIL:CEO_DIR:CEO.WP.STARTUP.CLI. ↓

If you wish to take the default, press ↓. Otherwise, type Y↓ and PREDITOR will ask for the new name, 0-63 characters. You must type the full pathname from the root, for example:

:UDD:username:LOGON.CLI ↓.

PROGRAM [default] CHANGE? (Y OR NL)

PROGRAM is the program the system will execute for the user when he/she logs on or submits batch jobs. If the user is allowed no sons (asked later), he or she will be restricted to this program — which may be just what you want.

The default program, :CLI.PR, is a good general-purpose choice. The CLI allows users to access text editors and build programs in all DG languages. It also allows users to execute other programs like BASIC. Take the default unless you want a non-CLI program run automatically — as for a BASIC user.

For a program other than the CLI, type Y↓ and the full pathname, with .PR suffix, of the program you want. For BASIC, there is often a BASIC directory off the root or :UTIL, if so, answer Y↓, then type :BASIC.PR↓ or :UTIL:BASIC.PR↓ to have the user come up in BASIC. (To work in BASIC, the IPC file must be a BASIC program.) To ensure that the user stays in BASIC (or other program), you must specify no sons, later on.

* *CREATE WITHOUT BLOCK [default] CHANGE? (Y, N, OR NL]*

A CREATE WITHOUT BLOCK value of YES (Y) allows the user to have at least two processes running concurrently. By default, the creating (father) process is blocked when it executes the son; this means that the father is eligible to be swapped, which may speed up the system. But if the user needs DG's SWAT debugger (for FORTRAN 77, PL/I, or COBOL programs), he or she must have CREATE WITHOUT BLOCK privilege.

CEO users must have this privilege also; and programmers often need it. Other users usually *don't* need this privilege. Take the default or change it as appropriate.

USE IPC [default] CHANGE? (Y, N, OR NL)

IPC privileges allow a person to use IPC calls, available in assembly language and some higher-level languages. IPC privileges are needed wherever two or more active processes must communicate. For IPC usage to work, a user may also need the CREATE WITHOUT BLOCK, if two or more of *that user's* processes are to use IPC.

CEO users must have the USE IPC privilege; most other people don't need it. Take the default or change it as appropriate.

USE CONSOLE [default] CHANGE? (Y, N, OR NL)

A user must have this privilege to log onto a console under EXEC. Without it, he or she may be able to submit batch jobs via a card reader. But nearly all users need this privilege. Take the default or change it as appropriate.

USE BATCH [default] CHANGE? (Y, N, OR NL)

A user must have this privilege to submit batch jobs via CLI QBATCH or QSUBMIT commands, or via a card reader. Depending on your system, you may or may not want to encourage batch jobs. Take the default or change it as appropriate.

MODEM [default]? (Y, N, OR NL)

If you want this user to be able to log on via a modem, this value should be YES. Generally, SUPERUSERs should not be able to use a modem, for the two privileges allow the user to explore the entire system from his own home or wherever a remote console is placed. To take the default, press ⤶; to change it, type the new value and ⤶.

USE VIRTUAL CONSOLE [default]? (Y, N, OR NL)

This question, and the next, are meaningful only if your system will run DG's XODIAC networking software. A YES (Y) value enables the user to log onto your system from a virtual console or use the XODIAC loopback feature.

Take the default or change it according to your wishes for this user.

ACCESS LOCAL RESOURCES FROM REMOTE MACHINES [default]? (Y, N, OR NL)

A YES (Y) value allows a remote user to access files and devices like tapes and printers on your system. This is different from being able to log on, as covered in the previous question. For remote resource access, the user must have a profile with the same username and password (but not necessarily the same privileges) on *both systems*. Details are in the *XODIAC User's Manual*. Depending on what you want this user to do from a remote system, take the default or change it.

CHANGE PASSWORD [default]? (Y, N, OR NL)

In general, users should be able to change their own passwords, per a YES value. But if this is a GUEST profile, to allow guests to use your system, the password must be public; the value should be NO (N) to prevent a guest from changing the password and barring other guests from the system. If this value is NO, the only way to change the password is with PREDITOR. Take the default or change it as desired.

UNLIMITED SONS [default]? (Y, N, OR NL)

A user who can create unlimited son processes has the potential for dominating the system. Each process requires some CPU time and disk I/O to the swap file. So far as possible, it's a good idea to minimize the number of processes. Generally, this value should be NO. Take the default or change it as *
you wish. If you type Y⤶, PREDITOR skips the next question.

SONS [default] CHANGE? (Y OR NL)

BASIC and clerical data entry users can get along with few sons: 0 or 1. To limit a user to the program specified in the PROGRAM question, the number of sons must be 0. CLI users who want to execute a non-CLI son process from within a son (instead of going back to the CLI to do it) need at least 5 sons. For other users, 3 sons is a good general-purpose value. It will allow a user to develop programs and use the SWAT debugger; it's a minimum for serious application programmers who will use FORTRAN 77, PL/I, or COBOL.

To take the default, press ⤶. To change it, type Y⤶; PREDITOR will say NEW(0-255) and you will type the new number and ⤶.

CHANGE PRIORITY [default]? (Y, N, OR NL)

Processes compete for CPU time, and processes of the same type with higher priority (closer to 0) get preference. But it's simpler and often better if all processes of the same type have the same priority. So, generally, processes should retain their initial priority.

To take the default, press ⏎. To change it, type Y⏎.

CHANGE TYPE [default]? (Y, N, OR NL)

Processes can run as one of three types: resident (always in main memory), pre-emptible (generally in main memory, but swappable if blocked), and swappable. Resident and pre-emptible always have priority over swappable.

Swappable is the most common type. It is the default for user processes. Resident is quite rare — often used only for the AOS peripheral manager and AOS itself. If a process can change type, it can become resident, and perhaps hobble the system. So, unless you know that a process must be able to change its type, this value should be NO. To take the default, press ⏎; to change it, type the new value and ⏎.

CHANGE USERNAME [default]? (Y, N, OR NL)

A process that can change its username can assume the name of OP or another SUPERUSER — giving it access to the entire system. Again, unless you know that the process must be able to change its username, this value should be NO. To take the default, press ⏎; to change it, type the new value and ⏎.

ACCESS DEVICES [default]? (Y, N, OR NL)

This privilege allows a process to bypass operating system safeguards and access devices directly, via privileged hardware I/O instructions. You should never give it unless the user is a programmer who needs it to write or debug device drivers. So, in nearly all cases, the value should be NO. To take the default, press ⏎; to change it, type the new value and ⏎.

SUPERUSER [default]? (Y, N, OR NL)

This privilege allows a user process to bypass all file access controls and execute, read, modify, or delete *any file* on the system. SUPERUSERs can run PREDITOR to change their own profiles; and they can find other users' usernames and passwords in their profile files. The master CLI needs SUPERUSER to control the system; but most other users do not need it, and they shouldn't have it. In most cases, the value should be NO. To take the default, press ⏎; to change it, type the new value and ⏎.

SUPERPROCESS [default]? (Y, N, OR NL)

This privilege allows a user process to issue process control commands against any process. It can block a process, become resident, or terminate any process, including the master CLI, which would bring down the entire system. Unless you know that a process needs SUPERPROCESS, the value should be NO. To take the default, press ⏎; to change it, type the new value and ⏎.

BECOME INFOS [default] CHANGE (Y, N, OR NL)

Any user who will start an INFOS II process needs this privilege. A user doesn't need it to *use* INFOS II. Normally, the operator process (PID 2) creates the INFOS II process. Most users don't need this privilege. To take the default, press ⏎; to change it, type the new value and ⏎.

MEMORY [default] CHANGE (Y OR NL)

This governs the maximum number of 2,048-byte memory pages this user's processes can use. Generally, each user should have the maximum of 32 memory pages. To take the default, press ⏎; to change it, type Y⏎; PREDITOR will then ask *NEW (1-32)* and you will type the new value and ⏎.

PRIORITY [default] CHANGE (Y OR NL)

Each initial user process is swappable. Swappable processes can have priority 1, 2, or 3. Generally, you will want to maintain process equality — usually a value of 2. To take the default, press ⏎. To change it, type Y⏎; PREDITOR will say NEW(1-3) and you'll type the new value and ⏎.

MAX QPRIORITY [default] CHANGE (Y OR NL)

Users type Q-series commands to print files and submit batch jobs. Priority 0 is the default and highest priority for these. If all users have the same QPRIORITY, they will receive equal treatment on their Q-series jobs. Generally — unless you want to prioritize different users' Q-series requests — keep the same value — usually 0. To take the default, press ⏎. To change it, type Y⏎; PREDITOR will ask NEW(0-255) and you'll type the new value and ⏎.

DISK QUOTA [default] CHANGE (Y OR NL)

This sets the limit on the size of the user directory that PREDITOR will create and that the system will maintain for this user process. (But this PREDITOR-assigned limit doesn't affect users in CEO.)

A good general-purpose amount of disk space is 15000 blocks. If this user process will serve CEO, guest, or other casual users, you might want to specify less space (perhaps the original default, 500). If this user process will serve many people (perhaps data entry clerks or students), you might want to specify a larger figure (e.g., 100000).

If this user process will deal with a large database and its directory will contain the database(s), you might want to allot an entire single- or multiple-disk LDU to it. A model 6061 disk contains about 370000 blocks; a model 6122 disk contains about 540000 blocks.

To take the default, press ↵. To change the space quota, type Y↵. PREDITOR will then ask *NEW(0-2147483647):* and you will type the new quota. For more on disk space quotas, see "Disk Space Control," below.

USER COMMENT [default] CHANGE? (Y OR NL)?

You can use this for text comment about the user: full name, date, etc. Or you can take the default by pressing ↵. To enter or change a comment, type Y↵, then the desired command; for example,

Y ↵

NEW (0-79 CHARS): JACK ARMSTRONG. GIVEN MODEM 17 APR 85 ↵

COMMAND:

You've finished the CREATE/EDIT session; and the new/edited profile is ready for its user. An EDIT example follows the next section.

Disk Space Control

PREDITOR's DISK QUOTA sets the maximum amount of disk space for a user not in CEO. This space includes all files and subordinate directories the user may create; it also includes space requirements of all processes run under this user process.

The quota limit should be large enough to allow the user (process) to work effectively. High quotas help prevent users from running out of space — an error situation that requires the system operator to edit profiles and provide more space.

On any LDU, you can allot users more disk space than actually exists. For example, if the LDU contains 370000 blocks, you can give each of 40 users a quota of 10000 blocks. This is called *oversubscribing* an LDU. It is possible because the quota is not actually *reserved* for each user; it is a theoretical limit, but is not guaranteed to the user.

The advantage of oversubscribing is that it allows generous disk quotas, which prevents users from running out of disk space and needing new profiles (with more space). There are several disadvantages: 1) users can become careless and saturate the LDU with unimportant files; 2) the LDU may become nearly filled with files, which will slow the system; 3) the LDU may be entirely filled, which will cause all users to get error messages when they try to create files, even though their individual disk quotas haven't been reached.

To check the *actual* amount of free space on an LDU, use the CLI command SPACE, with the LDU master directory name as an argument; for example type SPACE :↵. If 10% or less of the total LDU space remains, you will want to free some — by moving files to another LDU or dumping files, then deleting them from this LDU; or by telling users to clean up their directories.

When you *edit* a profile, don't reduce the disk quota below the amount of space the user is actually using. If you do this, the user won't be able to log on. When you wish to reduce a disk quota, use the SPACE :udd:username ↵ command before running PREDITOR to see how much space is occupied; this is the minimum disk quota you should specify to PREDITOR for this user.

EDIT Example

The following example shows several editing changes in the profile of user JACK.

```
) X  PREDITOR ↵

USER PROFILE EDITOR REV n date time

COMMAND:  E ↵
USERNAME:  JACK ↵
PASSWORD CHANGE? (Y OR NL)  ↵
INITIAL IPC FILE [:UDD:JACK:LOGON.CLI] CHANGE? (Y OR NL)  ↵
PROGRAM [CLI.PR] CHANGE? (Y OR NL)  ↵
CREATE WITHOUT BLOCK [NO]? (Y, N, OR NL)  Y ↵
USE IPC [NO]? (Y, N, OR NL)  Y ↵
USE CONSOLE [Y]? (Y, N, OR NL)  ↵
USE BATCH [YES]? (Y, N, OR NL)  ↵
USE MODEM [NO]? (Y, N, OR NL)  Y ↵
USE VIRTUAL CONSOLE [YES]? (Y, N, OR NL)  ↵
ACCESS LOCAL RESOURCES FROM REMOTE MACHINES [YES]? (Y, N, OR NL)  ↵
CHANGE PASSWORD [YES]? (Y, N, OR NL)  ↵
UNLIMITED SONS [NO]? (Y, N, OR NL)  ↵
SONS [3] CHANGE? (Y OR NL)  Y ↵
NEW (0-255):  5 ↵
CHANGE PRIORITY [NO]? (Y, N, OR NL)  ↵
CHANGE TYPE [NO]? (Y, N, OR NL)  ↵
CHANGE USERNAME [NO]? (Y, N, OR NL)  ↵
ACCESS DEVICES [NO]? (Y, N, OR NL)  ↵
SUPERUSER [NO]? (Y, N, OR NL)  ↵
SUPERPROCESS [NO]? (Y, N, OR NL)  ↵
BECOME INFOS [N] CHANGE (Y, N OR NL)  ↵
MEMORY [32] CHANGE (Y, N, OR NL)  ↵
PRIORITY [2] CHANGE (Y OR NL)  ↵
MAX QPRIORITY [0] CHANGE (Y OR NL)  ↵
DISK QUOTA [15000] CHANGE (Y OR NL)  Y ↵
NEW (0-2147483647):  20000 ↵
USER COMMENT [JACK ARMSTRONG 9 MAR 85] CHANGE? (Y OR NL)?  Y ↵
NEW (0-79 CHARS):  Jack Armstrong. Given MODEM  17 Apr 85 ↵
COMMAND:  BYE ↵

TERMINATING date time

)
```

DELETE Command — Delete a User Profile

PREDITOR's DELETE command deletes a user profile — preventing the user from using the system. Optionally, it also deletes the user directory.

PREDITOR asks for a username, then asks for confirmation of the profile delete, and then asks if you wish to delete the user directory. If you reply Y↵, PREDITOR deletes the user directory and all its subordinate directories.

You would use this command whenever you wanted to terminate someone's ability to use the system — perhaps because this person had left the organization or was no longer a suitable timesharing user. For the record, you might want to keep a DUMP copy of the profile (:UPD:username) and/or selected files (template :UDD:username:#).

DELETE Dialog and Example

COMMAND: DELETE ↵
USERNAME: SIMON ↵
DELETE USER :UPD:SIMON? (Y OR N) Y ↵
USER DELETED
DELETE USER DIRECTORY? (Y OR N) Y ↵
USER DIRECTORY DELETED

COMMAND:

EDIT Command — Edit an Existing Profile

See the CREATE command.

HELP Command — Describe PREDITOR Commands

HELP gives summary information on all PREDITOR commands.

HELP Dialog and Example

COMMAND: HEL ↵

THE LEGAL COMMANDS ARE:

BYE
CREATE
DELETE
EDIT
HELP
LIST
QUESTION (TURN QUESTIONS ON OR OFF)
RENAME (RENAME A PROFILE & USER DIRECTORY)
USE (USE ANOTHER PROFILE AS !DEFAULT!)

COMMAND:

LIST Command — Display Values in a Profile

The LIST command asks for a username, then displays all the current values in a profile. It's useful when you want to know every value in a user profile.

LIST Dialog and Example

COMMAND: LIST ⤶
USERNAME: JACK ⤶
INITIAL IPC FILE [:UDD:JACK:LOGON.CLI]
PROGRAM [CLI.PR]
CREATE WITHOUT BLOCK [YES]
USE IPC [N]
USE CONSOLE [YES]
USE BATCH [YES]
MODEM [YES]
USE VIRTUAL CONSOLE [YES]
ACCESS LOCAL RESOURCES FROM REMOTE MACHINES [YES]
CHANGE PASSWORD [YES]
UNLIMITED SONS [NO]
SONS [5]
CHANGE PRIORITY [NO]
CHANGE TYPE [NO]
CHANGE USERNAME [NO]
ACCESS DEVICES [NO]
SUPERUSER [NO]
SUPERPROCESS [NO]
BECOME INFOS [NO]
MEMORY [32]
PRIORITY [2]
MAX QPRIORITY [0]
DISK QUOTA [20000]
USER COMMENT [JACK ARMSTRONG. GIVEN MODEM 17 APR 85]

COMMAND:

QUESTION Command — Suppress or Restore PREDITOR Questions

The QUESTION command turns off display of any or all PREDITOR questions. The questions you suppress remain suppressed until you re-issue the QUESTION command or leave PREDITOR.

When you type QUESTION, PREDITOR displays each question individually, in the CREATE/EDIT order. Type ⤶ if you want PREDITOR to ask the question; type N⤶ to suppress the question. Instead of the usual [default] value, PREDITOR displays [Y], to indicate that by default it *will* display the question.

QUESTION is handy when you wish to change only a few parameters for many users on the system. You can suppress all the irrelevant questions with QUESTION, then edit all the profiles you want quickly.

WARNING: Don't create a new profile with questions suppressed; such a profile will likely be unusable. Suppress questions only when you plan to edit profiles.

QUESTION Dialog and Example

In the following example, the person running PREDITOR wants to edit only the CREATE WITHOUT BLOCK, DISK QUOTA, and USER COMMENT values for users SAM, F77, and others. The QUESTION command helps speed this up.

```
COMMAND:    Q ⤶
PASSWORD [Y]? (Y OR NL)    N ⤶
INITIAL IPC FILE [Y]? (Y OR NL)    N ⤶
PROGRAM [Y] CHANGE? (Y OR NL)    N ⤶
CREATE WITHOUT BLOCK [Y]? (Y, N, OR NL)    ⤶
USE IPC [N]? (Y, N, OR NL)    N ⤶
USE CONSOLE [Y]? (Y, N, OR NL)    N ⤶
USE BATCH [Y]? (Y, N, OR NL)    N ⤶
MODEM [Y]? (Y, N, OR NL)    N ⤶
USE VIRTUAL CONSOLE [Y]? (Y, N, OR NL)    N ⤶
ACCESS LOCAL RESOURCES FROM REMOTE MACHINES [Y]? (Y, N, OR NL)    N ⤶
CHANGE PASSWORD [Y]? (Y, N, OR NL)    N ⤶
UNLIMITED SONS [Y]? (Y, N, OR NL)    N ⤶
CHANGE PRIORITY [Y]? (Y, N, OR NL)    N ⤶
CHANGE TYPE [Y]? (Y, N, OR NL)    N ⤶
CHANGE USERNAME [Y]? (Y, N, OR NL)    N ⤶
ACCESS DEVICES [Y]? (Y, N, OR NL)    N ⤶
SUPERUSER [Y]? (Y, N, OR NL)    N ⤶
SUPERPROCESS [Y]? (Y, N, OR NL)    N ⤶
BECOME INFOS [N]    N ⤶
MEMORY [32]    N ⤶
PRIORITY [Y] CHANGE (Y OR NL)    N ⤶
MAX QPRIORITY [Y] CHANGE (Y OR NL)    N ⤶
DISK QUOTA [Y] CHANGE (Y OR NL)    ⤶
USER COMMENT [Y] CHANGE? (Y OR NL)?    ⤶

COMMAND:    EDIT ⤶
USERNAME:    SAM ⤶
CREATE WITHOUT BLOCK [NO]? (Y, N, OR NL)    Y ⤶
DISK QUOTA [15000] CHANGE (Y OR NL)    Y ⤶
NEW (0-2147483647):    25000 ⤶
USER COMMENT [SAM WILLIAMS 1 MAR 85] CHANGE? (Y OR NL)?    Y ⤶
NEW (0-79 CHARS):    SAM WILLIAMS, NO BLOCK, 25K BLOCKS, 17 APR 85 ⤶

COMMAND:    ED ⤶
USERNAME:    F77 ⤶
CREATE WITHOUT BLOCK [YES]? (Y, N, OR NL)    ⤶
DISK QUOTA [50000] CHANGE (Y OR NL)    ⤶
NEW (0-2147483647):    100000 ⤶
USER COMMENT [F77 PROGRAMMERS 30 MAR 85] CHANGE? (Y OR NL)?    Y ⤶
NEW (0-79 CHARS):    F77 PROGRAMMERS UP TO 100K BLOCKS 19 APR 85 ⤶
COMMAND:
```

RENAME Command — Rename a Profile

The RENAME command renames a user profile and its associated user directory. Within the profile, only the USERNAME changes; the password and all other values remain unchanged.

RENAME also changes the ACL for the user directory to new-username,OWARE — giving the new username all access privileges to the directory. This may cause problems if the user is logged on under the old name — so you should rename a profile only when the original username is not logged on.

All files and subordinate directories within the user directory retain their old ACLs — preventing access by the new username. After renaming a profile, either you or the user should update these ACLs via the CLI by typing

```
) DIR   :UDD:username ↵
) ACL/V   #   new-username,OWARE ↵
.
.
.
(CLI verifies new ACLs of all directories and files)
.
.
.
)
```

If you (not the user) change the ACLs, you will need to turn SUPERUSER on first.

You might use the RENAME command if a user didn't like his or her assigned USERNAME, or if a more explicit or descriptive username was desirable.

RENAME Dialog and Example

COMMAND: RENAME ↵
USERNAME: SAL ↵
NEW USERNAME: SALLY ↵

COMMAND: BYE ↵

TERMINATING date time

```
) SUPERU   ON ↵
*) DIR   :UDD:SALLY ↵
*) ACL/V   #   SALLY, OWARE ↵
.
.
.
(verification)
.
.
.
*) SUPERU   OFF ↵
) DIR/I   ↵
)
```

USE Command — Use another Profile as !DEFAULT!

Normally, PREDITOR displays the values in its !DEFAULT! profile as defaults for CREATE and EDIT questions. The USE command tells it to use the values in *another profile* for these defaults.

During a PREDITOR session, you can edit !DEFAULT! to change the defaults — but these changes remain only until PREDITOR terminates. User profiles, on the other hand, remain stable. So, when you create a new profile that will be similar to an existing one, you can simply USE the existing one and change only the username, password, and possibly a few other values. PREDITOR continues to use the values in the existing profile until it terminates or until you tell it to use another profile.

The USE command doesn't affect the existing profile; the values in it simply serve as defaults.

USE Dialog and Example

This example shows existing user JACK's profile used to help create a profile for user BARBARA.

```
) X  PREDITOR ⤶
AOS/VS USER PROFILE EDITOR REV n date time
COMMAND:    USE ⤶
USERNAME:    JACK ⤶
.
.
.
(PREDITOR displays values in JACK's profile)
.
.
.
COMMAND:    CREATE ⤶
USERNAME:    BARBARA ⤶
PASSWORD CHANGE? (Y OR NL)    Y ⤶
NEW (3-15 CHARS):    BARB ⤶
INITIAL IPC FILE [:UDD:JACK:LOGON.CLI] CHANGE? (Y OR NL)    Y ⤶
NEW (0-63 CHARS):    :UDD:BARBARA:LOGON.CLI ⤶
PROGRAM [CLI.PR] CHANGE? (Y OR NL)    ⤶
CREATE WITHOUT BLOCK [YES]? (Y, N, OR NL)    N ⤶
USE IPC [NO]? (Y, N, OR NL)    ⤶
USE CONSOLE [YES]? (Y, N, OR NL)    ⤶
USE BATCH [YES]? (Y, N, OR NL)    ⤶
MODEM [NO]? (Y, N, OR NL)    Y ⤶
USE VIRTUAL CONSOLE [YES]? (Y, N, OR NL)    ⤶
ACCESS LOCAL RESOURCES FROM REMOTE MACHINES [YES]? (Y, N, OR NL)    ⤶
CHANGE PASSWORD [YES]? (Y, N, OR NL)    ⤶
UNLIMITED SONS [NO]? (Y, N, OR NL)    ⤶
SONS [5] CHANGE? (Y OR NL)    ⤶
CHANGE PRIORITY [NO]? (Y, N, OR NL)    ⤶
CHANGE TYPE [NO]? (Y, N, OR NL)    ⤶
CHANGE USERNAME [NO]? (Y, N, OR NL)    ⤶
ACCESS DEVICES [NO]? (Y, N, OR NL)    ⤶
SUPERUSER [NO]? (Y, N, OR NL)    ⤶
SUPERPROCESS [NO]? (Y, N, OR NL)    ⤶
BECOME INFOS [N]    ⤶
MEMORY [32]    ⤶
PRIORITY [2] CHANGE (Y OR NL)    ⤶
MAX QPRIORITY [0] CHANGE (Y OR NL)    ⤶
DISK QUOTA [20000] CHANGE (Y OR NL)    ⤶
USER COMMENT [JACK ARMSTRONG 17 APR 85] CHANGE? (Y OR NL)?    Y ⤶
NEW (0-79 CHARS):    BARBARA CLEAVES, 1 JUN 85 ⤶
COMMAND:
```

PREDITOR Error Messages

While you are operating PREDITOR, you may receive one or more of the error messages shown in Table 7-1.

Table 7-1. PREDITOR Error Messages

Message	Meaning and Action
?1 TO 15 CHARACTERS NEEDED	The username must be from 1 to 15 characters. Try a username with the correct number of characters.
?ILLEGAL CHARACTER IN STRING	You typed a character that PREDITOR can't understand. Retype the entry without the questionable character.
?NAME IN USE FOR A NON-PROFILE	The username you typed is the same as the name of a nonprofile file in directory :UPD. To recover, try another username. (Generally, :UPD should be reserved for PREDITOR-created files.)
?PATHNAME MUST START AT ROOT	All filenames you give to PREDITOR must be full pathnames from the root directory (:); for example, :UDD:JACK:LOGON.CLI. Retype the entry as a full pathname.
?TYPE H FOR HELP	Your entry was not a PREDITOR command. Type HELP⤶ to display all commands.
?USER DIRECTORY ALREADY EXISTS	The username you specified already has a user directory (but not a valid profile). Use the EDIT command to produce a valid profile, or use the DELETE command to delete the existing profile and directory.
?USERNAME ALREADY IN USE	The username you specified is in use, with a profile and directory. Try another username.
?USERNAME DOES NOT EXIST	The username given doesn't exist. Back up to *COMMAND:* with CTRL-C CTRL-A and leave PREDITOR with BYE⤶. Then check all user directory names by typing) SUPERU ON ⤶ *) DIR :UDD ⤶ *) F/S ⤶ ...(CLI displays sorted names)... *) Then XEQ PREDITOR and try the command again.
YOU CAN'T DELETE !DEFAULT!	The !DEFAULT! profile is part of PREDITOR and you can't delete it.

What Next?

This chapter described all PREDITOR commands and how to use them. You may wish to try the profile(s) you've worked with — or have users log on and try them.

You may want to learn more about another major multiuser tool, EXEC, described in the next chapter; or about other runtime tools, described in Chapter 9; or about System Management, described in Chapter 14.

End of Chapter

Chapter 8
EXEC and User Processes

Read this chapter

- when you want to learn about the EXEC program or any of its commands;
- when you want to learn how user logon, batch, or device queues work;
- when you want to learn how to handle user tape mount and dismount requests.

EXEC — a program in :UTIL — supervises the AOS multiuser environment. If you created your system's multiuser environment (described in Chapter 5) you have some experience with EXEC. Chapter 5 had the reader start up EXEC, initialize EXEC queues, enable user consoles, log on as a user, check EXEC's spool and batch functions, and log off. Then it showed how to edit the DG-supplied UP and DOWN macros so that EXEC could be brought up via UP) and shut down via DOWN).

This chapter tells all about EXEC — its functions, commands, and operator and user messages. The major sections proceed as follows.

- What EXEC Does
- Creating and Terminating the EXEC Process
- Pertinent CLI Commands
- EXEC Command Overview
- General Commands to EXEC
- About User Log On
- About Batch and Spooling
- About User Tape Mount Requests
- EXEC Commands — Listed Alphabetically
- EXEC Messages
- What Next?

Appendix C, near the end of this book, is a concise, alphabetical summary of all EXEC commands.

What EXEC Does

When it is running, EXEC can do the following things.

1. Log users on and off. When a user tries to log on, EXEC checks for a valid profile; then, if the username and password that the user types match those in a profile, EXEC creates a user process, with parameters defined in the profile, for that user. When the user process terminates, EXEC logs the user off.

2. Manage batch and spool queues. EXEC maintains a batch input, list, and output file for each batch request; it has 4 batch streams to handle batch requests. EXEC also spools output to devices like line printers. Users queue batch and printing requests with CLI Q-series commands — and EXEC manages them with little or no operator intervention.

3. Help manage user requests to mount and dismount tapes on tape units.

EXEC has commands that control all these functions. But usually — after the UP.CLI macro brings it up — EXEC runs practically by itself.

All the EXEC files are in directory :UTIL. These include the EXEC program and overlay files, EXEC.PR and EXEC.OL; EXEC's card reader program, STACKER.PR; EXEC's line printer program, XLPT.PR; EXEC's digital plotter program, XPLT.PR; EXEC's Format Control Utility (FCU.PR); and EXEC's queue compacter program, QCMP.PR

Creating and Terminating the EXEC Process

The UP.CLI macro includes the PROCESS command that creates EXEC and EXEC commands to enable consoles and start queues. You can TYPE or QPRINT file :UP.CLI to see them. But, for completeness, we will show the PROCESS command here. It is

```
PROCESS/DEFAULT/DIRECTORY=@/NAME=EXEC   EXEC
```

where

PROCESS	is a CLI command that creates a new process (just as XEQ does, but PROCESS accepts more switches than XEQ, thus is more versatile).
/DEFAULT	Gives the new process all the privileges of the father process. The father is nearly always the master CLI, PID 2, which has the SUPER and USE IPC privileges that EXEC needs.
/DIRECTORY=@	Makes the peripheral directory (:PER, shorthand @) the home directory of the EXEC process. This is necessary because all the device entries EXEC needs are in :PER and the EXEC program file is in :UTIL.
/NAME=EXEC	Gives the EXEC process the simple process name EXEC so that the system can access it by this process name.
EXEC	is the program name to execute; here, EXEC.

When EXEC starts up, it checks for another EXEC process; if an EXEC is already running, the new EXEC will terminate with an error message. EXEC runs as a swappable process, since it runs only for brief periods. When ready, EXEC says

FROM PID n : (EXEC) REV n READY
FROM PID n : (EXEC) hours:minutes:seconds

EXEC is ready for CONTROL @EXEC (CX) commands — many of which follow the PROCESS command in the UP.CLI macro.

The master CLI, PID 2, is usually EXEC's father. The user processes EXEC creates are EXEC's sons. The process hierarchy looks like Figure 8-1.

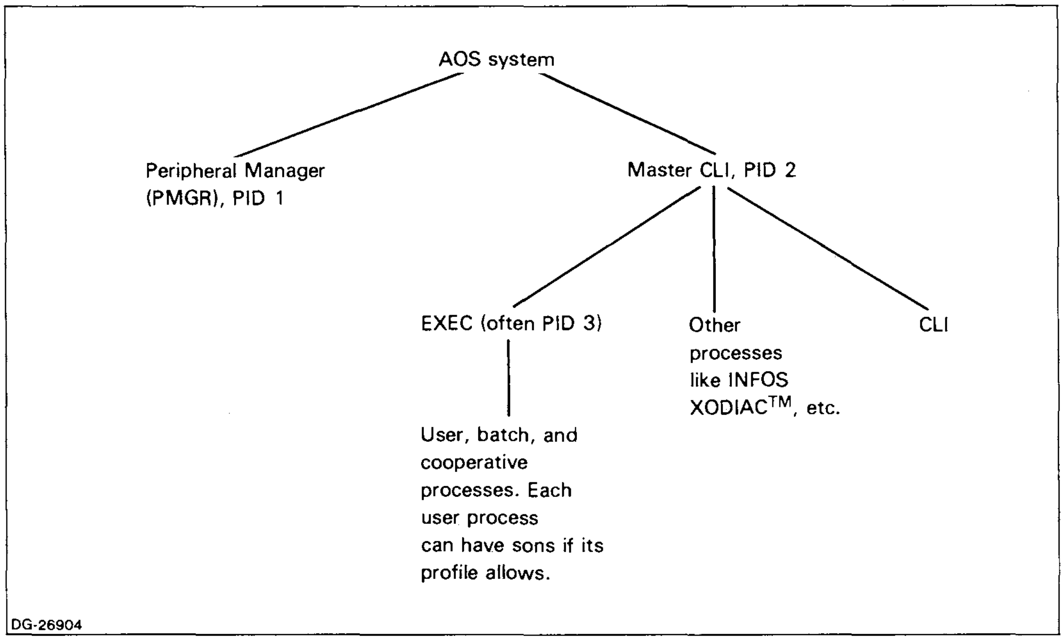


Figure 8-1. Process Hierarchy (Tree)

Terminating EXEC

As with creation, the TERMINATE command that terminates EXEC is in a macro: DOWN.CLI. Another command in DOWN.CLI — CHECKTERMS — relates to TERMINATE, so we'll explain both. They are

```
TERMINATE/2=ERROR   OP:EXEC
.
.
.
CHECKTERMS
.
.
.
```

where

TERMINATE	is a CLI command that terminates a process. The target process must be a son of the issuing process unless Superprocess is on.
/2=ERROR	tells the CLI to stop executing commands in the macro if an error occurs on the TERMINATE (otherwise, it might continue with the macro on some kinds of error, which would be undesirable).
OP:EXEC	is the full process name of the EXEC process. The master CLI is the father; EXEC is the son.
CHECKTERMS	tells the CLI to save and display messages that the system may display after it terminates a process (check termination messages).

Normally, you would pause EXEC queues and warn users before bringing down EXEC via the DOWN macro. Terminating a process terminates all its sons; and each user process created by EXEC is a son of EXEC. Users might lose work and be a bit upset if their processes died suddenly.

Pertinent CLI Commands

While EXEC runs, you'll use some or all of the CLI commands shown in Table 8-1 (given more detail in the CLI manual). You can also use some other macros described in Chapter 5.

Table 8-1. Process-Oriented CLI Commands

Command	What it Does	Example
ACL	Sets the access control list for one or more files; further described in Chapter 9, "Other Runtime Tools."	) ACL/V + OP,OWARE ⤶
BYE	Terminates the current CLI or SED text editor process. The process' father (from which the process was executed) gets control. If this is a user process, BYE logs the user off. If there is no father, as with the master CLI, BYE starts system shutdown.	) BYE ⤶ *PROCESS ...TERMINATED*
CONTROL @EXEC cmd or CX cmd	CONTROL @EXEC tells the CLI to pass an IPC message of "cmd" to EXEC. The "cmd" is one the the EXEC commands described in this chapter. CX invokes macro CX.CLI which contains the characters CONTROL @EXEC %-% allowing you to type CX instead of CONTROL @EXEC. If this macro doesn't exist, create it as described in Chapter 5.	) CONTROL @EXEC STAT ⤶) CX STAT ⤶
EXECUTE program	Creates a process; same as XEQ.	See XEQ.
FILESTATUS template	Describes filenames in any directory. Handy switches are /AS (for an assortment of information), and /S (for an alpha sort). If you get an ACCESS DENIED message, turn SUPERUSER on and try again.	) F/AS/S :UTIL:EXEC+ ⤶
HELP *[command]*	Gives help on CLI topics or commands.	) HELP ⤶) HE/V ACL ⤶
PROCESS arguments	Creates a process; described in previous section.	) PROC/PRI=1/DEF PROG ⤶
QBATCH, QPRINT	These are *user* commands that place requests on EXEC batch and spool queues. To use them, a person needs a user profile. So, even if you are the system manager or operator, you must have a profile. Chapter 5 has the reader create a privileged profile named OP.	) QBATCH X MYPROG ⤶) QPRINT MYFILE ⤶

(continues)

Table 8-1. Process-Oriented CLI Commands

Command	What it Does	Example
QDISPLAY	Describes the status of all batch and spool queues. For more information, include the /V switch.	) QDIS/V ⤶
RUNTIME *[pid]*	Describes how long a process has been running, and its CPU and I/O usage. The *pid* is the process ID; if you omit it, the command describes the current process.	) RUN 3 ⤶) RUN ⤶
SEND user(s)	Sends a message to one or more user consoles. The user(s) can be a process ID or a template like @CON- to specify all consoles. The BROADCAST macro, described in Chapter 5, is easier and somewhat more elegant than "SEND @CON-".	) SEND @CON13 Logoff ⤶) SEND @CON- Logoff ⤶) BROADCAST Logoff ⤶
TREE *[pid]*	Describes a process' family tree: process IDs of father, self, and son(s) if any.	) TREE 3 ⤶) TREE ⤶
TERMINATE process	Terminate a process, shown in the previous section. The "process" can be a full process name like OP:EXEC, a simple process name like EXEC, or a Process ID like 3.	) TERM 20 ⤶
WHO *[pid]*	Describes the username associated with a process ID.	) WHO 4 ⤶) WHO ⤶
?	Invokes a CLI macro, ?.CLI, that contains a series of WHO commands; describes all users on the system. Creating this macro is described in Chapter 5.	) ? ⤶

(concluded)

EXEC Command Overview

EXEC has a help feature that you can tap by typing

) XHELP ⤶

or

) XHELP cmd ⤶

where cmd is an EXEC command.

EXEC's help messages are not tutorial, but they can be very useful when you have forgotten a command word or the correct syntax of a command.

EXEC commands are special in that any process with the username OP can issue them. This can be very important if — as is often true — the CLI running on the system console is not the master CLI, PID 2. The UP.CLI macro has an EXECUTE CLI (or EXECUTE LOCK_CLI) command, which runs a son CLI on the system console. This son CLI may not have the SUPER privileges of its father — which means that the system operator may not be able to use this CLI to control the system. But *any* CLI with a username of OP can issue EXEC commands. So it doesn't matter which CLI the operator uses — if its username is OP, EXEC will obey the operator's EXEC commands.

Aside from XHELP, each EXEC command begins CONTROL @EXEC or CX. There are a lot of EXEC commands, but in daily routine, you will need only a few. Table 8-2 shows these frequently used EXEC commands.

Command Abbreviations

As with the CLI, you can abbreviate any EXEC command to its shortest unique string. For example, instead of CX STATUS, you could type

```
) CX   STAT ↵
```

Don't be afraid to try abbreviations; at worst you'll get a harmless *NOT UNIQUE* error message.

EXEC Command Response Messages

EXEC is a very talkative program, describing practically everything it does.

After you type a command to it, EXEC will acknowledge the command with either a

FROM PID n : (EXEC) ...

message or an error message. Error messages and recovery are described near the end of this chapter.

By default, EXEC sends all messages to the system console — the console connected to the master CLI or its son. But if you have logged on as OP on a different console, EXEC will acknowledge your commands on that console.

EXEC always displays messages about user MOUNT requests. But you can make other EXEC messages brief or verbose, or you can silence EXEC, with the EXEC BRIEF, VERBOSE, or SILENCE commands. SILENCE is especially useful if your system console is a hardcopy console.

If EXEC logging is on (EXEC LOGGING command), copies of all messages that you see on the system console will be sent to EXEC's log file.

Often-used EXEC Commands

Table 8-2 shows the most popular EXEC commands, alphabetically. In this table, and thoughout this chapter, we assume that macro CX.CLI exists in directory :UTIL. This macro allows operators to type CX cmd↵ instead of CONTROL @EXEC cmd↵.

Table 8-2. Often-used EXEC Commands

EXEC Command	Explanation	Example
ALIGN	Pauses EXEC's printer-managing process (XLPT); or continues process XLPT. Useful when someone wants to align the paper in the printer.	) CX ALIGN @LPB ↵) CX ALIGN/CONT @LPB ↵
CANCEL	Cancels a batch or printing request that is enqueued but not yet active. To stop an active request, use FLUSH.	) CX CANCEL 445 ↵
CONTINUE	Continues (resumes processing in) a batch stream (1-4) or device queue. Useful after EXEC startup (within UP.CLI macro) or after a stream or device has been paused.	) CX CONTINUE 2 ↵) CX CONTINUE @LPB ↵
DISABLE	Disables one or more user consoles for user log on. Useful when you plan to shut down EXEC, and don't want users logging on; also useful when you wish to free one or more EXEC-enabled consoles for use by another program, like DG/SNA or TPMS.	) CX DISABLE @CON8 ↵
ENABLE	Enables one or all user consoles for user logon via EXEC, primarily used within UP.CLI macro.	) CX ENABLE @CON45 ↵) CX ENABLE/ALL ↵
FLUSH	Flushes (kills) the job that a batch stream or device queue is processing. To cancel a request *before* it becomes active, use CANCEL.	) CX FLUSH 3 ↵
FORMS	Allows you to specify a file to be used for special form printing; e.g., bills.	) CX PAUSE @LPB ↵) CX FORMS BILLS @LPB ↵) CX CONT @LPB ↵
MOUNTED	In response to user mount request, this tells EXEC that a tape is mounted on a tape unit.	) CX MOUNTED @MTB1 ↵
OPERATOR	Tells EXEC that an operator is on or off duty. Unless the operator has given ON notice to EXEC, users will get error messages from their MOUNT and from batch jobs submitted with the /OPERATOR switch.	) CX OPERATOR ON ↵ ...) BROADCAST Lunchtime ↵) CX OPERATOR OFF ↵ ...(lunch)) BROADCAST Mounts ok ↵) CX OPER ON ↵
PAUSE	Pauses a batch stream, device queue, or device in an orderly way, after current request is done. It prepares for normal EXEC shutdown or change in device specifications. Later, CONTINUE commands in UP.CLI or typed on the system console will continue normal processing.	) CX PAUSE ↵) CX PAUSE @LPB ↵

(continues)

Table 8-2. Often-used EXEC Commands

EXEC Command	Explanation	Example
SILENCE	Tells EXEC to keep quiet; it suppresses batch or device queue messages to the system console and EXEC log file, if active. This is useful on hardcopy system consoles.	) CX SILENCE ↵) CX SILENCE @LPB ↵
START	Associates a queue and device with an EXEC co-operative process. The CONTINUE command then activates the device. This command is usually issued by the UP.CLI macro.	) CX START LPT @LPB ↵) CX CONT @LPB ↵
STATUS, SPOOLSTATUS	Displays information about batch or spool queues. CLI command QDISPLAY/V is also handy for this.	) CX STATUS ↵) CX SPOOLSTATUS ↵
XHELP	Displays information about all EXEC commands or any command; omit the leading CX.	) XHELP ↵) XHELP ALIGN ↵

(concluded)

EXEC Commands by Function

EXEC manages four types of function: general, user log on, batch and spooling, and user mount requests. Table 8-3 shows all EXEC commands, arranged by function. The commands are described in alphabetical order, later in this chapter.

Table 8-3. EXEC Commands by Function

Function Category	EXEC Command	Brief Description
General	LOGGING MESSAGE PROMPTS XHELP	Turns EXEC logging on or off. Writes a text message to the log file. Adds or removes time of day from EXEC message display. Describes all EXEC commands or a specified command (omit the leading CX).
Logon	CONSOLESTATUS DISABLE ENABLE TERMINATE	Displays console-user status. Disables idle EXEC consoles for user logon. Enables consoles for user logon. Kills the user process associated with a console or a cooperative process associated with a device.
Batch/ Spool	ALIGN BINARY BRIEF CANCEL CLOSE CONTINUE CPL	Stops or continues the line printer (used to align it). Puts a letter-quality printer or line printer in binary mode. Tells EXEC to make batch or spool messages brief. Cancels a request that is waiting in a queue. Closes a device queue to user requests. Continues a paused batch stream or device queue. Changes the maximum number of printed characters per line.

(continues)

Table 8-3. EXEC Commands by Function

Function Category	EXEC Command	Brief Description
Batch/Spool (cont)	CREATE	Creates a device queue.
	DEFAULTFORMS	Sets new printer CPL and LPP parameters.
	DELETE	Deletes a device queue.
	ELONGATE	Turns LP2/TP2 elongated printing on or off.
	EVEN	Turns even pagination on or off for a printer.
	FLUSH	Flushes (kills) an active batch or device request.
	FORMS	Specifies a file to be used for printer format control.
	HEADERS	Changes the number of printed header pages.
	HOLD	Holds (suspends) a batch or spool request.
	LIMIT	Enforces user- or operator-defined limits on CPU time (batch) or printed pages (spool).
	LPP	Changes the maximum number of printed lines per page.
	OPEN	Opens a device queue to user requests.
	PAUSE	Pauses one or more batch streams or devices.
	PURGE	Deletes entries in a stopped device queue.
	PRIORITY	Sets a new priority and/or type for batch streams and co-operative processes (XLPT).
	QPRIORITY	Sets a new range of batch/print priorities that will be accepted by a batch stream or device.
	RESTART	Restarts device queues after abnormal EXEC termination (batch restarts automatically).
	SILENCE	Suppresses all EXEC messages except user mount requests.
	SPOOLSTATUS	Displays queue-device association and status.
	START	Associates a queue and device with an EXEC process.
	STATUS	Displays batch steam or device status.
	STOP	Dissociates a device from a queue (opposite of START).
	TRAILERS	Changes number of printed trailer pages.
	UNHOLD	Negates HOLD command.
	UNLIMIT	Negates LIMIT command.
	UNSILENCE	Negates SILENCE command.
	VERBOSE	Tells EXEC to make batch/spool messages verbose.
	XBIAS	Sets a new EXEC small job versus large job bias factor.
Mount	DISMOUNTED	Tells EXEC that a person has physically dismounted a tape.
	MOUNTED	Tells EXEC that a person has physically mounted a tape on a unit.
	MOUNTSTATUS	Displays MOUNT and mount request status.
	PREMOUNT	Tells EXEC that a person has physically mounted a tape before a MOUNT request occurred.
	REFUSED	Tells EXEC and a user that you refused the user's MOUNT request
	UNITSTATUS	Describes the mount status of tape units.

(concluded)

About User Log On

Users cannot log on under EXEC unless they have valid user profiles, created with PREDITOR. To log on, a person presses NEW LINE on an enabled console; EXEC then prompts for a username and password. When — only when — the person types a valid username and password, EXEC tells AOS to create a user process for that person and logs the user on. The operating system uses the parameters in the profile when it creates the user process, and it enforces these thereafter.

Parameters in the profile include initial IPC file, program to be executed for the user, number of sons, Super-privilege specifications, and the amount of disk space allowed, among other things. All these issues are covered in the previous chapter.

The initial IPC file is often in the user's initial directory, where the user can edit it as desired. Another file — LOGON.MESSAGE — is in :UTIL; EXEC automatically displays this file (first 512 characters) on each user's console at log on. Examples of creating initial IPC files and LOGON.MESSAGE are given in Chapter 5.

While the user process runs, EXEC records console use, pages printed, device time used for MOUNT requests, and privileged-user logons. It places this information in the system SYSLOG file if system logging is on. SYSLOG is different from EXEC's own log file.

When a user terminates his/her initial program (as by typing BYE) to the CLI), EXEC terminates the user process; the console is then free for other users. Naturally, a privileged process like the master CLI or EXEC can terminate the user process whenever it wants, but this is unpleasant for the user and should be done only if necessary.

Logon Errors

EXEC expects the user profile and user directory created by PREDITOR to be intact. If the user directory was somehow deleted, or its ACL changed, or if the profile was tampered with, the user may receive an error message and be advised to call the system manager.

Should this problem occur, run PREDITOR on the profile of this username. PREDITOR will then automatically rectify file problems. (But if the directory was inadvertently deleted, the user's files will need to be reloaded from backup media.)

Logon Commands

EXEC logon commands are CONSOLE STATUS, DISABLE, ENABLE, and TERMINATE. These are summarized in Table 8-3 (earlier), and detailed in the alphabetical section, later in this chapter.

About Batch and Spooling

This section gives some background on batch and spooling.

Batch Processing

When someone wants to run a program, he/she can run it interactively from a console, or in batch.

Running a program interactively is fast, but makes the console unavailable until the program ends. Also, if many processes are running interactively, system performance may suffer. Users run programs interactively via XEQ or PROCESS commands to their CLIs, or via macros that contain XEQ or PROCESS.

Running a program in batch may take a little more time; but the console remains available. Also, the system may run more efficiently. Anyone with a user profile and the USE BATCH privilege can use batch. An easy way to do it is to type the QBATCH command, followed by the desired XEQ or PROCESS command, to the CLI. An easier way — which eliminates walking to and from the line printer — is to use the BATCH and CHEK macros described in Chapter 5.

Another way to use batch, if a user's job is on punched cards, is to give the card deck(s) to the system operator for processing.

Batch Details

Each time EXEC comes up, it creates three batch queues (if they do not already exist): BATCH_INPUT, BATCH_LIST, and BATCH_OUTPUT. These are the input, list, and output files for batch requests.

For each job a user submits, the CLI creates an input file with the needed commands in the user's directory; the filename ends in .JOB. Then the CLI notifies EXEC and EXEC places the request in the BATCH_INPUT queue (in directory QUEUE) under the full input file pathname.

The job comes up for processing. EXEC creates temporary output and list files, with pathnames :QUEUE:username.OUTPUT.seqno and :QUEUE:username.LIST.seqno, where seqno is the sequence number. Then, EXEC creates a CLI process to run the job; and the process runs, taking its commands from the .JOB file in the user's directory. If any program involved in the job writes to the console (as with a CLI WRITE command, error condition, or BASIC or FORTRAN PRINT statement), the text goes to the temporary output file. The listing file text (if the user specified any) goes to the temporary list file. When the job finishes, EXEC prints and deletes the output file and list file (if any). Finally — unless an error aborted the batch job — EXEC deletes the .JOB input file from the user's directory.

Instead of EXEC's temporary output and list files, users can specify other files with QBATCH switches, if they want. All files written by the job (except EXEC's temporaries) go to the same directory as they would have if the job had been run interactively.

To handle requests in the batch input file, EXEC has up to four batch *streams*. As each request comes up, EXEC assigns it as a job to an available stream. If more than one batch stream is running, then more than one batch job can run at a time. An EXEC CONTINUE command is needed to activate each stream; these are usually part of the UP.CLI macro. The default UP.CLI macro continues only stream 1, but you can edit UP.CLI to continue more streams if you want. Or, you can type CX CONTINUE commands to run other streams. You can also pause any batch stream using EXEC's PAUSE command.

EXEC processes batch requests according to their queue sequence number and queue priority. Requests with the highest priority (closest to 0) are processed first. Requests with the same priority are processed on a first-in, first-out basis.

By default, all batch streams accept requests of any priority. You can assign each stream a specific priority range using the CX QPRIORITY command. *Users* can assign priority to batch requests with QBATCH/QSUBMIT switches. If you set a priority range for a stream, the stream will accept only those requests with priorities in your specified range. You can use the CX XBIAS command to control apparent request priorities.

An example will help show how batch works. Assume that user Jack types

```
) QBATCH  XEQ  MASM/L  MY_PROGRAM ↵
QUEUED, SEQ=454
)
```

This sets up a macroassembly to run in batch (a compile would be similar). The CLI creates a temporary input file ending in .JOB in Jack's working directory; and it enqueues the request as sequence number 454 in the BATCH_INPUT queue. When the request is ready to run, EXEC creates a user process to run it. The new process gets input from the .JOB input file. The assembly occurs. The batch output file (:QUEUE.JACK.OUTPUT.454) shows the commands used and assembly errors (if any). The list file, produced by the MASM/L switch, shows the assembly listing. All three files — input, output, and list — are deleted after the assembly.

The system places the object file from the assembly in the directory from which Jack typed the QBATCH command. The end result is that, with a little extra time, Jack got the same result as from an interactive assembly — except that assembly errors were printed instead of being displayed on his console. By using the BATCH and CHEK macros (Chapter 5), he could have checked for assembly errors without going to the printer.

For batch processing to occur, EXEC must be running and at least one batch stream must have been continued. Also, the batch output and list queues must have been started on the line printer spool queue, described below. Every batch user process, like every console user process, is subordinate to EXEC.

Batch Jobs in Stacked Format

Batch jobs submitted to a card reader must be in stacked format. Job control cards must precede and follow each user's card deck, as follows.

card 1	$$JOB*[/switches]* username
card 2	$$PASSWORD password
cards 3-n	commands to CLI that make up job
card n+1	$$END
.	.
card m	end of file card (all holes in first column punched). Users don't insert this; you (the operator) should insert it after the last user's $$END card.

No CLI commands are involved in card processing. The system operator issues the EXEC command CX STACK, then the operator, or user, places the user's card deck in the card reader. (The STACK command is described in this section of the chapter.)

Then, EXEC creates a co-operative process that runs program STACKER.PR. The Stacker reads all cards from the one following the password up to, but not including, the end card, into a temporary disk file in directory :QUEUE. When it reads the end card, the Stacker enqueues the disk file it just created to run as a batch request. After the the job runs, EXEC deletes the temporary file. EXEC also deletes this file if the batch job aborts.

The Stacker can process multiple jobs with one STACK command if the operator removes individual user END OF FILE cards between the user card decks. There should be one END OF FILE card at the end of the last job to stop the STACKER co-operative process.

No card may have $$ in its first two columns except the $$JOB, $$PASSWORD, and $$END cards. If a job contains any other $$ entry, EXEC will not queue or process the job. An appendix in the CLI manual gives details on submitting jobs to the card reader.

Spooled Devices

In the multiuser environment, processes compete for the use of slow input or output devices, like line printers and plotters. Processes can send and receive data faster than devices; and EXEC ensures that this data moves in an orderly way. When a process wants to use a device, EXEC enqueues the process's request and writes the data involved to disk. Then when the request comes up, EXEC directs AOS to send it to the device.

Storing information temporarily on disk for processing at system discretion is called *spooling*. EXEC provides spooling to all slow I/O devices. The EXEC program XLPT, for example, controls spooling to any printing device.

Spooled devices include

- Card readers;
- Line printers;
- Digital plotters;
- Hard-copy consoles;
- Asynchronous communication lines connected via modems to remote consoles;
- Asynchronous communication lines connected to other systems in XODIAC networks;
- Synchronous communication lines working though a process like HASP II (HAMLET), DG/SNA, or XODIAC.

EXEC provides the needed program files for the card reader, line printers, local and modem-connected consoles, and digital plotter.

Spool Queues

EXEC places all user requests that involve spooled devices in spool queues. These requests can come from batch jobs or directly from users through the CLI commands QBATCH, QPRINT, QPLOT, QSUBMIT, QFTA, and QSNA.

As in batch, EXEC spool queues process requests according to priority and sequence number. Those with the lowest priority number have the highest priority and are processed first; those with the same priority are processed on a first-in, first-out basis. Many users can issue Q-series commands simultaneously. You can use the EXEC XBIAS command to change the apparent priority of spool queue jobs.

As in batch, EXEC stores each spool queue request as a temporary file in directory QUEUE; and it deletes the temporary file after the spooled device has processed the request.

Also as in batch, an operator process can hold or cancel a spool queue entry via EXEC HOLD and CANCEL commands; or it can FLUSH an active entry. Users can hold or cancel their own queue entries using the CLI commands QHOLD and QCANCEL. They can use QCANCEL (not FLUSH) to cancel an entry being processed.

Creating and Opening Queues

Before users can access a spool queue, the queue must be created, opened, and started on a device; then the device must be continued. The EXEC commands CREATE, OPEN, START, and CONTINUE do these things, as follows

`CREATE type queuename`	Creates the queuename of type.
`OPEN queuename`	Opens queuename as an entry in the peripherals directory.
`START queuename device`	Starts the queue and device on a co-operative process.
`CONTINUE device`	Activates the device.

If you created your own multiuser environment in Chapter 5, you typed these commands to create line printer and perhaps digital plotter queues. You need not recreate these queues; but whenever you want to bring the multiuser environment up, all needed queues must be started and continued. The UP.CLI macro should do this for you; if not, you will probably want to edit UP.CLI so that it *does* start and continue all needed queues. The remainder of this section, and the next, will give you the background you need.

The CLI command QPRINT accesses the queuename LPT, and the QPLOT command accesses the queuename PLT. If a print or plot queue has a name other than these, users must use the /QUEUE switch to access the queue.

If your system has one or more letter-quality printers, you can create queues named LQP, LQP1, and so on for these. (If you have the CEO software on your system, the CEO.PRINTER macro issues all of these commands for you.) For example, for a letter-quality printer on console line 13 (@CON15), you might type

```
) CX  CREATE  PRINT  LQP ↵
) CX  OPEN  LQP ↵
) CX  START  LQP  @CON15 ↵
) CX  HEADERS  @CON15  0 ↵
) CX  BINARY  @CON15  CLEANUP_FILE ↵
) CX  CONTINUE  @CON15 ↵
```

BINARY mode is needed for CEO users. CLI users can access the letter-quality printer via QPRINT/QUEUE=LQP... commands.

If a line printer is uppercase only, it must be started with the UPPER argument. If it is laser document printer (type LPD), it must be started with the /NL switch. Both UPPER and /NL are described under EXEC's START command.

If your system has a third or fourth line printer or plotter, you will need to create a queue name for each. For example, assume that you have a third line printer and want to create a queue — say LPT2 — for it. You'd use the following commands.

```
) CX  CREATE  PRINT  LPT2 ↵
) CX  OPEN  LPT2 ↵
) CX  START  LPT2  @LPB2 ↵
) CX  CONTINUE  @LPB2 ↵
```

And users could access the queue via QPRINT/QUEUE=LPT2 ... commands.

A single queue can be associated with more than one device; for example

```
) CX  CREATE  PRINT  EITHER ↵
) CX  OPEN  EITHER ↵
) CX  START  EITHER  @LPB ↵
) CX  START  EITHER  @LPB1 ↵
) CX  CONTINUE  @LPB ↵
) CX  CONTINUE  @LPB1 ↵
```

Users could access this queue (which sends output to either printer) via QPRINT/QUEUE=EITHER .... commands.

The device associated with a queue can be a disk file. For example, assume that you wanted to divert LPT queue output from your first line printer to a disk file. You'd create the disk file in :PER, stop the printer, start the old queue on the disk file, then continue the disk file, just as if it were a device. For example:

```
) SUPERUSER  ON ↵
*) CREATE  :PER:LPTFILE ↵
*) CX  STOP  @LPB ↵
*) CX  START  LPT  @LPTFILE ↵
*) CX  CONTINUE  @LPTFILE ↵
```

Now all QPRINT output will go to file @LPTFILE.

A queuename cannot be the same as its associated device name (because, as you can see, both are entries in :PER).

Any user can check the spool queues by typing the CLI command QDISPLAY. The operator can also use QDISPLAY, or the EXEC command SPOOLSTATUS.

Normally, the commands to start queues and continue spooled devices are in the UP.CLI macro — and the queue-device mechanism runs by itself. But knowing about them is important if you need to reorganize queues while EXEC is running.

EXEC allows up to 32 queues, including the three permanent batch queues. It allows up to 256 total queue entries at one time. If a user tries to submit an entry when there are 256 entries outstanding, he/she will receive an error message.

Communications and Network Queues

If you have certain optional DG communications or network products, you will probably need to create EXEC queues for them.

DG's IBM emulators — DG/SNA and HASP II (HAMLET) — and XODIAC networking agent FTA need EXEC queues to run properly. The name of the HASP II queue should be HAMQ; the name of the SNA_RJE queue should be SNQ, and the name of the FTA queue should be FTQ.

To *create and open* the HASP II queue, you'd type

```
) CX  CREATE  HAMLET  HAMQ ↵
) CX  OPEN  HAMQ ↵
```

Then you'd start and continue the queue according to the HAMLET documentation (or Release Notice). You might choose to start the HAMLET process, and put the EXEC START and CONTINUE commands in the bisync up macro (mentioned with the GSMGR process in Chapter 5; for HAMLET to work, GSMGR must be running). After the HAMQ is continued, users can access the queue via QSUBMIT/QUEUE=HAMQ... commands.

To *create and open* the DG/SNA (SNA_RJE) queue, you'd type

```
) CX  CREATE  SNA  SNQ ↵
) CX  OPEN  SNQ ↵
```

When you start and continue the SNQ, it must be associated with the SNA_RJE *process* name (not a device name). So the SNA_RJE process must be running. The sequence of commands goes as follows:

CONTROL @SNA_RJE START ...	CONTROL command creates SNA_RJE process.
CX START SNQ @SNA_RJE_EXEC	Start queue on process name (the "_EXEC" suffix was added by the process)
CX CONTINUE SNQ	Continue the queue.

Users can now access the SNA_RJE queue using the CLI command QSNA.

Later, to pause the queue, you'd pause the process name: @SNA_RJE_EXEC. You could then kill the process with a CONTROL @SNA_RJE STOP command. For convenience, you might choose to put the SNA-creating process and EXEC commands in an SNA up macro; and put the terminating commands in an SNA down macro.

For XODIAC FTA, to *create and open* the FTA queue, you'd type

```
) CX  CREATE  FTA  FTQ ↵
) CX  OPEN  FTQ ↵
```

EXEC commands to start and continue the FTQ on the FTA process are included in the XODIAC macro UP.NETWORK.CLI; and there are commands to terminate the FTA process in DOWN.NETWORK.CLI.

When the FTA process is up and running, and the queue FTQ has been started on it, users on the network can enqueue file transfers with the CLI command QFTA.

Communications and networking products have other, non-EXEC queues, CONTROL @name commands, and program files, described in their own manuals. We describe the queues above in *this* book because they are EXEC queues.

Spool Queue Parameters

Sometimes, you may want to change spool queue parameters like priority, lines per page, or forms. To do so, pause the device associated with the queue before changing the parameter. Then make the change and continue the device. For example, to change LPT's queue priority range, type

```
) CX   PAUSE   @LPB ↓
) CX   QPRIORITY   @LPB   127   255 ↓
) CX   CONTINUE   @LPB ↓
```

These commands tell EXEC to pause the print device named @LPB, to change the priority of the device to a high-priority value of 127 and a low-priority value of 255, and to continue the device. During a pause, users are allowed to submit jobs to the queue; however, EXEC does not instruct the device to process the jobs until you continue the device.

This is the kind of thing you might do at runtime, instead of in the UP.CLI macro.

The Queue Compression Program

EXEC stores all the temporary files used in batch and spooling in directory :QUEUE; then it deletes them. It stores the entry *names* in file Q, which expands to hold each new entry name.

The queue compression program, QCMP, is a utility that repacks the queue file and can delete unused files in :QUEUE. QCMP can delete all files/directories that don't have entries in the queue file — *so don't let anyone use directory :QUEUE for file storage.*

QCMP cannot run while EXEC is running; if you try to run it while EXEC is up you'll get a *CAN'T OPEN* error message. Usually, QCMP is executed via the UP.CLI macro before EXEC is brought up. But you can execute QCMP while EXEC isn't running via the form

XEQ   QCMP*[/YES][/L[=pathname]]*

The /YES switch tells QCMP to delete unused files; otherwise QCMP asks for confirmation before it deletes the files. */L[=pathname]* sends output to the @LIST file, or the file named in pathname if you include *=pathname*. If you omit /L, output goes to the system console.

QCMP repacks the queue file. Then, if you omitted the /YES switch, it asks

MAY I DELETE UNUSED FILES IN :QUEUE?

Type Y ↓ or YES ↓ if you want to delete unused files; type N↓ or NO ↓ if you do not. If you type Y↓ or used the /YES switch, QCMP will say either

NO UNUSED FILES FOUND.

or

DELETED FILES:

...
...(filenames deleted)...

Note that QCMP deletes user files in :QUEUE.

If you type N ↓ to the QCMP query, QCMP says

I WOULD HAVE DELETED:

...
... (filenames)...

When QCMP finishes, it says

REPACKING COMPLETE, n BLOCKS FREED

The *n* is the number of disk blocks reclaimed from :QUEUE.

FORMS Directory

If your installation prints data on special forms, you need a directory called :UTIL:FORMS. Chapter 5 had the reader create this directory. You can check it, and its access control list (ACL) by typing

```
) FILES/AS  :UTIL:FORMS ↵
:UTIL FORMS DIR date size
) ACL/V  :UTIL:FORMS ↵
FORMS OP,OWARE +,RE
)
```

The ACL of OP,OWARE +,RE gives the operator all access and users read and execute access to :UTIL:FORMS. Read and execute access lets users read the names of the form types in the FORMS directory and use them. If you give users append access, they can add their own forms. If you give users owner or write access, they can delete forms or change the ACLs of current forms. To give them append access, you'd type

```
) ACL/V  :UTIL:FORMS  OP,OWARE  +,REA ↵
:UTIL:FORMS
)
```

To create and use forms control files in FORMS, either you or a user must

- Get into directory FORMS (DIR :UTIL:FORMS↵).
- Create a forms file with a filename; e.g., PAYCHECKS. If needed, change the ACL for the finished forms file so that pertinent users can access it with QPRINT/FORMS= commands.
- Execute the Forms Control Utility program (XEQ FCU↵). This is an interactive program that assigns printer format specifications to existing files; it has a help feature (H). Give FCU the C command and the new filename, e.g., PAYCHECKS. The FCU dialog is described in the CLI manual.
- Test the form by pausing EXEC, putting the paper form in the printer, typing the EXEC FORMS command as described later in this chapter, continuing the printer. Then type QPRINT/FORMS=formname datafilename↵.
- If the form prints as desired, you're done. If not, edit the form file with FCU and try the QPRINT command again. Repeat these steps until you're satisfied with the printing.

For example, suppose your installation has three-part paper forms, available to all users, and check forms only for the users with usernames PAYROLL and ACCOUNTING. To create these forms, you might do the following.

- Create the files and ACLs by typing

```
) PUSH ↵
) DIR  :UTIL:FORMS ↵
) CREATE  3_PART_FORMS ↵
) ACL  3_PART_FORMS  OP,OWARE  +,R ↵
) CREATE  PAYCHECKS ↵
) ACL  PAYCHECKS  OP,OWARE  ACCOUNTING,R  PAYROLL,R ↵
) POP ↵
)
```

(PUSH and POP save the old environment and restore it — returning you to the original directory after the operation.)

- Insert format specs in the two form files with FCU.
- Pause EXEC; issue EXEC FORMS command; put forms in printer and test as described above.

When users want output printed on a particular form, they include the /FORMS=formname switch with the QPRINT command. EXEC holds these jobs until you insert the specified form in the appropriate printer and issue the EXEC FORMS command. Printing requests without the /FORMS= switch are enqueued as usual until you issue the EXEC FORMS command.

Batch and Spool Commands to EXEC

Many EXEC batch and spool commands relate to both batch and spooling. The commands are

ALIGN, BINARY, BRIEF, CANCEL, CLOSE, CONTINUE, CPL, CREATE, DEFAULTFORMS, DELETE, ELONGATE, EVEN, FLUSH, FORMS, HEADERS, HOLD, LIMIT, LPP, OPEN, PAUSE, PRIORITY, PURGE, QPRIORITY, RESTART, SILENCE, SPOOLSTATUS, STACK,START, STATUS, STOP, TRAILERS, UNHOLD, UNLIMIT, UNSILENCE, VERBOSE, and XBIAS.

These are summarized in Table 8-3 (earlier) and detailed in the alphabetical section, later in this chapter.

About User Tape Mount Requests

This section describes user tape mount requests, tape labeling, and the use of labeled tapes. The related EXEC *commands* appear in the next section.

User mount requests include a broad range of system operations. Often, timesharing users will want to have the system operator/manager (you) mount tapes for them. Also — when you dump files for backup — you will probably want to log on *as a user* so that EXEC's MOUNT and batch features will be available to you. In either case, you will want to know something about labeled tapes.

User tape mount requests are often used for system backup — generally by the system operator *acting* as a user. If your system does not have a tape unit, tape mount requests are irrelevant to you. For backup, you'll use labeled diskettes, which are managed by the CLI, not EXEC. For backup on diskettes, see "Labeled Diskettes" in Chapter 9.

Tape Labeling

Magnetic tapes are either *labeled* or *unlabeled*. An unlabeled tape contains no label information. A labeled tape has information — including a volume ID (volid) for the tape, and a filename and expiration date for the tape fileset.

There are several different kinds of labels on a labeled tape. The most important labels are the volume header label (created by the LABEL program), and the first file header label (HDR1, created by the system when it first writes to the tape). Generally, the other labels are useful only if user application programs specifically read and write them.

The information on a labeled tape includes the

- volume ID (volid). This is a one- to six-character name assigned to the tape volume via the LABEL program. It's stored in the volume header label at the beginning of the tape.
- filename. This is a 1- to 17-character name, assigned when the tape set is written to, and stored in the HDR1 label. For example, if you type DUMP/V TAPE:FILE5 #), the filename is FILE5.
- fileset ID and sequence number. These are generated and written by the system when the tape is written to, and stored in the HDR1 label.
- creation and expiration date, written by the system when the tape is written to. The default retention period is 90 days from the tape write. The DUMP command has a /RETAIN switch that allows you to select a different period. This includes /RETAIN=0, which allows the tape set to be written to immediately, without relabeling.

The whole approach to labeled tape differs from the approach to unlabeled tapes. For example, typical I/O to unlabeled and labeled tape might look like this.

Unlabeled Tape	Labeled Tape
DIR :	MOUNT/VOLID=xxx TAPE PLEASE
DUMP @MTB0:0 +	DIR :

DUMP @MTB0:1 UDD:SALLY:#	DUMP :UDD:OP:TAPE:USERS UDD:#
DUMP @MTB0:2 UDD:JACK:#	
.	.
.	.
.	.
(LOAD sequence would be the same, with LOAD instead of DUMP).	(LOAD sequence would be the same, with LOAD instead of DUMP.)

The labeled approach won't work on unlabeled tapes. But the unlabeled approach *can* work on labeled tapes, because the tape labels are individual tape files.

Although labeling tapes requires extra effort, labeled tapes offer many advantages. These include the the following:

- Users can create multivolume tape files without worrying about the physical ends of the tapes. One file can extend across many tape volumes.
- The tape itself contains information about material stored on it: user label text, filename, and expiration date.
- Users can create tapes to be read on other operating systems, like an IBM system or a system that uses ANSI-standard labeled tapes.
- Users have control over the amount of time each tape file will be retained. The system will not overwrite files dumped to a labeled tape until the retention period has passed or until the tape is relabeled.
- DG data management programs — INFOS II and DG/DBMS — have logging utilities that expect labeled tapes.

Planning and Assigning Tape Labels

Generally, tape labeling must be consistent to be worthwhile. There's a big difference between typing

1. DUMP/V @MTB0:3 +.F77 ⤶

   and

2. DUMP/V MYTAPE:F77_SOURCES +.F77 ⤶

The first approach is hardware oriented; people must know the unit number and tape file number they want. The second approach is software oriented; people can use any linkname and a real filename up to 17 characters instead of the unit and file number. But if people try to use both the labeled and unlabeled approaches, they may get confused.

This means that the use of labels must be planned, usually by the system manager, administrator, operator, or someone in authority. You may not want to let users create their own labels, because then you will have to keep track of their labels and tapes. If your installation lets users do their own labeling, there should be a standard for the labels — and you, the operator, will have to administer a filing system for the user-labeled tapes so that you can find them easily. If users can't create their own labels, you may want to prepare a stockpile of blank, labeled tapes for them. You can do this with the LABEL utility.

The industry standard length of labeled tape volume IDs *(volids)* is 6 uppercase alphanumeric characters. So the volids chosen should be 6 characters or less, yet be as descriptive as possible. You might choose an arrangement in which the leading characters specify the *usage type* for the tape, and the trailing characters are numbers that give the tape sequence. For example,

volids GP0000 through GP9999 are for General Purpose (GP) user use;

volids DB0000 through DB9999 are for DataBase (DB) archiving.

Or, you could use the date and sequential numbers; e.g., 130100, 130101, etc. for 13 January. Or, if you alternate daily dumps, use a suffix A or B. If you have multiple AOS systems, you may want to include a short system identifier.

After deciding on the volids, you can use the CLI LABEL utility to prepare prelabeled tapes for each use type. For example,

Put a new tape on unit MTB0.

```
) X   LABEL/UVL=1ST_USER_VOLUME   @MTB0   GP0000 ⤶
```

Dismount tape and put another new tape on unit.

```
) X   LABEL/UVL=2ND_USER_VOLUME   @MTB0   GP0001 ⤶
```

and so on. A tape labeled with the LABEL program has the label at the beginning, but users can ignore it: users write to the tape and read from it by linkname and filename. The tape label includes the volume ID, UVL text (if any), and system name and revision. If the tape has been written to, the label also contains the expiration date and fileset ID. Labels written by LABEL are in ASCII, so you can use the CLI command TYPE to read them; e.g., TYPE @MTB0 ⤶. (All LABEL utility syntax and switches are further described in the CLI manual.)

After using LABEL to label a tape, you may want to write the label name on a paper tape label and stick it on the tape reel. Later, when someone needs a blank tape, you can get the next sequential reel and mount it on a unit, then tell the person what the volid is with the SEND command. After the person is done with the tape, you can file the tape with the person's name so that you can find it easily when desired.

After a labeled tape has been written to, the system won't allow the material on it to be overwritten until the retention period (default 90 days) has passed. The DUMP command has a /RETAIN= switch that can select a different number of days. For example, if a user types

```
) DUMP/V/RETAIN=14   TAPE:SOURCES   +.PL1 ⤶
```

then the system won't overwrite file SOURCES until 14 days have passed. However, anyone can overwrite the *label* (with new retention period and other information) by relabeling the tape with the LABEL program.

This gives people great control over backup periods and can be very helpful for file backup.

User MOUNT and DISMOUNT Requests

A *user* — in this context — means a user process, run for anyone with a valid profile. The system operator can issue a "user mount request" if logged onto a user console. A user mount request won't work if it originates at the system console, or any console not "owned" by EXEC.

A user mount request must *come from* a user process — but EXEC will honor it only if someone is available to mount tape. So, someone must have typed CX OPERATOR ON ⤶ at the system console.

To request a tape mount explicitly, a user issues the CLI command MOUNT. One of the arguments to MOUNT is the filename that will become the *linkname.* EXEC then displays a mount message on the system console and you must respond to it. The user can't issue any more commands until you do respond. You can either physically mount the tape and tell EXEC where you mounted it; or you can respond. You can either physically mount the tape and tell EXEC where you mounted it; or you can refuse the request. If you mount the tape, EXEC gives the user process exclusive access to it, and creates the linkname in the user's initial directory. The user can access the tape with the linkname. When the user's tape read or write is done, or on an error, or on a user DISMOUNT command, EXEC deletes the linkname file, rewinds the tape, prints a *DISMOUNT* prompt at the system console, and restores the old tape ACL; you must physically dismount the tape and tell EXEC you dismounted it.

EXEC creates, and later deletes, the user-specified linkname in the user's *initial working directory*. This means that the user must access the tape from the initial directory, or use a pathname to the link (e.g., :UDD:JACK:MYTAPE) in all references to the tape. Otherwise, from outside the initial working directory, the user will get a *DIRECTORY DOES NOT EXIST* or *FILE DOES NOT EXIST* error message. We say all this to help you resolve user confusion: if you don't issue mount requests, you may never see a user linkname.

Unlabeled Tape MOUNT Example

A sequence for an unlabeled tape might go like this:

1. User JACK has logged onto AOS under EXEC. On his console, he types

) MOUNT MYTAPE Please mount a scratch tape -- ring in. ↵

   MYTAPE is the linkname; it could be any legal filename.

2. EXEC responds to JACK's command by displaying the following message to you on the system console:

   *FROM PID n (EXEC): ** UNIT MOUNT ***
   FROM PID n : MID=n USER=JACK PID=n
   FROM PID n : REQUEST IS 'Please mount a scratch tape -- ring in'.
   FROM PID n : RESPOND: CONTROL @EXEC MOUNTED @UNITNAME
   FROM PID n : OR: CONTROL @EXEC REFUSED

   The *MID* (mount ID) identifies each request; MIDs accumulate as users make requests.

   If you don't want to mount a tape, type CX REFUSED↵; JACK will then get a REFUSED message. You might send a message to JACK's PID explaining why you refused. Otherwise, get a blank tape with ring in, physically mount it on a free unit (say MTB1), and type

) CX MOUNTED @MTB1 ↵

3. The CLI prompt now returns to JACK's console. JACK can access the tape by linkname; e.g., MYTAPE:0 for the first file, MYTAPE:1 for the second. In a program, he could OPEN and read or write to it just like any other file.

   When he is done with it, he types

) DISMOUNT MYTAPE Please file under Jack: Tape 1. Thanks. ↵

4. EXEC responds by displaying the following to you on the system console:

   *FROM PID n : (EXEC) ** UNIT DISMOUNT ***
   FROM PID n : UNIT(S) ARE @MTB1
   FROM PID n : REQUEST IS 'Please file under Jack: Tape 1. Thanks.'
   FROM PID n : RESPOND CONTROL @EXEC DISMOUNTED

   As you can see, EXEC gives pretty clear instructions. (If JACK forgetfully logged off without typing a DISMOUNT command, EXEC would tell you so; but your response would be the same.)

5. Remove the tape from the unit. You may want to file it where you can find it again if JACK asks for it. Then type

) CX DISMOUNTED ↵

Components of Labeled Tape

There are several different kinds of labels written to a labeled tape. The only ones you really *need* are the volume header label that contains the volid (the LABEL utility creates this), and the first file header label, HDR1 (the system creates this when it writes material to the tape). Generally, the other labels are useful only if user application programs specifically read and write them, via the ?OPEN system call or higher level language equivalent.

Figure 8-2 shows a tape's structure after you label it with the LABEL utility and write disk-based material to it.

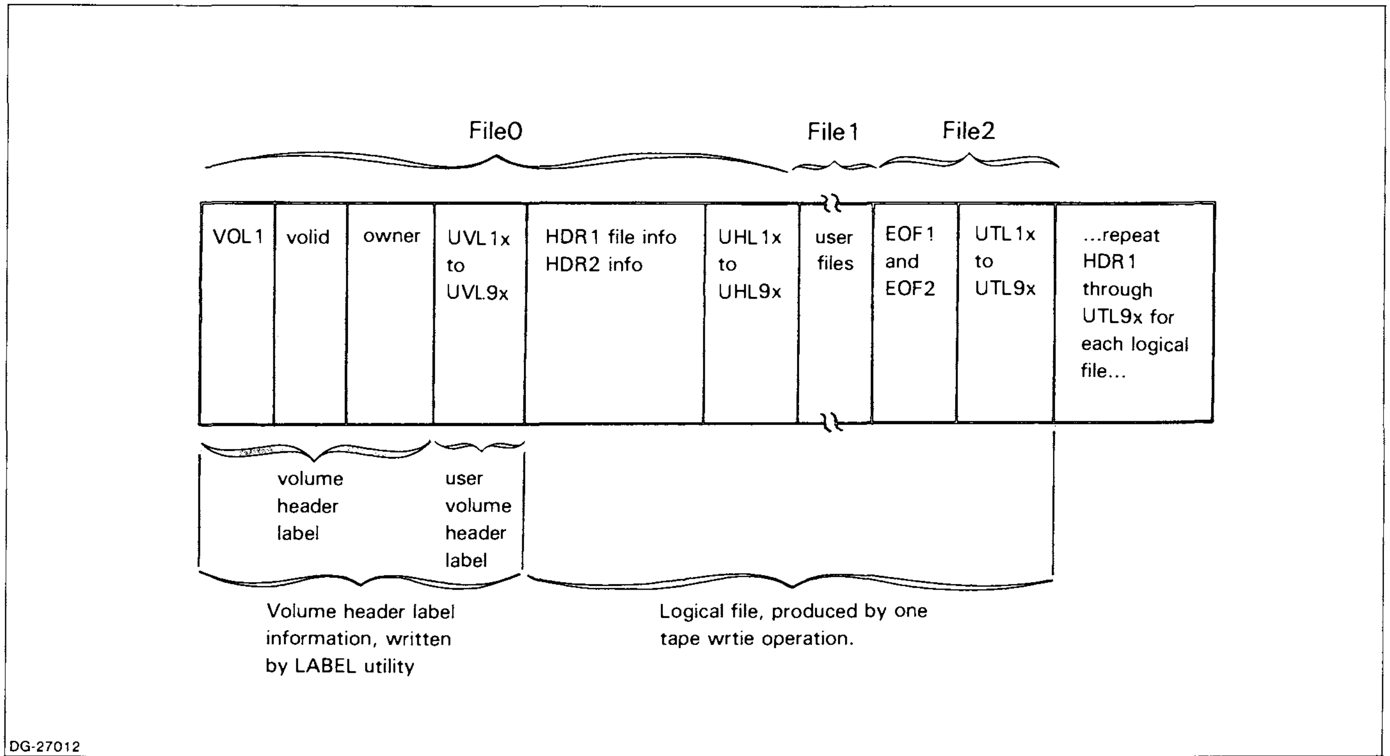


Figure 8-2. Information on a Labeled Tape

The fields shown on the tape in Figure 8-2 have the following meanings.

VOL1 — is a fixed string, meaning "volume type 1." The LABEL utility always writes VOL1 on every tape.

volid — is the volume ID that you assign using the LABEL utility (X LABEL unitname volid ↵). The reserved length is 6 characters; the volid can't exceed 6 characters.

owner — is the owner field, that you can specify with the LABEL program's /OWNER= switch. This label is optional. The reserved length is 14 characters.

UVL1x to UVL9x — are user volume labels you can specify with the LABEL program's /UVL=x switch. On the tape, each UVL field begins with the characters UVLx, followed by the text of x given with the /UVL=x switch. These labels are optional. Generally, they are useful only if your application programs can read them. The maximum length you can specify for a user volume label is 76 characters.

HDR1 file info	is information written by the system during the write operation. It contains the following items. (All items are created and used only by the operating system, unless noted otherwise.) • filename for the fileset, as supplied by the person who started the tape write. (This fileset has no relationship to the individual files actually written to the tape.) For example, the command) DUMP/V TAPE:XFILE ↲ creates the filename XFILE. The maximum filename length is 17 characters. • fileset ID. • file section number. • file sequence number. • generation number and version number. • creation date. • expiration date (the default is 90 days from the creation date; the DUMP command's /RETAIN switch can override the default). • block count (always 0 in HDR1). • operating system ID (as set by SYSID command, or the default).
HDR2 info	In DG format (default), the system always writes this label. In non-DG format, it's written only if a user program opened the tape using an extended packet on the ?OPEN system call. HDR2 contains the record format specifier (a one-letter code), block length (buffer size), and a code for record length.
UHL1x to UHL9x	are user header labels. These are optional. The only way to have them written is via the ?OPEN system call or higher level language equivalent. They are useful only if you have application programs that read and write them.
user files	is the disk-based information copied during the tape write. This information includes all information written by any *single* CLI command or program write statement. For example, if the command is DUMP, the *user files* will include all the files you dump.
EOF1 and EOF2	At the end of each logical file written, the system writes a label with EOF1 and EOF2. It writes an EOV1 and EOV2 label at the end of each reel. (There is no EOV label if the file fits on a single reel.) EOF1 and EOV1 each record the number of blocks written; this number serves as an error check when the tape is read.
UTL1x to UTL9x	are optional user trailer labels, written after the EOV or EOF label.

DG, ANSI, or IBM Format?

By default, labeled tapes are written in DG format. To write labeled tapes that will be read on a DG, ANSI, or IBM (EBCDIC) system, proceed as follows.

- If the destination system is a DG system (AOS, AOS/VS, etc.), execute the LABEL program without the /I switch. To write to the tape, you can use any CLI command or the default format in any write statement.
- If the destination system is an IBM system (EBCDIC), use the LABEL program with the /I switch. To write to the tape, you cannot use CLI commands. For IBM format, the program must open the labeled tape with the ?OPEN call (using the pathname @LMT:volid:filename) and write to the tape with ?WRITE calls that specify field translation. To create EBCDIC format, you may use a higher level language equivalent of ?OPEN/?WRITE.

- If the destination system is an ANSI-based, non-DG system, use LABEL without the /I switch. To write to the tape, you cannot use CLI commands. For ANSI format, the program must open the labeled tape with the ?OPEN call (using the pathname @LMT:volid:filename), or use a higher level language equivalent of ?OPEN/?WRITE to specify ANSI format.

Labeled Tape MOUNT Example

Users request labeled tape mounts by adding the /VOLID= switch to the MOUNT command.

A sequence in which a user writes to a multiple-volume fileset might go like this:

1. User DATABASE is logged onto AOS under EXEC. On her console, she types

   ```
   ) SEND  2  I want to do a dump. Need at least three 2400-ft. blank & ↲
   &) tapes. Please tell me the tape volids to use. ↲
   ```

2. The system console then displays the message that user DATABASE typed — and you find out who typed it:

   FROM PID 8 : I want to do a dump. Need at least three 2400-ft. blank tapes. Please tell me the tape volids to use.

   ```
   ) WHO  8 ↲
   ```
   PID 8 DATABASE 008 :CLI.PR
)

   You note that 2400-ft. tape volids DB0056, DB0057, DB0058, and DB0059 are available. (The extra volume can do no harm). So you type

   ```
   ) SEND  8  Use volids DB0056 DB0057 DB0058 DB0059 ↲
   ```

3. User DATABASE sees the message

   FROM PID n Use volids DB0056 DB0057 DB0058 DB0059.

   and she types

   ```
   ) MOUNT/VOLID=DB0056/VOL=DB0057/VOL=DB0058/VOL=DB0059  & ↲
   &) TAPE  Rings in and HIGH density please. ↲
   ```

4. Now EXEC takes notice of the MOUNT command and displays

   *FROM PID n (EXEC): ** UNIT MOUNT ***
   FROM PID n : MID=n USER=DATABASE PID=8
   FROM PID n : VOLID(S) ARE: DB0056, DB0057, DB0058, DB0059
   FROM PID n : REQUEST IS 'Rings in and HIGH density please.'
   FROM PID n : RESPOND: CONTROL @EXEC MOUNTED @UNITNAME
   FROM PID n : OR: CONTROL @EXEC REFUSED

5. Now you have a choice. If several tape units are free, you might mount the tapes on them, and type the appropriate CX MOUNTED to one and CX PREMOUNT to the others (as shown under the MOUNTED command). But let's assume you decide to mount only one tape, volid DB0056. Mount it on a unit, say MTB0, and type

   ```
   ) CX  MOUNTED  @MTB0 ↲
   ```

6. User DATABASE's CLI prompt now returns to her console and she can start her tape write. She could write to separate tape filenames, but for simplicity let's say she decides to call her tape file DATABASES. She types

) DUMP/V/BUFFER=8192/RETAIN=14 :UDD:DATABASE:TAPE:DATABASES & ⤶
    &) DATAB+ ⤶

   (Just to be safe, DATABASE uses a full pathname to linkname TAPE.) EXEC now checks the tape to see that the volid is the one that the user requested. The volid is correct, so EXEC does not report an error. If the volid were wrong, EXEC would say ***WRONG VOLUME*** on the *system console.*

   User DATABASE's command dumps all disk files that begin with characters DATAB+ to file DATABASES on TAPE. The big buffer size consumes less tape. The number of days to RETAIN the tape file is 14. (In practice, this user would want to batch the dump, but this is an example.)

7. The tape I/O proceeds. When it reaches the end of tape mark on volid DB0056, EXEC starts rewinding the tape and says

   *FROM PID n (EXEC): ** NEXT VOLUME ***
   FROM PID n : MID=n USER=DATABASE PID=8, VOLID=DB0057,UNIT=@MTB0
   FROM PID n : VOLID(S) ARE: DB0056, DB0057, DB0058, DB0059
   FROM PID n : REQUEST IS 'Rings in and HIGH density please.'
   FROM PID n : RESPOND: CONTROL @EXEC MOUNTED @UNITNAME
   FROM PID n : OR: CONTROL @EXEC REFUSED

8. Remove the tape from unit @MTB0. Mount volid DB0057 on @MTB0. Since you're using the same tape unit, @MTB0, you can omit the unit argument and simply type

) CX MOUNTED ⤶

9. Before the I/O proceeds on the new volume, the system checks its volid. If it were wrong (here, if it were not DB0057), EXEC would say ***WRONG VOLUME*** and prompt for the correct volume. But the volid is correct, so EXEC says nothing.

   The I/O now proceeds on volid DB0057. At the end of tape mark on volume DB0057, EXEC says

   *FROM PID n (EXEC): ** NEXT VOLUME ***
   FROM PID n : MID=n USER=DATABASE PID=8, VOLID=DB0058,UNIT=@MTB0
   FROM PID n : VOLID(S) ARE: DB0056, DB0057, DB0058, DB0059
   FROM PID n : REQUEST IS .....

   Once again, you dismount the tape on MTB0, mount a new volume (this time, DB0058) and type CX MOUNTED ⤶.

10. Again the system checks the new volume's volid. It's correct, so I/O proceeds to the third volume.

    Somewhere in the middle of this tape volume, the dump is done. All files that DATABASE specified have been copied to tape. The CLI prompt returns to DATABASE's console. (Her console would have been tied up by the DUMP command, which is why she probably would have used batch.)

    She types

) DISMOUNT TAPE Thanks. ⤶

11. EXEC displays on the system console:

    *FROM PID n : (EXEC) ** UNIT DISMOUNT ***
    FROM PID n : MID=n USER=DATABASE PID=8
    FROM PID n : REQUEST IS 'Thanks'
    FROM PID n : RESPOND CONTROL @EXEC DISMOUNTED

12. Remove the tape from its unit. File the three tapes wherever you file this type of labeled tape. These tapes will remain read-only until 14 days have passed or until you relabel them via the LABEL program. The extra tape, DB0059, can go back into the stockpile of labeled tapes.

13. Then tell EXEC that the unit is free by typing

) CX DISMOUNTED ↵

    If DATABASE (or anyone) wished to reload the entire file on the three volumes, she or he could generally follow the procedure above. The person might need to have SUPERUSER on, and would need to type LOAD/BUFFER=8192... instead of DUMP/BUFFER=8192... . Any linkname could be used, but tape volumes DB0056, DB0057, and DB0058 would be essential.

    Any load recreates the entire directory/file structure, as it was DUMPed, from the working directory. This can lead to duplication and user confusion. Therefore, only sophisticated users, and/or people in authority, should do loads from multiple-volume filesets. But the example makes the point.

Volume ID (volid) Lists

In his or her MOUNT command, a user can specify a list of tape volume IDs (volids) with /VOLID= switches. If /VOLID switches are omitted from MOUNT, tape I/O is restricted to one volume.

On a write (like a dump), a user can specify enough volume IDs to hold the disk-based material, or tell EXEC to extend the volid list, or both. If the amount of material to be written won't fit on the volid(s) specified, and the /EXTEND switch was omitted, EXEC will abort the tape write. Then, sad to say, the write must be restarted from the beginning. To avoid this situation, a user can be generous with the volid list (the system will use only the volids it needs), and/or use the MOUNT /EXTEND switch. If the latter, EXEC will prompt the operator to mount another tape if one is needed.

For example, assume user process SAM issues this command:

```
) MOUNT/VOLID=VOL1/VOLID=VOL2/EXTEND  TAPE  Please extend to VOL3 & ↵
&) etc. if needed ↵
```

Someone at the system console gets EXEC's mount prompt, mounts a tape with volid VOL1, and types CX MOUNTED @MTn ↵.

The SAM process then issues this command:

```
) DUMP/V/L=DUMP_LIST/BUFFER=8192  :UDD:[!USERNAME]:TAPE:FILES  # ↵
```

The dump proceeds through VOL1; EXEC then rewinds the tape and prompts for VOL2. The person at the system console mounts a tape with volid VOL2 and types the CX MOUNTED command. The dump continues through VOL2. If the dump completes on VOL2, EXEC rewinds the tape and prompts for a dismount. But let's assume that — at the end of VOL2 — more material remains to be dumped. EXEC displays on the system console

*** EXPLICIT LABELED MOUNT ** NEXT VOLUME*
MID= n, USER=n, PID=n, EXEC SUB-TREE= n
EXTEND VOLID LIST
REQEUST IS 'Please extend to VOL3 etc. if needed'
UNITS ARE: @MTBn
*CURRENT VOLUME: ****, ALL VOLUME(S): vol1, vol2*
RESPOND: CONTROL @EXEC MOUNTED @UNITNAME VOLID
OR: CONTROL @EXEC REFUSED

Naturally, the person at the system console can refuse the request. But, having come this far, let's assume the person wants to fulfill it. He mounts a blank tape, labels it via X LABEL @MTxn VOL3 ↲ (as user Sam asked), and types CX MOUNTED @MTxn VOL3 ↲. The dump then proceeds on VOL3. If needed, it can continue through VOL4, VOL5, and so on. When all Sam's specified material has been dumped, EXEC issues a ***WAITING TO BE DISMOUNTED*** message at the system console. The person at the system console dismounts the last tape, types CX DISMOUNTED ↲ and the dump is done.

MOUNT/EXTEND has the advantage of allowing the operator to extend tape writes (up to the 128-character volid list limit) — and it has no disadvantages. You may want to encourage people to use it as a matter of course, and/or use it yourself.

Implicit Mount Requests

To help with labeled tape operations, the system maintains a file named @LMT in the peripherals directory. While any labeled tape is mounted, EXEC maintains the user's linkname to file @LMT:first-volid. For example, during DATABASE's dump above, the name TAPE would be linked to file @LMT:VOL1.

Even when a labeled tape is *not* mounted, users can use pathname @LMT:volid to request a labeled tape. This type of "on the fly" request is called an implicit mount request.

For example, assume all tape units are idle. User SACKVILLE types

) LOAD/V @LMT:GP0076:MY_SOURCE.F77 ↲

As with an explicit MOUNT command, the CLI prompt does not return to the user's console until you — the operator — take action.

After SACKVILLE's LOAD command, EXEC displays on the system console:

*FROM PID n (EXEC): ** UNIT MOUNT ***
FROM PID n : MID=n ,USER=SACKVILLE,PID=n, VOLID=GP0076
FROM PID n : RESPOND: CONTROL @EXEC MOUNTED @UNITNAME
FROM PID n : OR: CONTROL @EXEC REFUSED

There is no *REQUEST IS* line, but otherwise the message is the same as any labeled MOUNT message. You can decide to mount the volume, in which case you'll find it, mount it, and type CX MOUNTED @unitname↲. Or you can refuse the request with CX REFUSED. If you mount the tape, the user's I/O will occur; then EXEC will prompt for a ***UNIT DISMOUNT***. Remove the tape and type CX DISMOUNTED.

Users can specify only one volume in an implicit mount request. If more volumes are needed (on a write), EXEC will prompt you for them as if the user had typed a MOUNT/EXTEND command. Generally, if users or their programs may use implicit mounts, they should tell you beforehand so that you can have the pertinent volume(s) ready. Because they are restricted to one volume, implicit mounts are most useful for tape reads, where a user knows that the material he or she needs is in a single volume. Also, he or she must know the tape volume ID.

Specific Volume Requests

All material written by write operation (the DUMP or COPY command) makes up a single dump file (logical file). To read any part of this logical file, the system must read sequentially from the beginning — through multiple tape volumes if needed — until it arrives at the desired part.

There *is* a way to add a logical file to the end of a fileset, or to read from a logical file that starts on a tape other than the first tape in the dump fileset — without reading all preceding tape volumes. The LOAD and DUMP switch /SPECIFIC allows you to do this, by telling EXEC to start at a specific volume ID. For example, assume the commands

```
*) MOUNT/EXTEND/VOLID=VOL1/VOL=VOL2/VOL=VOL3 XTAPE Please ⤶
   (EXEC prompts for VOL1; tape is mounted and EXEC notified.)
*) DIR :⤶
*) DUMP/V :UDD:[!USERNAME]:XTAPE:USERS UDD:# ⤶
   (Dump proceeds to the middle of the second tape volume, VOL2)
*)
```

This DUMP command creates a logical file named USERS, containing all user directories and files. File USERS begins on VOL1 and ends on VOL2. At the end of the last volume, VOL2, the system rewinds the tape.

Let's say that the next step is to dump system (not user) material. You could use a new, different tape fileset. But (to use /SPECIFIC) let's assume you want to use the tape fileset that holds file USERS. Instead of having the tape dismounted, you'd type something like

```
*) DUMP/V/SPECIFIC @LMT:VOL2:SYSTEM #\UDD:# ⤶
   (EXEC prompts for VOL2; tape is mounted and EXEC notified)
```

The system spins tape VOL2 to the end of the preceding logical file (USERS) then starts dumping all non-UDD material to a logical file named SYSTEM. Let's say the dump proceeds to the third tape volume, VOL3. Then the system rewinds the tape.

At this point, you have two logical files, USERS and SYSTEM, on a three-volume fileset. If — later — you want to read a dumped disk file from file USERS, you'll need to start at the first volume. But if you want to read a dumped disk file from file SYSTEM, you can use the /SPECIFIC switch to avoid reading the first volume. For example, to retrieve file :UTIL:NEWERMES.CLI, you could type

```
*) DIR :⤶
*) LOAD/V/SPECIFIC @LMT:VOL2:SYSTEM UTIL:NEWERMES.CLI ⤶
   (EXEC prompts for VOL2 on the system console)
```

Then you mount and the system starts reading the specific volume VOL2. Doing it the other way would have required VOL1 to be mounted and read. The other way might look like this:

```
*) MOUNT/EXTEND/VOLID=VOL1/VOL=VOL2/VOL=VOL3 ZTAPE Buster ⤶
   (EXEC prompts for VOL1; tape is mounted and EXEC notified.)
*) DIR :⤶
*) LOAD/V [!USERNAME]:ZTAPE:SYSTEM UTIL:NEWERMES.CLI ⤶
```

Here, the system would need to read through VOL1 before reading VOL2, which is where logical file SYSTEM begins.

This technique, with the /SPECIFIC switch, is useful only if your site does multiple writes (like multiple dumps) to the same tape fileset.

You can discover the files on each volume by keeping a user log (CLI LOGFILE command at user console) during a dump; all the filenames dumped *and* CX MOUNTED commands will be recorded in this user log file. Then, you can print and delete the user log file. From the printout, you can tell which files are on each tape volume.

User Programs and Labeled Tapes

Users are not restricted to DUMP and LOAD with labeled tapes; user programs can also access labeled tapes. Such a program must open linkname:tape-file (if the user issued a MOUNT command before running the program). Or a program can open @LMT:volid:file, for an implicit mount. In either case, the program must open the tape file for either input or output, not both (this is true for all tape I/O). The program must write/read sequential fixed- or variable-length records; it cannot write/read data-sensitive or dynamic records.

If users want to create their own labels, they should use the standard formats described in the *AOS Programmer's Manual.*

Generally, the mount/dismount procedure will be smoother if users request explicit mounts via the MOUNT command before they run programs that need the volume(s). Implicit mounts can be tricky in such situations.

Using MOUNT and Label Tapes in Batch

Batch is ideal for large labeled tape dumps and loads. Such batch requests free the user console for other work and they can be queued for a time when workload is light. However, labeled tape I/O *does* require that an operator be on duty — which anyone can ensure by applying the /OPERATOR switch to the QBATCH or QSUBMIT command. Such jobs will not start if the operator is OFF duty (EXEC OPERATOR OFF command).

If you, as the system operator, want to DUMP or LOAD via labeled tapes, you should log onto a user console and do it in batch. This ensures that the job will run under EXEC, allowing for multiple-volume files and volume checking. (EXEC cannot run on the system console, so you cannot do effective labeled tape backup interactively from the system console.)

Figures 9-11 and 9-12, in the next chapter, show two macros that you can use to batch a multifileset, multivolume dump.

Mount/Dismount Summary and Pointers

- Whenever you, the operator, see a ***UNIT MOUNT*** message, you can get the (specified) tape, mount it on an idle tape unit, and type CX MOUNTED @unitname ↓. Or you can refuse the request via CX REFUSED ↓.
- If EXEC's MOUNT message includes a volid, the request is for a labeled tape. You must find a tape with the specified volid, and mount it; or refuse the request. *Make sure that that volid is correct*; if you are uncertain, use TYPE @unitname ↓ to verify the volid. If the request is for multiple volumes, you can eliminate future UNIT MOUNTs by mounting all volumes (or as many as possible), then typing MOUNT for the first volume and PREMOUNT for the others. Then EXEC will manage the tape volumes without operator intervention.
- You can use the LABEL utility to create tape labels, at any time — even when EXEC is prompting you with a ***UNIT MOUNT*** request.
- You can use EXEC's MOUNTSTATUS command to keep track of mount requests. And you can use the UNITSTATUS command to see which tape units are mounted and premounted.
- If you get confused, or if EXEC seems unwilling to let go of a tape unit, you can always solve the problem by having the user who issued the request log off. This prompts EXEC to sever all connections with the unit and all you need type is CX DISMOUNTED↓.
- Within any labeled tape fileset, writes must occur sequentially, from one volume to the next. If you set up multiple filesets with different linknames (e.g., TAPE1, TAPE2, TAPE3), then you can dump to each fileset simultaneously — saving time if you have multiple tape units. If you have only one fileset, you can dump to only one volume at a time.
- Any user (including yourself) can specify *more* volumes than needed for a tape write; e.g., a dump. When all specified material has been written to tape, the user DISMOUNT command will write an EOF trailer on the current tape volume. The fileset write will be complete and you can ignore "extra" volids specified.

  Or, any user can apply the /EXTEND switch to the MOUNT command. If so, if the user's tape I/O needs more tape volumes than the user specified, EXEC will prompt the operator to label another tape and mount it.

But if a user specifies *too few* volumes, and omits /EXTEND, tape file space will be exhausted. No EOF trailer can be written; EXEC will display a *NO MORE VOLIDS*... error message on the user's console; and the fileset will be incomplete. And — generally — the write must be redone from the beginning. This can be a real bummer at the end of a 10-volume dump. So, generally, users should be generous in the volid lists, or they should use MOUNT/EXTEND, or both.

- A user can start a labeled tape read or write on a specific volume with the LOAD or DUMP command and /SPECIFIC switch. The mount can be explicit or implicit. If explicit (MOUNT/VOLID=...), the LOAD command must use the form @LMT:first-volid:fileset-name (not tape linkname) to access the tape.
- A 2400-foot tape, at high density (1600 bpi), dumped with a buffer size of 8192 bytes, can hold about 39 megabytes of disk-based information. A 1000-foot tape can hold about 16 megabytes. Either tape can hold slightly more with a bigger buffer size, if you want to generate a system with a maximum tape buffer size larger than 8K. On an MTB tape unit, a 2400-foot tape takes about 12 minutes to fill (DUMP_II program) or 20 minutes to fill (DUMP command). On an MTC tape unit, a 1000-foot tape may take about the same amount of time (12 or 20 minutes).
- Batch processing is ideal for multivolume labeled tape I/O.
- An essential part of the MOUNT/DISMOUNT/labeled tape business is planning and implementing the labels and organizing the tape library.

Mount Commands to EXEC

The EXEC mount commands are DISMOUNTED, MOUNTED, MOUNTSTATUS, PREMOUNT, REFUSED, and UNITSTATUS.

These are summarized in Table 8-3 (earlier) and detailed in the alphabetical section, next.

EXEC Commands — Listed Alphabetically

This section describes all EXEC commands, in alphabetical order. For an EXEC command summary, see Appendix C.

ALIGN

Halts or continues the XLPT process so you can align paper.

CX ALIGN *[/CONTINUE]* @devicename *[[-]n]*

where

/CONTINUE	continues the device.
devicename	is the name of a spooled device; it begins with @; e.g., @LPB.
n	is the page number where you want XLPT to restart processing. A positive *n* indicates an absolute page number in the file. A negative *n*, like -1, indicates a page *before* the last page printed.

The ALIGN command tells XLPT to stop printing. If even pagination is enabled (default), printing stops at the first line of the next even-numbered page. To check the page XLPT was printing when the printer stopped, use the EXEC command STATUS @devicename.

When you reissue ALIGN, you can specify where XLPT should restart the job. To restart, use the command

CX ALIGN/CONTINUE @devicename *[-] [n]*

XLPT keeps track of the previous 32 pages. If you request XLPT to continue printing at -33 pages before the one it was printing when the job stopped, XLPT won't know where the page is in the file. In this case, XLPT must go to the beginning of the file and calculate which page you are requesting. This command may take more time to process.

Why Use It?

Printer paper may be out of alignment, or the paper may jam. ALIGN stops the printer so you can align paper or do other things with the printer; then resume printing.

Example

If you are printing a large file on a printer named @LPB, you'd type

) CX ALIGN @LPB ↵

When you have corrected the jam, continue printing by typing

) CX ALIGN/CONTINUE @LPB ↵

This restarts the active job at the page where it stopped. If printing stopped on page 10, however, you might want to restart printing on page 9, instead of on page 11. In that case, you'd type

) CX ALIGN/CONTINUE -1 ↵

XLPT would then restart printing at the page where it stopped, minus one. This would be 10-1, or page 9.

BINARY

Tells the XLPT process to enable or disable binary mode.

```
CX  BINARY  @devicename  { filename }
                         { OFF      }
```

where

devicename	is the name of a spooled device; it must begin with @; e.g., @CON27.
filename	is the cleanup filename. EXEC expects to find this file in directory :UTIL:FORMS.
OFF	disables binary mode.

BINARY directs the XLPT co-operative process to enable or disable binary mode on a device. Binary mode is useful when users want to print on a device that interprets characters its own way; for example, a graphics printer or plotter. For such printers, you don't want XLPT to edit special characters that have meaning to the device; you want it to pass each character along as is. This is what binary mode does. Binary mode is not useful for a printer that supports only a Vertical Forms Unit (VFU); for example, LPB-type printers.

In binary mode, the XLPT process does not interpret characters with values 0 through 376 octal; it passes them directly from the file to the device. Since it doesn't interpret these characters, XLPT cannot keep track of page numbers, line numbers, or any other data that requires character interpretation. So, after a file has been printed in binary mode, the printer is left in an unknown state — perhaps with paper positioned in the middle of a page.

Thus, you must supply the name of a cleanup file (filename) when you enable binary mode. The cleanup file can position paper at the physical top of page, restore margins and tabs, and put the printer in its prebinary state. A cleanup file *can* include only a form feed (ASCII 14, CTRL-L) to position the paper at physical line 1 for the next user. The XLPT process sends the cleanup file to the device when you enable binary mode and after the device prints a file in binary. The cleanup file must be in directory :UTIL:FORMS. For printers under the CEO system, you create cleanup files when you define printers.

To have a file printed in binary mode, a CLI user must append the /BINARY switch to the QPRINT command; for example,

```
) QPRINT/QUEUE=LQP/BINARY  MYFILE ↲
```

If binary mode is not enabled for the device associated with the specified queue, the printed output will report a *BINARY MODE NOT ENABLED* error.

The printer that will process files in binary mode must be paused and idle before you can enable binary mode. After enabling binary mode, you must continue the printer.

There is one character that XLPT does interpret in binary mode: a character with a value of 377 octal. XLPT ignores 377 octal and any character (except 377 octal) that follows the first 377 octal. So, if someone wants to pass a 377 octal to the printer in binary mode, the file must contain two sequential octal 377s.

To see whether binary mode is enabled or disabled, use EXEC's SPOOLSTATUS command.

Why Use It?

Binary mode is required for letter-quality printers used by CEO users. It is also required when you need to have all characters printed precisely as they are in the file, without intepretation by the XLPT process.

If a device will be used exclusively in BINARY mode, you may want to put the BINARY command in the UP.CLI macro.

Example

```
) CX   PAUSE   @CON26 ↵
```

FROM PID 3 : (EXEC) @CON26 PAUSED

```
) CX   BINARY   @CON26   CLEANUP_26 ↵
```

FROM PID 4 : (EXEC) @CON26 BINARY MODE ENABLED

```
) CX   CONT   @CON26 ↵
.
.
.
) CX   PAUSE   @CON26 ↵
) CX   BINARY   @CON26   OFF ↵
) CX   CONT   @CON26 ↵
```

BRIEF

Tells EXEC to make its batch or spool messages brief.

CX BRIEF [*stream* | *@devicename*]

where

stream is a batch stream number; e.g., 1, 2, 3, or 4. If you omit a stream number, the command affects all batch streams.

devicename is the name of a spooled device; it must begin with @; e.g., @LPB.

When a batch stream or spooled device accepts or processes a request, EXEC sends a message to the system console. This message may be either "brief" or "verbose." The EXEC BRIEF and VERBOSE commands determine the type of EXEC messages sent. Each time you issue a BRIEF or VERBOSE command, it overrides the current message setting. BRIEF is the default mode.

BRIEF messages include

the batch stream number or devicename;

the job sequence number; and

the user's username.

Why Use It?

You may find EXEC messages easier to read if they contain less information. (You can also suppress EXEC's time of day prompt with the command CX PROMPTS OFF⤶. And you can suppress all batch/spool messages with the CX SILENCE command.)

Example

```
) CX  BRIEF  2 ⤶
.
.
.
FROM PID 3 : (EXEC) STREAM_2 SEQ=446, USR=sackville
) CX  BRIEF ⤶
.
.
.
FROM PID 3 : (EXEC) STREAM_1 SEQ=447, USR=F77
) CX  BRIEF  @LPB ⤶
FROM PID 3 : (EXEC) @LPB SEQ=448,USR=SALLY
```

CANCEL

Cancels all waiting queue entries with the given sequence number.

CX CANCEL sequence-number

where

sequence-number — is the sequence number of the queue entry to be cancelled.

The CANCEL command cancels the specified queue entry immediately. It doesn't work for an active entry; use the FLUSH command for this.

The batch input, output, and list for each batch job have the same sequence number. One sequence number removes the output and list files; the input file remains in the user's directory.

The CLI command QDISPLAY lists queue entries and their queue sequence numbers. Active entries are marked with an asterisk (*) in the display. Entries that the operator has cancelled appear with an F flag in the display.

After you cancel an entry, the batch output file, or the printed output file, will show the message CANCELLED BY OPERATOR. For batch requests, the temporary batch input file remains in the user's directory.

Instead of cancelling entries yourself, you might send a message to the user, asking him or her to cancel it with the CLI command QCANCEL. QCANCEL (unlike EXEC's CANCEL) works with active requests.

Why Use It?

There may be times when you don't want requests to remain enqueued. For example, there may be many requests or large requests waiting when you want to shut down a stream, device, or EXEC. CANCEL allows you to empty a stream or spool queue quickly.

Example

```
) QD ↵
BATCH_INPUT BATCH OPEN
535 DA JACK :UDD:JACK:F77:?040.CLI.004.JOB
.
.
.
) CX  CAN  535 ↵
FROM PID 3 : (EXEC) @LPB USR=JACK
) QD ↵
BATCH_INPUT BATCH OPEN
.
.
.
```

Here, the CANCEL command cancelled the request with sequence number 535. The printer became active because it was printing the batch output file with the message CANCELLED BY OPERATOR.

CLOSE

Closes the specified queue to user requests.

CX CLOSE queuename

where

queuename is the queue you want to close.

The CLOSE command closes the specified queue and prevents users from submitting more requests to the queue. Once you close the queue, users will get *CLOSED* error messages when they submit requests to the queue. The queue will continue to process requests within it, but no new requests can be added to the queue.

If you try to close a queue that is not open, EXEC gives an error message.

To reopen a queue, you must use the EXEC OPEN command.

The CLI command QDISPLAY will tell you which queues are open and which are closed.

Why Use It?

Occasionally, you may want a queue to stop accepting requests; perhaps if you want to reorganize it or change it, or if the device involved needs servicing. CLOSE prevents the queue from accepting any more requests. If you wish, you can then use CX CANCEL commands to clean it out.

Example

) CX CLOSE LPT ⮠

) QPRINT MYFILE ⮠

WARNING: QUEUE IS NOT OPEN

)

CONSOLESTATUS

Displays status of each console enabled by EXEC.

CX CONSOLESTATUS *[@consolename]*

where

@consolename is the device name for the console; it must begin with @ to specify the peripherals directory; e.g., @CON44.

If you omit an argument, the CONSOLESTATUS command displays the status of all enabled consoles. To check a specific console, give its name as an argument.

For each active console, EXEC displays the username, console name, and process ID; it also tells you whether the console will be disabled when the current user logs off, and if a user is in the process of logging on or off.

Why Use It?

CONSOLESTATUS is EXEC's version of the ?.CLI macro. But since it shows consolenames first, it's more useful than ?.CLI if you are more interested in console numbers than in the PIDs or users associated with them.

Also, since it identifies consoles on which someone is logging on or off, it can help you identify noisy console lines.

Example

) CX CONSOLES ↵

...(status information on all enabled consoles)...

) CX CONSOLES @CON46 ↵

FROM PID 3 : (EXEC) @CON46 PID=50 USR=jack

) CX CONSOLES @CON4(6 7 8) ↵

FROM PID 3 : (EXEC) @CON46 NOT LOGGED ON
FROM PID 3 : (EXEC) @CON47 LOG ON/LOG OFF IN PROGRESS
FROM PID 3 : (EXEC) @CON48 CONSOLE UNKNOWN TO EXEC
)

These messages indicate an enabled console in use, not in use, in logon/logoff transition, and a console not enabled by EXEC. The parentheses enclosing 6 7 8 in the last command are a CLI feature to help you write compact command lines.

CONTINUE

Directs one or more batch streams or devices to continue processing (opposite of PAUSE).

CX CONTINUE [*stream* | *@devicename*]

where

stream — is a batch stream number; e.g., 1, 2, 3, or 4. If you omit a stream number, the command affects all batch streams.

devicename — is the name of a spooled device; it must begin with @; e.g., @LPB.

The CONTINUE command continues a paused batch stream or device.

Continuing one or all batch streams has EXEC continue processing BATCH_INPUT jobs queued to the stream(s). Continuing a device name has EXEC continue processing requests from that device's spool queue.

If the command works, EXEC responds

FROM PID n : (EXEC) stream-or-device CONTINUING

If the command fails, EXEC gives an error message.

Issuing the CONTINUE command to an active device or stream has no effect.

Why Use It?

CONTINUE is needed to restore normal processing whenever you have paused a stream or device. It is also needed after you have started one or more queues on a device. The macro UP.CLI issues several CONTINUE commands as it brings up the multiuser environment.

Example

) CX PAUSE 3 ↵

FROM PID 3 : (EXEC) STREAM_3 PAUSED

) CX QPRIORITY 3 127 255 ↵

) CX CONT 3 ↵

FROM PID 3 : (EXEC) STREAM_3 CONTINUING

CPL

Changes the number of characters per line for a device.

CX CPL @devicename number

where

devicename is the name of a spooled device; it must begin with @; e.g., @LPB.

number is the number of characters per line. It must be an integer between 16 and 255.

The CPL command sets a new number of characters per line (CPL) for this device. It overrides any previous AOSGEN, CHARACTERISTICS/CPL, or EXEC CPL settings. The new CPL remains until you change it with EXEC's CPL or bring down EXEC's XLPT cooperative process.

To change a CPL setting:

- Type CX PAUSE @devicename ↵
- When EXEC tells you that the device is paused, type the desired CX CPL command.
- Type CX CONTINUE @devicename ↵

CPL does nothing on a device if an EXEC DEFAULTFORMS or FORMS command is in effect on the device.

Why Use It?

The default number of characters per line is 80.

On any printer, if a line to be printed exceeds the max CPL characters, it will be truncated to CPL on output unless the user applied the /FOLDLONGLINES switch to the QPRINT command. Having lines truncated is undesirable, and having them folded is often messy. To avoid both, you will generally want the longest possible line to print as in the file — and most printing paper has space for at least 85 characters per line (80 column) or 136 characters per line (132 column). To have more than 80 characters printed, you must use the CPL command.

After deciding on a good CPL for each printer, you may want to specify it in the UP macro, before continuing the printer.

Example

) CX PAUSE @LPB1 ↵

FROM PID 3 : (EXEC) @LPB1 PAUSED [IDLE]

) CX CPL @LPB1 136 ↵
) CX CONT @LPB1 ↵

To put the CPL command in the UP macro, you'd simply insert the line

CX CPL @LPB1 136

before the EXEC CONTINUE command for @LPB1.

CREATE

Creates a spool queue.

```
                ( PRINT  )
                ( PLOT   )
CX   CREATE     { HAMLET }
                ( FTA    )
                ( SNA    )
```

where

queuename is the name of the new spool queue.

The CREATE command creates a spool queue for a line printer or other hardcopy printing device (PRINT), plotter device (PLOT), HAMLET process (HAMLET), XODIAC FTA process (FTA), or SNA_RJE process (SNA). The spool queuename can contain any legal filename characters. But if the queue is to be accessed by a CLI Q-series command without a /QUEUE= switch, the queuenames must be

Queue Type	Queue Name	Accessed by CLI command
PRINT	LPT	QPRINT
PLOT	PLT	QPLOT
HAMLET	HAMQ	QSUBMIT
FTA	FTQ	QFTA
SNA	SNA	QSNA

To access a queue with any other name, users must append the /QUEUE= switch. For example, for a queue named LPT1:

) QPRINT/QUEUE=LPT1 MYFILE ↵

Users can submit jobs to a spool queue in three ways. They can

- specify the spool queue pathname as a listing file with the /L= switch; e.g., WRITE/L=@LPT1 Hello ↵
- specify the spool queue pathname as an output destination file in a CLI command; e.g., COPY @LPT1 MYFILE ↵
- explicitly type a CLI QPRINT, QPLOT, QPUNCH, QFTA, or QSUBMIT (for HASP II/HAMLET) command; e.g., QPRINT MYFILE ↵

One CREATE command is needed for each queue in your system. Then, for the new queue, the OPEN, START, and CONTINUE commands are needed to make the queue usable. Users cannot submit jobs to the queue until it is open, and jobs won't be processed until the queue has been started on a device and the device has been continued.

CREATE and OPEN are generally issued only once, to set up the queue in :QUEUE and :PER. Thereafter, START and CONTINUE commands make the queue available. START and CONTINUE are often in the UP.CLI macro.

Chapter 5 of this book had the reader create and open the needed queues, then edit START and CONTINUE commands into the UP.CLI macro.

Why Use It?

EXEC, as shipped, contains no spool queue names. Someone must create at least one type PRINT queuename (default LPT) before users can submit batch or printing requests. Other queues may be desirable. The CREATE command allows you to create queues.

Example

The following commands create, open, and start the default print queue named LPT for the printer LPB:

```
CX  CREATE  PRINT  LPT
CX  OPEN  LPT
CX  START  LPT  @LPB
CX  CONTINUE  @LPB
```

The following example shows a create, open, and start for a print queue named PRINTER:

```
) QPRINT/QUEUE=PRINTER  MYFILE ⤶
```

WARNING: QUEUE DOES NOT EXIST: PRINTER

```
) CX  CREATE  PRINT  PRINTER ⤶

) CX  OPEN  PRINTER ⤶

) CX  START  PRINTER  @LPB ⤶

) CX  CONTINUE  @LPB ⤶

) QPRINT/QUEUE=PRINTER  MYFILE ⤶
```

QUEUED, SEQ=566

)

The following command lines create, open, and start a HAMLET (HASP II) queue:

```
) CX  CREATE  HAMLET  HAMQ ⤶

) CX  OPEN  HAMQ ⤶

) CX  START  HAMQ  @SLNO ⤶
```

DEFAULTFORMS

Sets the characteristics of the default form for a device.

CX DEFAULTFORMS devicename *[form-name]*

where

devicename	is the name of a spooled device; it must begin with @; e.g., @LPB1.
form-name	is the name of a file containing formatting commands for printed output. EXEC expects to find this file in directory :UTIL:FORMS.

The DEFAULTFORMS command sets the default form for a device. All files printed on this device will be printed per this form unless a user asks for a special form with the /FORMS= switch.

The new form specifications must be placed in a form file in :UTIL:FORMS. You, or a user, can put printing directives in the form file with the FCU utility, as sketched earlier in the chapter.

If you omit a form-name argument, EXEC uses the following default characteristics for printing files.

- the current setting for lines per page (LPP command); default is 66;
- the current setting for characters per line (CPL command); default is 80;
- the top of form is line 1 if LPP is less than 7; otherwise, the top of form is line 4;
- the bottom of form is the number of lines per page (LPP).

If you use DEFAULTFORMS to set lines per page or characters per line, you cannot use EXEC's LPP or CPL command to change these while the new DEFAULTFORM is in effect. Instead, you must reissue DEFAULTFORMS without a form-name argument; this restores the standard form, allowing LPP or CPL commands to work.

EXEC's SPOOLSTATUS command will tell you if nonstandard DEFAULTFORMS values are in effect.

As with any device parameter change, the device must be paused before EXEC will accept the command.

Why Use It?

For specific nonstandard printing jobs, you'd use EXEC's FORMS command. But if most or all of a printer's work will be on nonstandard forms, you might want to make the nonstandard form the DEFAULTFORM for this printer.

Example

```
) CX   PAUSE   @LPB1 ↵
```

FROM PID 3 : (EXEC) @LPB1 PAUSED [IDLE]

```
) CX   DEFAULTFORMS   @LPB1   STANDARD_LPB1 ↵
) CX   CONT   @LPB1 ↵
```

This sequence sets the standard form for @LPB1 to the specifications in file :UTIL:FORMS:STANDARD_LPB1 (built with the FCU program). Later, to restore the standard form to @LPB1, you'd type

```
) CX   PAUSE   @LPB1 ↵
```

FROM PID 3 : (EXEC) @LPB1 PAUSED [IDLE]

```
) CX   DEFAULT   @LPB1 ↵
) CX   CONT   @LPB1 ↵
```

DELETE

Deletes a spool queue (opposite of CREATE).

```
CX  DELETE  queuename
```

where

queuename is the name of the spool queue you want to delete.

The DELETE command deletes the specified queue. The queue must be closed, stopped, and empty before EXEC will accept this command. Use the CX CLOSE, STOP, and PURGE commands to do this. You cannot delete the permanent queues, BATCH_INPUT, BATCH_OUTPUT, or BATCH_LIST.

Why Use It?

DELETE can be handy if you have created a temporary queue and want to get rid of it. Generally, you would not want to delete often-used, general-purpose queues like LPT.

Example

The following commands close a print queue, dissociate the queue from whatever devices are processing it, purge the queue of all its entries, and, finally, delete the queue entirely.

```
) CX  CLOSE  PRINTER ↵
) CX  STOP  PRINTER ↵
) CX  PURGE  PRINTER ↵
) CX  DELETE  PRINTER ↵
```

DISABLE

Removes EXEC log on capability from a console.

CX DISABLE @consolename

where

@consolename is the device name for the console; it must begin with @; e.g., @CON40.

The DISABLE command removes the EXEC logon capability provided by ENABLE. It prevents users from logging on at the specified console. If a user is logged on to the console, EXEC will not log him or her off; instead, it will wait until the user has logged off before implementing the DISABLE; and it will display a *WILL BE DISABLED* message on the system console.

You can undo a DISABLE command to an active console by issuing an ENABLE command to that console before the user logs off.

After EXEC disables a console, it displays the message

sysid Console DISABLED FOR LOGGING ON

on that console.

Why Use It?

Often, you will want to shut the system down, or release a console from EXEC so that another program (like DG/SNA or DATAPREP) can use it. In either case, you don't want people to log on via EXEC. Use DISABLE to handle either situation.

Example

) CX DISABLE @CON2 ↵

) CX DISA @CON4(7 8 9) ↵

FROM PID 3 : (EXEC) CONSOLE WILL BE DISABLED
FROM PID 3 : DISA @CON47
FROM PID 3 : (EXEC) CONSOLE UNKNOWN TO EXEC
FROM PID 3 : DISA @CON48
FROM PID 3 : (EXEC) FILE DOES NOT EXIST
FROM PID 3 : DISA @CON49

) CX ENABLE @CON47 ↵

FROM PID 3 : (EXEC) DISABLE CANCELLED, CONSOLE ENABLED
FROM PID 3 : ENABLE @CON47
)

These commands attempt to disable consoles CON2, CON47, CON48, and CON49, then undo the disable on CON47. EXEC returned no message from the first command, meaning that @CON2 was disabled. The other messages mean that CON47 was in use, that CON48 wasn't enabled, and that CON49 wasn't specified to AOSGEN.

DISABLE (continued)

For convenience, you might want to write a CLI macro that disables all consoles. It should use the CLI PAUSE command to PAUSE for about a second or so per console. It might look something like

```
CONTROL   @EXEC   DISABLE   @CON(2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18)
PAUSE  16
CONTROL   @EXEC   DISABLE   @CON(19,20,21,22,23,24,25,26)
PAUSE  8
.
.
.
etc.
```

Then, when you want to disable all consoles, you'd simply type the macro name.

DISMOUNTED

Tells EXEC that you have physically dismounted the tape(s) associated with a mount request.

CX DISMOUNTED [*mid* | @*devicename*]

where

mid — is the mount identifier: an integer shown by EXEC for each MOUNT request and displayed by EXEC's MOUNTSTATUS command.

devicename — is a magnetic tape unit; it must start with @; e.g., @MTB1 or @MTC0.

The DISMOUNTED command tells EXEC that you have physically removed the tape from the unit. EXEC deletes the user's linkname to the tape unit, and restores the unit's ACL to the setting it had before the mount.

If you omit arguments, the command applies to all tape volumes in the current dismount request. If you give a mount identifier (mid), this tells EXEC that *all* tapes associated with the *specified* request have been dismounted; it is handy when you want to specify a request other than the current request.

If you give a device name, this tells EXEC that the tape on a unit has been dismounted. This form is meant for situations where you PREMOUNT a volume, then change your mind and want to dismount it. It's also useful when you have mounted two units for a user, and want to dismount one of them for another user.

If SYSLOG logging is on (Chapter 9), the log file will record the elapsed time that the user had the tape mounted.

EXEC will prompt you to type the DISMOUNTED command

- when a user who has a tape mounted types the CLI command DISMOUNT↵;
- when a user who has a tape mounted logs off without typing the CLI command DISMOUNT↵; or
- when a user's I/O is complete after you've followed his or her implicit mount request (e.g., LOAD/V @LMT:GP0076:MY_FILESET PROG+ ↵).

When EXEC prompts you to type DISMOUNTED, it may also display a REQUEST IS line with additional information from the user. The REQUEST line will not appear in an implicit mount request. It will also not appear if the user program terminated before the tape was dismounted or if the user omitted a comment when he or she typed the CLI DISMOUNT command.

DISMOUNTED (continued)

(EXEC does not check to see that you have actually taken the tape off line and unloaded it — so you can type DISMOUNTED before you physically dismount it. But if you do this, the unit ACL will revert before you have actually removed the tape, possibly allowing someone to write to it.)

Why Use It?

You must issue DISMOUNTED when EXEC prompts for it to release the tape unit(s) from EXEC's mount mechanism.

Type CX DISMOUNTED @devicename when you have premounted a volume, but have dismounted it before it was been used.

Example

An unlabeled tape user asks for DISMOUNT:

*FROM PID 4: (EXEC) ** UNIT DISMOUNT ***
FROM PID 4: UNIT(S) ARE: @MTB3
FROM PID 4: REQUEST IS 'Thanks'
FROM PID 4: RESPOND: CONTROL @EXEC DISMOUNTED

Remove the tape from the unit and type

) CX DISMOUNTED ↓

A labeled tape user asks for a dismount

*FROM PID 3: (EXEC) ** UNIT DISMOUNT ***
FROM PID 3: MID=55,USER=F77,PID=33
FROM PID 3: UNIT(S) ARE: @MTB2, @MTB3
FROM PID 3: REQUEST IS 'Done. Please file. Thanks.'
FROM PID 3: RESPOND: CONTROL @EXEC DISMOUNTED

Type

) CX DISMOUNTED ↓

and remove the user's tapes from MTB2 and MTB3. If you premount a tape:

) CX PREMOUNT @MTB2 DB0067 DATABASE ↓

then change your mind about the premount, remove the tape from the unit and type

) CX DISMOUNTED @MTB2 ↓

ELONGATE

Turns elongated printing on or off on an LP2 or TP2 printer.

CX ELONGATE @devicename { ON | OFF }

where

devicename — is the name of a spooled device; it must begin with @ and be the name of a DASHER LP2 line printer or DASHER TP2 console printer.

DASHER LP2 and TP2 printers feature elongated printing that allows you to vary character width. The ELONGATE command turns elongated printing for either of these on or off.

The device must be paused. After you change the setting, type the CX CONTINUE command.

Why Use It?

If one of these DASHER devices is under control of EXEC's XLPT process, you must use ELONGATE to get elongated printing on it.

Example

) CX PAUSE @LPC ↵

FROM PID 4 : (EXEC) @LPC PAUSED [IDLE]

) CX ELONGATE @LPC ON ↵

) CX CONTINUE @LPC ↵

FROM PID 4 : (EXEC) @LPC CONTINUING

ENABLE

Provides EXEC log on capability on a user console.

```
CX  { ENABLE   @consolename }
    { ENABLE/ALL            }
```

where

@consolename is the device name for the console; it must begin with @; e.g., @CON24.

/ALL tells EXEC to enable all consoles that were identified to AOSGEN, and virtual consoles. EXEC will report an error, but continue enabling, if a console is already enabled or is "owned" by another console-managing program.

The ENABLE command gives the specified console — or all consoles — log-on capability. A user may then log on to each enabled console via EXEC and have access to the AOS system.

If an ENABLE @consolename succeeds, EXEC displays the following message on the system console

FROM PID n : (EXEC) ENABLED CONSOLE, @CONn

When it starts executing an ENABLE/ALL command, EXEC displays

FROM PID n : (EXEC) ENABLING ALL CONSOLES

.

.

.

FROM PID n : (EXEC) ALL CONSOLES ENABLED

In every case, if a console has already been enabled, or is in use by another console-managing program, EXEC will display an error message.

After being enabled, each console (if ON LINE) shows the message

SYSID TYPE NEW-LINE TO BEGIN LOGGING ON

You cannot enable the system console, CON0.

Why Use It?

ENABLE is needed to give most users access to the system. Usually, the UP.CLI macro enables the desired consoles via ENABLE/ALL or multiple ENABLE commands.

Example

) CX ENABLE @CON3 ↓

FROM PID 3 : (EXEC) ENABLED CONSOLE @CON3

) CX ENABLE @CON(4,5,6) ↓

FROM PID 3 : (EXEC) CONSOLE ALREADY ENABLED @CON4
FROM PID 3 : (EXEC) COULD NOT ENABLE @CON5
FROM PID 3 : (EXEC) ENABLED CONSOLE @CON6

) CX ENABLE/ALL ↓
FROM PID 3 : (EXEC) ENABLING ALL CONSOLES

(pause)

FROM PID 3 : (EXEC) ALL CONSOLES ENABLED

EVEN

Turns even pagination on or off for a device.

CX EVEN @devicename { ON | OFF }

where

devicename is the name of a spooled device; it must begin with @; e.g., @LPB.

The EVEN command turns even pagination on or off for the specified device. The default setting is ON. You can check the current setting with EXEC's SPOOLSTATUS command. The device must be paused before EXEC will accept the command.

When EVEN is on, XLPT prints all files as though they contain an even number of pages. If a file has an odd number of pages, XLPT puts out an extra, blank sheet to make the file appear even numbered. This puts all header pages on the same fold of the paper.

When EVEN is off, XLPT will print queued files just as they are, one after the other, with no intervening blank pages.

Why Use It?

Under some circumstances — perhaps to conserve printer paper — you might want files printed just as they are, with an even or odd number of pages. In most cases, you'll want to stick with the default, ON.

Example

) CX PAUSE @LPB1 ⤶

FROM PID 3 :(EXEC) @LPB1 PAUSED [IDLE]

) CX EVEN @LPB OFF ⤶

) CX CONT @LPB1 ⤶

FLUSH

Flushes (kills) the active request in a stream or queue.

```
CX  FLUSH  { stream      }
           { @devicename }
```

where

stream — is a batch stream number; e.g, 1, 2, 3, or 4.

devicename — is the name of a spooled device; it must begin with @; e.g., @LPB.

The FLUSH command tells EXEC to stop processing the active request in a stream or queue; EXEC then starts the next request (if any).

For each flushed job, the batch output or printed file on the printer will show a *TERMINATED BY OPERATOR* message.

Why Use It?

Sometimes you will want to stop an active batch or spool request. For example, to

- terminate a job that is looping,
- stop printing a file that you don't want printed; or
- empty queues so you can shut down EXEC.

FLUSH is the easiest way to kill an active request.

Example

FROM PID 3 : (EXEC) STREAM_1 SEQ=582 USR=sackville

) CX FLUSH 1 ↵

FROM PID 3 : (EXEC) STREAM_1 FLUSHING CURRENT JOB

FROM PID 3 : (EXEC) STREAM_1 SEQ=583 USR=mammon

) CX FLUSH (2,3) ↵

FROM PID 3 : (EXEC) STREAM_2 FLUSHING CURRENT JOB
FROM PID 3 : (EXEC) STREAM_3 HAS NO CURRENT JOB

) CX FLUSH @LPB ↵

FROM PID 3 : (EXEC) @LPB FLUSHING CURRENT JOB

This command sequence flushes stream 1 — which then accepts the next request. Then, in one FLUSH command, it attempts to flush streams 2 and 3. Finally it flushes the current @LPB request.

FORMS

Tells a printer to use special forms for printing.

CX FORMS @devicename *[form-name]*

where

devicename is the name of a spooled device; it must begin with @; e.g., @LPB1.

form-name is the name of a file containing formatting commands for printed output. EXEC expects to find this file in directory :UTIL:FORMS.

The FORMS command directs a printing device to process all requests that specify the special form called form-name. Form-name is the name of a special form, like a paycheck form for a line printer. Someone must have created the form file in the :UTIL:FORMS directory and tailored it with the FCU program (sketched earlier in this chapter).

A user can request any special form by appending the /FORMS=form-name switch to a CLI QPRINT or QPLOT command. After a user does this, EXEC retains the request in the output queue until you type a FORMS command that specifies the form-name. To see which forms await special forms handling, type

) QDISPLAY/V ⤶

When you decide to process output requests that need a special form on any device, follow these steps:

1. Pause the printer with CX PAUSE @devicename ⤶. Wait until EXEC tells you that the printer is paused.
2. Insert the desired forms in the printer.
3. Tell the printer XLPT process to use the new form specification by typing

   CX FORMS @devicename form-name ⤶
4. Continue the printer (CX CONTINUE @devicename ⤶).

   The printer will then start printing the forms. It will print *only* those files submitted with QPRINT/FORMS=form-name; other print requests to that printer will wait in the queue.

   When the printer finishes the special form requests, it will become idle.
5. Now, type another CX PAUSE devicename ⤶.
6. Reinsert the original forms and type

) CX FORMS @devicename ⤶
7. Continue the printer with CX CONTINUE @devicename ⤶.

The printer will now process output requests submitted without a /FORMS= switch.

FORMS (continued)

Why Use It?

Many installations use special printing forms; e.g., for paychecks or invoices. The FORMS command allows you to dedicate a printer to printing all user requests that specify a special form. (You could use DEFAULTFORMS instead if you wanted *all* requests to the printer printed the same way.)

Example

FROM PID 15 : (PAYROLL) Ready to print paychecks.

```
) SEND 15 Ok, type QPRINT/FORMS=PAYCHECK_FORM CHECKS ↵
```

FROM PID 15 : (PAYROLL) Ok, typed QPRINT command.

```
) CX PAUSE @LPB1 ↵
```

FROM PID 3 : (EXEC) @LPB1 PAUSED [IDLE]

... someone puts paycheck forms in printer ...

```
) CX FORMS @LPB1 PAYCHECKS ↵
) CX CONTINUE @LPB1 ↵
```

... paycheck printing occurs ...

FROM PID 3 : (EXEC) @LPB1 [IDLE]

```
) CX PAUSE @LPB1 ↵
```

FROM PID 3 : (EXEC) @LPB1 PAUSED [IDLE]

... someone puts standard paper in printer ...

```
) CX FORMS @LPB1 ↵
) CX CONTINUE @LPB1 ↵
```

FROM PID 3 (EXEC): @LPB1 CONTINUING

Here, the operator receives a message that Payroll is ready to print paychecks. The operator sends Payroll the QPRINT command syntax; Payroll types and verifies the QPRINT command; and the operator pauses the printer. Then, someone puts the blank paycheck forms in the printer and continues the printer. The operator sets the PAYCHECK printing specification with EXEC's FORMS command; and the checks are printed.

After printing, the operator receives the *[IDLE]* message and pauses the printer again. Someone puts standard paper in the printer; the operator sets standard @LPB form specification; and the printer resumes processing standard printing requests.

HEADERS

Changes the number of header sheets before each printed job.

```
                         { 0 }
CX  HEADERS  @devicename { 1 }
                         { 2 }
```

where

devicename — is the name of a spooled device; it must begin with @; e.g., @LPB1.

0, 1, or 2 — is the number of header sheets you want.

The HEADERS command sets the the number of header sheets to be printed before the body of each printing request. By default, EXEC provides one header sheet per request.

The header sheet gives a lot of information, including

- Username;
- Queuename;
- Device name;
- Sequence number;
- Qpriority;
- Lines per page;
- Characters per line;
- Number of copies requested;
- Page limit;
- Switches applied by user;
- Date(s) the file was created, queued, and processed;
- Pathname of file;
- System identifier (SYSID);
- Revision of AOS and XLPT program.

EXEC's SPOOLSTATUS command describes the number of headers set for a device.

As with CPL, etc., the device must be paused before EXEC will accept the command; and you must continue the device afterward.

Why Use It?

On a slow printing device (like a letter-quality printer), omitting a header page can speed up processing.

For standard printers: some installations like 2 header pages to provide more separation between printing jobs. Some may be desperate to save paper, thus want 0 headers. Generally, the default of 1 page works best.

Example

```
) CX  PAUSE  @LPB1 ↵
) CX  HEADERS  @LPB1  2 ↵
) CX  CONTINUE  @LPB1 ↵
)
```

HOLD

Suspends a batch stream or spool queue entry until you issue an UNHOLD or CANCEL command.

```
CX  HOLD  { sequence-number }
          { username        }
```

where

sequence-number — is the sequence number of a job in the queue, displayed by the CLI command QDISPLAY.

username — is the name of the user who submitted the request.

The HOLD command keeps any batch stream or device from processing the specified queue entry until you issue an EXEC UNHOLD or CANCEL command. If you specify a sequence-number, EXEC holds all queue entries with that number. If you specify a username, EXEC holds all queue entries with that user's name.

A user can hold or release (unhold) his or her own queue requests with the CLI QHOLD and QUNHOLD commands. If a request is held by both an EXEC HOLD command and a user's CLI QHOLD command, the EXEC won't process the request until both hold flags are cleared.

The CLI QDISPLAY command lists queue entries and their sequence numbers. Active entries appear with an asterisk (*) in the display. (You cannot HOLD an active entry; to stop it, CX FLUSH it.) Entries held by the system operator appear with an E in the display. Entries held by users appear with H in the display.

Why Use It?

Sometimes you will have doubts about a request — perhaps not want to cancel it, but want to think about it. Use HOLD in such situations.

Example

```
) CX  HOLD  26 ↵
) CX  HOLD  OP ↵
```

These commands hold requests with sequence number 26 and those submitted by username OP.

LIMIT

Enforces user- or operator-specified limits of CPU time (batch) or number of printed pages (spool).

CX LIMIT [*stream [hh:mm:ss]* / *@devicename [pages]*]

where

stream	is a batch stream number; e.g, 1, 2, 3, or 4. If you omit arguments, the command imposes a CPU limit of 34:24:32 (considered infinity) on all batch streams.
hh:mm:ss	is the maximum amount of CPU time in hours, minutes, and seconds that a request in the batch stream(s) may use. The legal range is 0:00:00 to 34:24:32 (infinity).
devicename	is the name of a spooled device; it must begin with @; e.g., @LPB.
pages	is the optional maximum number of pages that a request may print. "pages" are projected by EXEC, unless users specify them with the /PAGES switch.

The LIMIT command enforces operator- or user-specified limits on CPU time (for batch streams) or printed pages (for printing requests).

If limiting is not in effect, EXEC processes all batch and spool requests on a first-come, first-served basis, based on priority. EXEC ignores user-defined limits on jobs. Each batch stream can use up to 34:24:32 of CPU time (considered infinity); each printer can print up to 65,535 pages per request.

When you plan to enforce limiting, you should inform users when the LIMIT command is in effect, so they can decide how to submit their requests.

You can undo the effect of the LIMIT command with EXEC's UNLIMIT command. Any LIMIT command remains in effect until you change it, undo it with UNLIMIT, or until EXEC terminates.

Batch Stream Limiting

Batch stream limiting without a CPU time limit has some value. If you specify limiting without a time limit; e.g.,

```
) CX  LIMIT )
```

then EXEC sets the default limit of 34:24:32. By itself, this is practically meaningless. But users can give it meaning by specifying their own limits with QBATCH or QSUBMIT switch /CPU=. A user might want to do this if he feared his request would loop — monopolizing the CPU — or if your organization's rules specified short batch times. For example, a user might type

```
) QBATCH/CPU=1:00  XEQ  MYPROG )
```

to make sure his job didn't consume more than a minute of CPU time. If stream limiting is enabled and a user specifies /CPU=, EXEC will abort the job if it consumes more than the user-specified limit.

LIMIT (continued)

A more useful form of limiting involves specifying a time limit. You can specify different limits for different streams; for example

) CX LIMIT (1,2) 1:00 ↵	(1 minute of CPU time for streams 1 and 2)
) CX PAUSE 3 ↵	(Pause stream 3.)
)	(Leave stream 4 alone; no limits.)

This sequence restricts streams 1 and 2 to "average" requests, and takes stream 3 out of the action; stream 4 remains unlimited for big requests. Generally, you would want to let users know about this so — for big requests — they could use the /CPU= switch.

If users don't use the /CPU= switch, and their jobs are accepted by stream 1 or 2 and take more than a minute of CPU time, the jobs will be aborted — a waste of time for both system and user. But if users *do* use the /CPU= switch, EXEC will place requests with /CPU=1:00 or more in stream 4, and place others in stream 1 or 2.

Generally, for any form of batch stream limiting to work well, users must participate by using the /CPU= switch when they submit large batch requests.

To find out if stream limiting is enabled, and the maximum CPU limit, use EXEC's STATUS command.

Device Page Limiting

For every printing request, EXEC projects the number of printed pages as follows.

pages = (number-of-bytes-in-file / 1000) + 4

(It uses integer division.) This is the number of pages you see displayed from a QDISPLAY/V command, for requests submitted without the /PAGES switch.

If you limit a device without a pages limit, and users omit the /PAGES= switch from their QPRINT commands, EXEC imposes its own projected number of pages, instead of the number actually in the file. As in batch, limiting without a value is not particularly useful unless users participate. For example, if you say

) CX LIMIT @LPB ↵

and users type QPRINT commands without the /PAGES= switch, EXEC will refuse the request if the number of pages it projects is less than the number actually in the file (their printed files will show a *PAGE LIMIT EXCEEDED* error message). If users *include* the /PAGES= switch, EXEC will print only the number of pages they specify in /PAGES=. This might be useful in some situations — but basically, whether or not users include /PAGES=, limiting a printer without giving a pages limit has little value.

Limiting a printer can be useful, though, if you specify a page limit — especially if you have more than one printer. For example, assume you have two printers. You can restrict one printer to small printing jobs:

) CX LIMIT @LPB 40 ↵

and leave the other printer unlimited. Users can post small printing jobs to @LPB (QPRINT command) and large ones to @LPB1 (QPRINT/QUEUE=LPT1 command).

Even with only one printer, device limiting with a page limit can be useful. If users append the /PAGE= switch, and their specified number of pages exceeds the LIMIT, EXEC will hold their requests until a printer that allows their /PAGE= figures becomes available. For example, if you say

```
CX   LIMIT   @LPB   50 ↵
```

and user F77 wants to print a big file:

```
) QPRINT/PAGES=400   PRODUCT_SUMMARY ↵
```

EXEC will hold F77's request until @LPB is allowed to print 400 pages. The request can run as soon as you unlimit @LPB or raise its limit to 400 or more. This allows you to favor small or medium requests — perhaps, again, based on time of day.

As with batch, device limiting requires user participation, via the /PAGES= switch.

To find out whether a device is limited, and the maximum pages (if any), use EXEC's SPOOLSTATUS command. You can check EXEC's projected (or the user specified) number of pages with the CLI command QDISPLAY/V.

Why Use It?

LIMIT gives your site control over requests *before* they are processed; and it can help make the system more efficient by dedicating streams or devices to small or medium jobs. You can use it in conjunction with EXEC's QPRIORITY and XBIAS commands to fine tune your system's batch and spooling mechanism, perhaps based on the time of day. You can change the emphasis quickly — for example, you can impose limiting only during the day, and lift it for big batch runs at night.

Also, LIMIT can help prevent program errors from slowing down the system; for example, it can prevent looping batch or runaway printing jobs from monopolizing the CPU or printer.

Example

```
) BROADCAST     Batch streams 1-3 limited to 1:00 of CPU time. & ↵
&)  stream 4 unlimited. & ↵
&)  Big batch users please use /CPU=0:10:0 to submit jobs. ↵
.
.
.
) CX   LIMIT   1:00 ↵
) CX   UNLIMIT   4 ↵
) CX   STAT ↵
.
.  (hours pass while streams run)
.
) BROADCAST   Am unlimiting all streams. ↵
) CX   UNLIMIT ↵
)
```

This sequence tells users about the impending LIMIT. (The BROADCAST macro is described in Chapter 5.) The LIMIT commands impose the default CPU time limit on all streams and a 10-minute limit on stream 4. The STATUS command verifies limiting. Later, operator tells users about the impending unlimit, and unlimits all streams.

LIMIT (continued)

```
) SEND   @CON-   Main printer limited to 50 pages, & ↵
&)   for fast action print only small files, or print large & ↵
&)   files and wait. Use QPRINT/PAGES= to this printer. ↵
.
.
.
) CX   LIMIT   @LPB   50 ↵
) CX   SPOOLS   @LPB ↵
.
.  (printer prints only those files with less than 50 pages...)
.
) SEND   @CON-   Am unlimiting main printer. ↵
.
.
.
) CX   UNLIMIT   @LPB ↵
```

Here, the operator tells users about the impending printer limit, limits the main printer to jobs of 50 EXEC-computed pages, then unlimits the main printer.

LOGGING

Starts or stops EXEC logging to a file.

```
CX  LOGGING  [ /START[/MAX=blocks] [pathname] ]
             [ /STOP                           ]
```

where

blocks is the maximum number of 512-byte disk blocks you want the logging file to be. This must be between 1 and 32,767.

pathname is the pathname of the file to receive EXEC messages. If you enter a filename only, EXEC will create the file in the directory @ (:PER) and use it as the log file. If the file doesn't exist, EXEC will create it. If it already exists, EXEC will append to it.

If you omit arguments, EXEC tells you whether or not logging is on, and, if so, the pathname of the logging file.

If you use the /START switch, EXEC starts logging all its console messages to the file indicated by *pathname*. The optional /MAX=n switch allows you to limit the size of the logging file. If the file exceeds the maximum size limit, EXEC goes back to the beginning of the file and overwrites the information with current log messages. If you omit /MAX=, EXEC doesn't limit the file's size.

The /STOP switch tells EXEC to stop logging messages; it turns off LOGGING.

If you are logging messages to a console, don't disable the console without first turning off LOGGING. If you disable a console log file, logging will stop.

You can write text strings to EXEC's log file with the CX MESSAGE command.

Why Use It?

An EXEC log can help you keep track of batch and print queue usage, and user MOUNT requests. (But note that EXEC's log file is *different* from the AOS SYSLOG log file. SYSLOG generally is more useful for user accounting information. EXEC puts account-oriented information into SYSLOG, not the EXEC log. SYSLOG is described in Chapter 9, "Other Runtime Tools."

Example

```
) CX   LOGGING/START   :UTIL:EXEC_LOG_MAY.85 ↵
   .
   .
   .
) CX   LOGGING/STOP ↵
) QPRINT   EXEC_LOG_MAY.85 ↵
   .
   .
   .
```

LPP

Sets the number of lines per page (LPP) for a device.

CX LPP @devicename number

where

devicename — is the name of a spooled device; it must begin with @.

number — is the number of lines per page. This must be an integer between 6 and 144.

The LPP command sets a new number of lines per page for this device. It overrides any previous AOSGEN, CHARACTERISTICS/CPL, or EXEC LPP settings. The new LPP remains until you change it with EXEC's LPP or bring down EXEC's XLPT co-operative process.

To change an LPP setting:

- Type CX PAUSE @devicename ↵
- When EXEC says the device is paused, type the desired CX LPP command.
- Type CX CONTINUE @devicename ↵

LPP does nothing on a device where an EXEC DEFAULTFORMS or FORMS command is in effect.

Why Use It?

The default number of lines per page is 66. You may want to specify fewer lines; perhaps to produce more white space for greater readability. Or, you may want to print a special form in the printer (although you'd usually use EXEC's FORMS or DEFAULTFORMS command for this).

Example

) CX PAUSE @LPB1 ↵

FROM PID 3 : (EXEC) @LPB1 PAUSED [IDLE]

) CX LPP @LPB1 60 ↵

) CX CONTINUE @LPB1 ↵

)

MESSAGE

Sends a message to EXEC's log file.

CX MESSAGE message

where

message is a text string to be entered in EXEC's logging file.

If EXEC is currently logging, the MESSAGE command sends the message to EXEC's log file. EXEC will always echo the message on your console.

The maximum length of the MESSAGE command is 80 characters. Therefore, the text string itself can't exceed 70 or so characters (with space for MES↵). However, you can type additional message commands if you want to say more.

If EXEC is not logging messages, it will not record the message in a log file, nor will it report an error.

Why Use It?

If EXEC is logging messages, you may want to put text messages in its log file for the record.

Example

) CX MESSAGE Running 4 batch streams 11-MAY-85 10am. ↵

FROM PID 3 : (EXEC) Running 4 batch streams 11-MAY-85 10am.

)

MOUNTED

Tells EXEC that a tape is physically on a unit and on line.

CX MOUNTED *[/MID=mid] [@devicename] [volid]*

where

/MID=mid allows you to select any request in the mount queue. *mid* is an integer — the mount ID of the desired request.

devicename is the magnetic tape unit name; it must start with @; e.g., @MTB1.

volid is the volume identifier of an existing tape in the mount request.

The MOUNTED command tells EXEC that you have physically placed a tape on a specific tape unit, and that the unit is on line.

Normally, EXEC processes mount requests on a first come, first served basis. If you want EXEC to service the next request, you need type only CX MOUNTED @devicename↵ — after putting the tape on the unit. If this is the second or subsequent volume of a request and you are using the same unit, you need type only CX MOUNTED↵.

If you want EXEC to select a request that is not the next request, include */MID=mid*, where *mid* is the mount ID of the request you want.

The MOUNTED command has EXEC create a linkname in the user's initial working directory, to the tape unit (unlabeled tape) or file @LMT:volid (labeled tape). EXEC sets the unit's ACL to username,WARE or username,RE (depending on whether the user said /READONLY on his or her mount command). This prevents anyone else from using the unit.

The unit(s) assigned via the MOUNTED command "belong" to the user until you issue a DISMOUNTED command. Usually EXEC will prompt you for this.

EXEC will prompt you to type the MOUNTED command:

- whenever a user types (or batch job submits) an error-free MOUNT command;
- when the system is ready for the next tape in a multivolume request and you haven't premounted this tape;
- if you mount the wrong volume in a labeled tape request;
- when a user has made an implicit mount request (e.g., LOAD/V @LMT:FOO:SOURCES↵.

With EXEC's UNIT MOUNT prompt, there may be a REQUEST IS line with additional information from the user. The REQUEST line will not appear on an implicit mount request.

Why Use It?

EXEC's MOUNTED command and users' CLI MOUNT commands are the heart of tape handling in a multiuser system.

You must type CX MOUNTED ↵, when prompted, to allow the user to access the tape.

Example

For an unlabeled mount, SACKVILLE types a CLI MOUNT command without a volid switch. The system console displays

*FROM PID 4: (EXEC) ** UNIT MOUNT ***
FROM PID 4: MID=94,USER=SACKVILLE, PID=14
FROM PID 4: REQUEST IS 'Any blank tape. Ring in please.'
FROM PID 4: RESPOND: CONTROL @EXEC MOUNTED @UNITNAME
FROM PID 4: OR: CONTROL @EXEC REFUSED

EXEC will repeat this message at intervals until you respond. The CLI prompt won't return to SACKVILLE's console until you *do* respond. So you mount a tape on MTB0 and type

```
) CX   MOUNTED   @MTB0 ↓
```

and the user can use the tape.

A labeled tape user asks for a multivolume labeled tape mount, knowing which volids to specify. EXEC prompts you

*FROM PID 4: (EXEC) ** UNIT MOUNT ***
FROM PID 4: MID=95,USER=COBOL, PID=32
FROM PID 4: VOLID(S) ARE: DB0033, DB0034, DB0035
FROM PID 4: REQUEST IS 'Rings in. Hi-density please.'
FROM PID 4: RESPOND: CONTROL @EXEC MOUNTED @UNITNAME
FROM PID 4: OR: CONTROL @EXEC REFUSED

Let's say you mount the tapes on MTB0, MTB1, and MTB2 and decide to use the PREMOUNT command. You type

```
) CX   MOUNTED   @MTB0 ↓

) CX   PREMOUNTED   @MTB1   DB0034   COBOL ↓

) CX   PREMOUNTED   @MTB2   DB0035   COBOL ↓
```

The tape I/O will proceed. When it is done, the user will request a dismount and EXEC will prompt you to DISMOUNT the tapes. There are other MOUNT examples in the preceding section, "User Mount Requests".

MOUNTSTATUS

Displays status of each MOUNT and DISMOUNT request.

CX MOUNTSTATUS *[mid]*

where

mid is a mount identifier: the integer number of a mount request in the mount queue.

The MOUNTSTATUS command displays all mount or dismount requests, or displays the mount request with the specified mount identifier (mid).

If you omit an argument, EXEC reports each entry on a first come, first served basis with all dismounts coming before mounts. If there are no requests, EXEC says *NO MOUNT REQUESTS*.

Why Use It?

MOUNTSTATUS is the most convenient way to check existing mount requests. You'll be using it often. (To check the status of *tape units*, use EXEC's UNITSTATUS command.)

Example

) CX MOUNTSTATUS ↲

FROM PID 3 : (EXEC) NO MOUNT REQUESTS

Later on:

) CX MOUNTST ↲

*FROM PID 3 : (EXEC) ** UNIT DISMOUNT ***
FROM PID 3 : MID=5
FROM PID 3 : UNIT(S) ARE: @MTB1
FROM PID 3 : RESPOND: CONTROL @EXEC DISMOUNTED

*FROM PID 3 : (EXEC) **UNIT MOUNT** WRITE RING OUT*
FROM PID 3 : MID=6, USER=JR, PID=8, VOLID=GP0059
FROM PID 3 : VOLID(S) ARE: GP0059, GP0060
FROM PID 3 : REQUEST IS 'Can you mount both - hi density?'
FROM PID 3 : RESPOND: CONTROL @EXEC DISMOUNTED
FROM PID 3 : OR: CONTROL @EXEC REFUSED.

Here, one tape request is awaiting a dismount and another is awaiting a mount. EXEC repeats these messages at intervals even if you don't use the MOUNTSTATUS command.

OPEN

Opens a queue to users (opposite of CLOSE).

CX OPEN queuename

where

queuename is an existing spool queue you want to open.

The OPEN command opens queuename so that users may submit requests to it. You must create and open a queue before users can submit jobs to it. Use the CREATE and OPEN commands, respectively, to do this. Generally, an output queue must be started on a device, and the device must be continued, before users can access the queue.

Actually, OPEN tells EXEC to create a queuename entry in directory @ (:PER). Users access this queue via CLI commands like QSUBMIT, QBATCH, QPRINT, QPLOT, QPUNCH, or QFTA. User programs can also access the queue by name via program OPEN, WRITE, and READ statements.

The first time EXEC is brought up, someone must open EXEC's BATCH_INPUT, BATCH_OUTPUT, and BATCH_LIST queue; and someone must create and open the print and possibly plotter and networking queues. This was covered in Chapter 5 and at the beginning of this chapter.

The CLI command QDISPLAY describes queues and their open and closed status.

Why Use It?

Each queue must be opened before it can be accessed; this is usually done the first time EXEC is brought up. Each time you create a new queue you must open it.

Also, if you close a queue for any reason, you must open it to make it accessible to users.

Example

```
) CX OPEN LQP ⤶
) CX START LQP @CON25 ⤶
) CX CONTINUE @CON25 ⤶
```

This sequence opens, starts, and continues the letter-quality printer queue, LQP, on the printer attached to line @CON25.

OPERATOR

Tells EXEC that an operator is available or not available at the system console.

CX OPERATOR { ON
OFF }

An operator must be available (ON) for EXEC to process user mount requests or batch requests submitted with the /OPERATOR switch. If an operator is not on duty (OFF), EXEC will reject mount requests with an *OPERATOR NOT AVAILABLE* error message; it will hold the /OPERATOR batch requests in the queue until an operator is available.

The default operator status is OFF.

You (and any user) can tell whether the OPERATOR status is on or off with the !OPERATOR pseudo-macro. For example

) WRITE An operator is [!OPERATOR] duty. ↵

An operator is OFF duty.

)

EXEC's OPERATOR command has no relation to the CLI command OPERATOR. The latter command enables the CLI to write and read labeled diskettes, described in Chapter 9.

Why Use It?

Some EXEC-controlled functions require a person in the role of system operator. For example, if a user requests a tape mount, someone must be available to mount the tape. The OPERATOR command tells EXEC that such a person is available.

If your system is to process user tape mount requests, or batch requests submitted with the /OPERATOR switch, you must issue the CX OPERATOR ON ↵ command. Later, if you leave the system console and no one is available to handle such requests, you must issue the CX OPERATOR OFF ↵ command so that user requests will not wait in vain for operator service.

Example

) WRITE [!OPERATOR] ↵

OFF

) CX OPERATOR ON ↵

)

PAUSE

directs batch stream(s) or a device to stop processing requests.

```
CX  PAUSE  [ stream      ]
           [ @devicename ]
```

where

stream — is a batch stream number; e.g., 1, 2, 3, or 4. If you omit a stream number, the command affects all batch streams.

devicename — is the name of a spooled device; it must begin with @; e.g., @LPB.

The PAUSE command stops processing when the current request is finished. All incoming requests remain in the queue until you issue EXEC's CONTINUE command.

If you omit arguments, PAUSE pauses all batch streams. To pause a specific stream or device, give its name as an argument. If a stream or device is active (processing a request), EXEC tells you that the pause will occur after the current job is done. Issuing a PAUSE command to a paused device or batch stream has no effect.

Why Use It?

Often, you will want to stop batch or spool queue processing; for example:

- to suspend output before changing parameters (like priority) for a batch stream or device; or
- before shutting down EXEC.

Example

To pause all batch streams, you'd type

```
) CX  PAUSE ↵
```

FROM PID 3 : (EXEC) STREAM_1 WILL PAUSE AT END OF CURRENT JOB
FROM PID 3 : (EXEC) STREAM_2 PAUSED
FROM PID 3 : (EXEC) STREAM_3 PAUSED
FROM PID 3 : (EXEC) STREAM_4 PAUSED

.

FROM PID 3 : (EXEC) STREAM_1 PAUSED

To pause the main line printer, @LPB, type

```
) CX  PAUSE  @LPB ↵
```

FROM PID 3 : (EXEC) @LPB PAUSED

PREMOUNT

Tells EXEC you have mounted a volume before there was a specific request for it.

```
CX PREMOUNT [/IBM] @devicename volid username
```

where

/IBM	is the switch indicating that you want the volid argument converted from ASCII to EBCDIC before EXEC compares it to the volid on the tape.
devicename	is the magnetic tape unit; it must begin with @; e.g., @MTB2.
volid	is the volume id of the tape you're premounting.
username	is the name of the user you are premounting the tape for.

The PREMOUNT command forms an association between a volume and a unit on behalf of a user.

When you premount a volume, EXEC sets the ACL for that unit to null — meaning that no one but a SUPERUSER can use it. Then, when the user's process needs that volume, EXEC changes the ACL to username,WARE or username,RE (depending on whether the user requested /READONLY on the mount). Later, when you dismount the tape, EXEC restores the ACL to its setting before the premount.

If you premount a volume, and then change your mind before I/O to it occurs, type CX DISMOUNTED @unitname) to sever the connection set by premount. Then dismount the tape.

You cannot premount a volume for implicit mount requests. You must use the MOUNTED command for these.

Why Use It?

PREMOUNT can be a big timesaver if you have more than one tape unit. When a user requests a multivolume mount, you can MOUNT the first volume on a unit, then PREMOUNT the second and subsequent volumes on other units.

The user process can then access all tapes as needed without operator intervention — even if an operator is not physically present. If you didn't use PREMOUNT, you'd have to mount each tape sequentially, as prompted by EXEC.

Example

The following command tells EXEC that you've premounted tape volume DB1103 on unit MTB10 for user SORT:

```
) CX PREMOUNT @MTB10 DB1103 SORT )
```

There is another PREMOUNT example under MOUNT.

PRIORITY

Changes the system priority and process type for batch steams and spooler processes.

CX PRIORITY [*/PREEMPTIBLE* | */RESIDENT* | */SWAPPABLE*] { stream | @devicename } priority

where

/PREEMPTIBLE	makes the process type pre-emptible.
/RESIDENT	makes the process type resident.
/SWAPPABLE	makes the process type swappable. This is the default.
stream	is the batch stream number; e.g., 1, 2, 3, or 4.
priority	is an integer specifying the priority of the process. The range is 1 to 255 for pre-emptible and resident processes; it is 1 to 3 for swappable processes. 1 is highest, 255 lowest.
devicename	is the name of a spooled device; it must begin with @; e.g., @LPB.

The PRIORITY command changes the system priority for batch streams and spooler cooperative processes like XLPT. The default priority for batch streams is 2. The default priority for devices is 3.

To change the process *type*, include the switch /PREEMPTIBLE or /RESIDENT. /SWAPPABLE is the default for batch streams and device processes.

To check the current process type and system priority, use CX STATUS for batch streams; or use CX SPOOLSTATUS for device processes.

Why Use It?

Sometimes, you may want to *favor* printing requests over other processing; if so, give a line printer a high priority, like 1. To favor printing even more, make it pre-emptible or even resident. The same principle applies to batch streams — you can give certain streams higher priority and/or make them pre-emptible or resident. (But if you give many processes a high priority — or make them pre-emptible or resident — system performance may suffer.)

If you settle on specific nondefault process type and priority values, put the EXEC commands for them in the UP.CLI macro.

EXEC's PRIORITY command combines the functions of the CLI commands PRIORITY and PRTYPE. And it is tailored for EXEC operations. Also, any CLI process with the username of OP can issue it.

Example

```
) CX   STATUS ↓
```

...(batch stream status information, including type and priority)...

```
) CX   PRIORITY/RES   1   1 ↓
) CX   PRIORITY   4   255 ↓
```

```
) CX   SPOOLSTATUS   @LPB ↓
```

...(spooler status information, including type and priority)...

```
) CX   PRIORITY/PREEMPTIBLE   @LPB   2 ↓
)
```

These commands check stream status with CX STATUS; then make stream 1 resident with a priority of 1 and give stream 4 the lowest priority, 255. Other streams retain the default types and values. Then the operator checks the spoolstatus of @LPB, and makes it pre-emptible with a priority of 2.

PROMPTS

Turns EXEC's time prompt off or on.

CX PROMPTS { ON | OFF }

If you set PROMPTS to OFF, only the CLI prompt will appear when EXEC is ready for a command. If PROMPTS is on (default), EXEC displays the prompt

FROM PID n : (EXEC) hours: minutes: seconds

after each command.

Why Use It?

Sometimes EXEC's time-of-day prompt can be distracting, especially on a hardcopy console. On the other hand, you may like the prompt as an acknowledgement of your commands.

Example

```
) CX   PROMPTS   OFF ↵
.
.
.
) CX   PROMPTS   ON ↵
```

FROM PID 3 : (EXEC) 15:33:30

)

EXEC displays the time of day in hours (24-hour clock), minutes, and seconds.

PURGE

Deletes all entries in a stopped and closed queue.

CX PURGE queuename

where

queuename   is the spool queue you want to purge.

The PURGE command deletes all entries in the specified queue. After the command succeeds, the queue is empty. The queue must not be processing a job on any device when you issue this command. It must be closed and stopped before it can be purged. Use EXEC's CLOSE and STOP commands to do this.

Why Use It?

Sometimes, you'll want to clean out a queue — for example, if it has a lot of useless requests or if you want to delete the queue. Use PURGE in such situations.

To get rid of an individual *request* in a queue, use EXEC's CANCEL or FLUSH instead of PURGE.

Example

To stop, purge, and delete a queue named PRINTER:

```
) CX PAUSE @LPB ↵
) CX CLOSE PRINTER ↵
) CX STOP PRINTER ↵
) CX PURGE PRINTER ↵
) CX DELETE PRINTER ↵
) CX CONTINUE @LPB ↵
)
```

QPRIORITY

Changes the priority range to be accepted by batch stream(s) or a device.

CX QPRIORITY [*stream* | *@devicename*] [*high-value* *low-value*]

where

stream	is a batch stream number; e.g., 1, 2, 3, or 4.
devicename	is the name of a spooled device; it must begin with @; e.g., @LPB.
high-value	is the highest priority value of a request that the stream or device can accept. It must be an integer between 0 (highest) and 255 (lowest).
low-value	is the lowest priority value of a request that the stream or device can accept. It must be an integer from 0 to 255.

The QPRIORITY command displays or changes the range of priority that a stream or device will process.

If you omit arguments, EXEC displays the queue priority for all batch streams. If you specify a stream number or device name, EXEC displays its queue priority.

Users specify queue priority with the /QPRIORITY switch. If a user omits a /QPRIORITY= switch when submitting a request, EXEC assigns the request a queue priority. This is the median of the highest queue priority and the lowest queue priority indicated by the user's profile. For example, if a user's highest queue priority is 0 (PREDITOR default), then his requests are assigned a queue priority of 127, which is halfway between 0 and 255.

Do not confuse QPRIORITY with a *process'* priority, set with the CLI command PROCESS or PRIORITY, or EXEC's command PRIORITY.

Why Use It?

Your installation might want to have requests processed in different queues, depending on each request's priority. For example, you could establish different queue priority ranges for two batch streams. This would give users some control over queue processing. Users may request a certain queue priority with the /QPRIORITY= switch.

If a user's MAX QPRIORITY, set in the profile during the PREDITOR dialog, is less than (farther from 0) other users' priorities, his printing requests will have less priority than those of other users. This is so regardless of the EXEC QPRIORITY command.

Example

```
) CX  QPRIORITY ↵
.
.
.
```

...(EXEC displays current batch stream QPRIORITY)...

```
) CX  QPRIORITY  1  0  100 ↵
) CX  QPRIORITY  (2,3)  101  255 ↵
```

After this sequence, requests submitted with QBATCH/QPRIORITY=0 through QBATCH/QPRIORITY=100 will go to batch stream 1. Requests submitted with QBATCH/QPRIORITY=101 through QBATCH/QPRIORITY=255 will go to streams 2 and 3. Simple QBATCH requests, which generally get priority 127, will also go to streams 2 and 3. Stream 4, if continued, will accept all requests.

For two line printers:

```
) CX  QPRIORITY  @LPB  0  126 ↵
) CX  QPRIORITY  @LPB1  127  255 ↵
```

This works the same way as the batch streams described above, except that users use the QPRINT command instead of QBATCH.

REFUSED

Tells EXEC and a user that you refused a MOUNT request.

CX REFUSED [*mid*]

The REFUSED command tells EXEC to cancel a user mount request (for whatever reason). It also sends the message *REQUEST REFUSED BY SYSTEM OPERATOR* to the user who typed the MOUNT command. The *mid* is the mount ID — you can omit it for the last (or only) request.

Why Use It?

There may be times when you want to cancel a mount request. This can be your own mount request, or another user's. For the latter, you can also use the SEND command to explain why you refused.

Example

*FROM PID 3 (EXEC): ** UNIT MOUNT ***
FROM PID 3 (EXEC): MID=50, USER=DAVE, PID=44, EXEC SUB-TREE PID=44
FROM PID 3 (EXEC): REQUEST IS: HIGH DENSITY- PLEASE

.
.
.

```
) CX   REFUSED ↲
) SEND   44   Sorry Dave -- backup time. Try again in 2 hours. Thanks. ↲
```

The following messages appear on Dave's terminal:

ERROR: REQUEST REFUSED BY SYSTEM OPERATOR
Sorry Dave -- backup time. Try again in 2 hours. Thanks.

RESTART

Restarts output of the current file on a device.

CX RESTART @devicename *[start-page]* *[end-page]*

where

devicename	is the name of a spooled device; it must begin with @; e.g., @LPB1.
start-page	is the first page of the file to be restarted; it must be an integer between 1 and 65535.
end-page	is the last page of the file to be printed. It must be an integer, larger than the *begin-page*, and be between 1 and 65535.

The RESTART command restarts output of the current file on the specified device. You can specify beginning and ending page numbers for printing. If you omit page numbers, EXEC's XLPT process uses the default start and end page numbers 1 and 65535, respectively. If you specify only one number, XLPT uses it as the start page number and uses the default value as the end page number.

If the user included a /NORESTART switch when submitting the request, the RESTART command will not work. The printed file will contain a *CANNOT RESTART, NORESTART SPECIFIED* message.

You can restart a file any time you want. The XLPT process always writes the current output buffer before restarting the file.

Why Use It?

EXEC's RESTART command is similar to its ALIGN command, except that the job in the printer is returned to the queue. You could then hold it for printing on another device.

Example

To restart printer @LPB and print the entire current file:

```
) CX  RESTART  @LPB ↵
```

To restart from page 140:

```
) CX  RESTART  @LPB  140 ↵
```

SILENCE

Suppresses EXEC messages about batch stream(s) or a device.

```
CX  SILENCE  [ stream      ]
             [ @devicename ]
```

where

stream is a batch stream number; e.g, 1, 2, 3, or 4. If you omit a stream number, the command affects all batch streams.

devicename is the name of a spooled device; it must begin with @; e.g., @LPB.

The SILENCE commands tells EXEC not to send messages when a job is queued in a batch stream or processed on a device. By default, EXEC prints BRIEF messages, described under EXEC's BRIEF command.

If you omit an argument, EXEC suppresses messages about all batch streams. If you specify a stream or device, EXEC suppresses messages from the stream or device.

SILENCE also suppresses messages to EXEC's log file, if EXEC logging is on.

To restore message output, use the CX UNSILENCE command.

Why Use It?

If your system console is a hardcopy console, EXEC messages can tie it up and make it difficult to issue commands. Or, you may want to use the system console for something like text editing, and want to avoid distractions. Use CX SILENCE in either of these situations.

Example

```
) CX  SILENCE ↓
) CX  SILENCE  @LPB ↓
```

...(no EXEC batch or device messages to system console)..

```
) CX  UNSILENCE ↓
) CX  UNSILENCE  @LPB ↓
```

These commmands silence all batch streams and printer @LPB; then the UNSILENCE command restores message output.

```
) CX  SILENCE  4 ↓
```

SPOOLSTATUS

Displays queue-device status information.

CX SPOOLSTATUS [*queuename* / *@devicename*]

where

queuename is a spool queue.

devicename is the name of a spooled device; it must begin with @; e.g., @LPB.

The SPOOLSTATUS command describes the devices used by a queue, the queues using a device, or both. If you omit arguments, e.g.,

) CX SPOOLS ↵

the command tells you

- each spooled devicename and the queuename(s) associated with it;
- the CPL, LPP, HEADERS, and TRAILERS settings;
- whether limiting is enabled, and what the limits are, if not the defaults;
- whether or not even pagination or binary mode is enabled;
- if a DEFAULTFORMS command is in effect;
- if a FORMS command is in effect;
- the device bias factor, set with EXEC's XBIAS command;
- the priority range of the device as set by EXEC's PRIORITY command.

If you use a queuename argument, SPOOLSTATUS tells you which device(s) the queue is associated with.

If you use a devicename argument, SPOOLSTATUS gives most of the information above.

Why Use It?

SPOOLSTATUS is your primary status checking command for device queues (as STATUS is for batch streams and devices). You will use it often, along with the CLI's QDISPLAY, to see what's happening with queues.

You'll probably use SPOOLSTATUS routinely before and after changing a device specification (as with CPL or FORMS).

Example

) CX SPOOLSTATUS ↵

.
. (information)
.

) CX SPOOLSTAT LPT1 ↵

FROM PID 3 : (EXEC) LPT1 BEING PROCESSED BY @LPB1

STACK

Tells EXEC to read card images from an input device or file.

```
CX  STACK  { @devicename }
           { pathname    }
```

where

devicename — is the name for the device containing the stacked job; e.g., @CRA. There may be more than one job in stacked format.

pathname — gives the disk filename that holds the card image records.

The STACK command tells EXEC to stack (enqueue) batch requests from an input source — usually a card reader, but it can be a disk file. EXEC creates a co-operative Stacker process to read the cards and submit the requests.

The format of input cards for each request is shown earlier in this section.

Why Use It?

You must issue STACK if you want EXEC to read cards as batch requests from a card reader. (If the card images are already on disk, it's possible to text edit the disk file, then submit it as a batch input file with the CLI's QBATCH/M command, instead of using the STACK command.)

Example

Place cards in reader, turn it on.

```
) CX  STACK  @CRA ↵
```

FROM PID 3 : (EXEC) @CRA CO-OPERATIVE INITIATED
FROM PID 3 : (EXEC) SEARCHING FOR $$JOB
FROM PID 3 : (EXEC) @CRA-$JOB USER 1
FROM PID 3 : (EXEC) @CRA-JOB ENQUEUED
FROM PID 3 : (EXEC) @CRA-END OF FILE ON INPUT
FROM PID 3 : (EXEC) @CRA CO-OPERATIVE TERMINATED

Remove cards from reader.

If the *END OF FILE ON INPUT* message doesn't appear before the *CO-OPERATIVE TERMINATED* message, an error occurred — probably an invalid job card.

STATUS

Describes the status of batch streams or devices.

CX STATUS [*stream* / *@devicename*]

where

stream is a batch stream number; e.g., 1, 2, 3, or 4. If you omit a stream number, the command affects all batch streams.

devicename is the name of a spooled device; it must begin with @; e.g., @LPB.

The STATUS command tells you the active or paused status of batch stream(s) or a device. If the stream or device is active, EXEC displays the

- sequence number;
- user's queue priority;
- username;
- process ID (PID);
- pathname of the source file;
- bias factor as defaulted or set by EXEC's XBIAS command;
- process type and process priority as defaulted or set by EXEC's PRIORITY command; and
- current page being processed and the number of copies left to print (active printers only).

If the device or stream is not processing a job, EXEC displays an *IDLE* message, then describes the EXEC bias factor, process type, and priority.

If you omit arguments, EXEC displays the status of all the batch streams. If you use a stream number or devicename argument, EXEC displays the status of that stream or device.

Why Use It?

STATUS is the primary status checking command for batch streams and devices. (To check device *queues*, use EXEC's SPOOLSTATUS command.) You'll use the STATUS command often, along with the CLI's QDISPLAY command, to see what's happening with batch queues and devices.

You'll probably use STATUS routinely before and after changing a batch queue specification (as with QPRIORITY); you'll also use it to see what's going on when you plan to shut down EXEC.

The additional information given by STATUS @devicename can be handy when you want to align a device or see how long the current request will take.

Example

) CX STAT ↵

FROM PID 3 : (EXEC) STREAM_4 [IDLE]
FROM PID 3 : BIAS FACTOR=0 PTYPE=SWAPPABLE PRIORITY=2

FROM PID 3 : (EXEC) STREAM_3 [IDLE] PAUSED
FROM PID 3 : BIAS FACTOR=0 PTYPE=SWAPPABLE PRIORITY=2

FROM PID 3 : (EXEC) STREAM_2 [IDLE]
FROM PID 3 : BIAS FACTOR=0 PTYPE=SWAPPABLE PRIORITY=2

FROM PID 3 : (EXEC) STREAM_1 SEQ=3650 QPRI=127 USER=ROGER
FROM PID 3 : PID=37 PATH=:UDD:DATABASE:?029.CLI.004.JOB
FROM PID 3 : BIAS FACTOR=0 PTYPE=SWAPPABLE PRIORITY=2

This is a status report on all streams. Streams 4 and 2 are idle; stream 3 is paused and idle; and stream 1 is active.

) CX STAT @LPB1 ↵

FROM PID 3 : (EXEC) @LPB1 SEQ=3646 QPRI=127 USER=MAG
FROM PID 3 : PID=6 PATH=:UDD:MAG:RELEASE_1.50:MYFILE
FROM PID 3 : CURPAGE=46 COP LEFT=2

This status report on @LPB1 shows the current page (CURPAGE) and copies left to print (COP LEFT), after other things.

STOP

Dissociates a queue from a device (opposite of START), stops a device, or does both.

```
              ⎧ queuename                  ⎫
CX   STOP     ⎨ @devicename                ⎬
              ⎩ queuename   @devicename    ⎭
```

where

queuename — is the the name of a queue you want to dissociate from its device(s).

devicename — is the name of a spooled device; it must begin with @; e.g., @LPB1.

The STOP command dissociates a queue from a device, stops a device, or does both — depending on the form you use.

- CX STOP queuename dissociates the queuename from all associated devices. If this is the only queue associated with a device, the device stops and the pertinent EXEC co-operative process (e.g., XLPT) terminates.
- CX STOP @devicename dissocates the device from *all* associated queues, and stops the device. The pertinent EXEC co-operative process terminates.
- CX STOP queuename @devicename dissociates *only the queuename* from the devicename. This is useful only if there are multiple queues associated with a device and you want to sever *only this* queue from the device; else you would use the first form above.

In all cases, the pertinent device will finish processing an active request. The queue will accept new requests, and will remain open. But the device won't process other requests in the queue (if any), until you issue the appropriate START command.

Why Use It?

You might use STOP when you want to stop a device and think about the current queue-device situation. Or, you might use it if you want to get rid of a queue (but you must also close and perhaps purge the queue before you can delete it).

If things are okay and you want an orderly shutdown, use EXEC's PAUSE command instead of STOP. To stop an active request, use EXEC's FLUSH command.

To prevent a queue from accepting new requests, use EXEC's CLOSE command.

Example

To dissociate queue LQP from its device:

```
) CX   STOP   LQP ↵
```

To dissociate all queues from a device and terminate EXEC's device-handling co-operative process:

```
) CX   STOP   @CON25 ↵
```

.
. (finishes current request)
.

FROM PID 3 : (EXEC) @CON25 CO-OPERATIVE TERMINATED

TERMINATE

Terminates the user process running on a console or a co-operative process associated with a device.

```
CX   TERMINATE   { @consolename }
                 { @devicename  }
```

where

@consolename	is the name of the console on which you want to terminate the process; it begins with @.
@devicename	is the name of a device whose co-operative process you want to terminate.

The TERMINATE @consolename command terminates the user process on the specified console and logs the user off. The console on which you type TERMINATE displays the message

CONSOLE JOB TERMINATED

and the user's console displays the messages

PROCESS n TERMINATED BY OPERATOR COMMAND
CONNECT TIME hours:minutes:seconds
USER user LOGGED OFF @CONNn
date time

You can terminate co-operative processes like XLPT, XPLT, or Hamlet, as well as console jobs. But you cannot use EXEC's TERMINATE to terminate a XODIAC File Transfer Agent (FTA) or SNA co-operative. Use the DOWN.NETWORK.CLI macro, supplied with XODIAC, to terminate XODIAC agent processes.

Why Use It?

A time may come when you must terminate a user process to protect other users or the system. You can use EXEC's TERMINATE command for this.

EXEC's TERMINATE mirrors the functionality of the CLI's TERMINATE. However, any process with the username OP can issue EXEC's TERMINATE — even if the target process is a brother process. A process must have Superprocess on to use CLI's TERMINATE to kill a brother process. An example follows.

Example

This dialog ultimately terminates the user process on CON5.

) SEND 13 Logoff NOW! ↵	Warn user.
...(a minute or so passes)...	Wait.
) WHO 13 ↵	See if PID still exists.
PID 13 CAIN CON5 :CLI.PR	Still running on console CON5.
) TERM 13 ↵	Try to kill it with CLI TERMINATE.
WARNING...PROCESS NOT IN HIERARCHY	Doesn't work...
) CX TERM @CON5 ↵	Try EXEC's TERMINATE...
CONSOLE JOB TERMINATED	EXEC's TERMINATE works.

TRAILERS

Changes the number of trailer sheets after each printed file.

```
CX  TRAILERS  @devicename  { 0 | 1 | 2 }
```

where

devicename	is the name of a spooled device; it must begin with @; e.g., @LPB1.
0, 1, or 2	are the number of trailer sheets you want.

The TRAILERS command sets the number of trailer sheets that follow the output for each printing request on the specified device. By default, EXEC provides no trailer sheet.

Information on each trailer sheet includes

- destination name;
- username;
- queuename;
- devicename;
- number of pages printed;
- system identifier (SYSID).

EXEC's SPOOLSTATUS command describes the number of trailers on a device.

As with the HEADERS command, the device must be paused before EXEC will accept the command; and you must continue it afterward.

Why Use It?

Your organization might want one or more trailer sheets between each file — for greater separation of printing jobs or to describe the number of pages printed (AOS' SYSLOG, described later, logs pages printed by each user).

Example

To set the number of trailers on @LPB1 to 1, type

```
) CX  PAUSE  @LPB1 ↵
) CX  TRAILERS  @LPB1  1 ↵
) CX  CONTINUE  @LPB1 ↵
```

UNHOLD

Negates a previous HOLD command.

CX UNHOLD { sequence-number | username }

where

sequence-number — is the sequence number of the request you want to unhold (release).

username — is the name of the user whose requests you want to unhold (release).

The UNHOLD command cancels a previous EXEC HOLD command. It does not cancel a *user's* QHOLD or QPRINT/HOLD command — the user must QUNHOLD this, or you can QCANCEL it. Use the CLI command QDISPLAY to check the hold status of a queue entry.

EXEC does not keep track of whether you held (HOLD command) a request by sequence number or username. So you can release any request from hold by using either its sequence number or username. The sequence number releases only the request with the sequence number; the username releases all requests with that username.

See EXEC's HOLD command for more information.

Why Use It?

You must use UNHOLD if you want EXEC to process a request that you previously suspended with the HOLD command.

Example

To remove the previous operator hold on all queue requests with the username OP:

```
) CX UNHOLD OP ↵
```

To remove the previous operator hold on request 588:

```
) CX UNHOLD 588 ↵
```

UNITSTATUS

Displays the status of one or all tape units.

CX UNITSTATUS *[@devicename]*

where

devicename is a magnetic tape unit; it must start with @; e.g., @MTB12.

Without an argument, the UNITSTATUS command displays each unit's status. With an argument, it describes a unit. If a unit is mounted, EXEC displays the following.

- the user name;
- user process ID (PID);
- the volid(s) for a labeled tape request;
- any premount on the unit.

Why Use It?

UNITSTATUS is better than MOUNTSTATUS if you are more interested in units than in mount requests. You might use it to decide which unit to use for a premount.

Example

```
) CX UNITSTATUS ↵
FROM PID 3 : (EXEC) @MTB0 NOT MOUNTED
FROM PID 3 : @MTB2 PREMOUNTED USER=ANDERS, VOLID=GP6423
FROM PID 3 : @MTB1 MOUNTED MID=40, USER=ANDERS,PID=9, VOLID=GP6422
.
.
.
) CX UNITSTATUS @MTB0 ↵
FROM PID 3 : (EXEC) @MTB0 NOT MOUNTED
```

UNLIMIT

Stops limiting on a queue or device (negates LIMIT).

CX UNLIMIT [*stream* / *@devicename*]

where

stream — is a batch stream number; e.g., 1, 2, 3, or 4. If you omit a stream number, the command affects all batch streams.

devicename — is the name of a spooled device; it must begin with @; e.g., @LPB.

The UNLIMIT command negates a previous EXEC LIMIT command. If you omit arguments, EXEC removes the CPU time limit on all batch streams. If you use a batch stream argument, EXEC removes the time limit for that stream. With a devicename argument, EXEC removes the page output limit for the device.

See EXEC's LIMIT command for more information.

Why Use It?

If limiting is enabled, you must use UNLIMIT to lift the limits. (To set new nondefault time/page limits, use another EXEC LIMIT command.)

Example

...(limiting has been enabled on batch streams)...

...(hours pass while streams run)...

```
.
.
.
) BROADCAST   I am unlimiting all streams. ↵
) CX   UNLIMIT ↵
)
```

...(limiting has been enabled on printer @LPB)...

```
) BROADCAST   I am unlimiting main printer. ↵
) CX   UNLIMIT   @LPB ↵
```

UNSILENCE

Restores EXEC display of batch/spool messages.

```
CX UNSILENCE [stream     ]
             [@devicename]
```

where

stream is a batch stream number; e.g., 1, 2, 3, or 4. If you omit a stream number, the command affects all batch streams.

devicename is the name of a spooled device; it must begin with @; e.g., @LPB.

The UNSILENCE command negates a previous EXEC SILENCE command. EXEC will then send a message to the system console each time it queues or processes a request. EXEC will also send a message to its log file if EXEC LOGGING is started. The contents of each message will depend on whether an EXEC VERBOSE or BRIEF command is in effect.

If you omit arguments, EXEC will send messages about all batch streams. If you enter a stream or devicename argument, EXEC will send messages about the stream or device.

Why Use It?

Having silenced EXEC, you way want to restore its batch or spool messages — especially if you want to log EXEC messages (LOGGING command).

Example

```
) CX SILENCE )
) CX SILENCE @LPB )
```

.
. (hours of batch and spool silence pass)
.

```
) CX UNSILENCE )
) CX UNSIL @LPB )
```

Here, the operator silences EXEC batch and spool messages, then restores them with UNSILENCE.

VERBOSE

Tells EXEC to give detailed batch or spool messages.

CX VERBOSE [*stream* / *@devicename*]

where

stream is a batch stream number; e.g., 1, 2, 3, or 4. If you omit a stream number, the command affects all batch streams.

devicename is the name of a spooled device; it must begin with @; e.g., @LPB.

When a batch stream or spooled device accepts or processes a request, EXEC sends a message to the system console. This message may be either "brief" or "verbose." The EXEC BRIEF and VERBOSE commands determine the verbosity of EXEC messages sent. Each BRIEF or VERBOSE command overrides the current message setting. BRIEF is the default setting.

VERBOSE messages include all BRIEF information, plus the user's process ID (PID) and pathname of the request's source file.

Why Use It?

There may be situations where you want to know the pathname of user request source files — especially if EXEC logging is started. Use VERBOSE for this.

Example

FROM PID 3 : (EXEC) STREAM_2 SEQ=446, USR=sackville

) CX VERBOSE ↓

FROM PID 3 : (EXEC) STREAM_1 SEQ=447, USR=sackville
FROM PID 3 : PID=16 PTH=:UDD:F77:NAVY_TEST:?016.CLI.002.JOB

Here, you can see the difference between the BRIEF (default, at top) and VERBOSE messages.

XBIAS

Sets EXEC's bias factor to favor either small or large requests.

```
CX  XBIAS  { stream      } bias-factor
           { @devicename }
```

where

stream	is a batch stream number; e.g., 1, 2, 3, or 4.
devicename	is the name of a spooled device; it must begin with @; e.g., @LPB.
bias-factor	is an integer, range -32767 through 32767. A positive bias factor favors small batch requests; a negative factor favors large requests.

The XBIAS command sets EXEC's bias for the specified stream or device to a new value. If the bias factor equals 0 (default), biasing is off.

A positive bias factor directs EXEC to favor small requests; the larger the number, the greater EXEC's bias for small requests. A negative factor has EXEC favor large requests; the larger the absolute value of the negative number, the greater EXEC's bias for large requests.

Normally, requests are processed in the order in which users submitted them, depending on priority. Biasing makes favored jobs appear to have been submitted earlier than they really were. When you specify a bias factor, EXEC multiplies it by the cost unit (CPU-seconds for batch, number of pages for devices) for each request. EXEC then adds this signed value to the actual time the request was submitted. EXEC then uses *this* time value when it chooses the next request; and it picks the request that appears to have been submitted earliest.

To check the current bias of batch streams, use EXEC's STATUS command; to check it for devices, use SPOOLSTATUS.

Don't confuse EXEC's XBIAS command with the CLI BIAS command, which pertains to interactive or noninteractive processing. The two are not related.

Why Use It?

Sometimes, perhaps during a certain period of each day, you might want to favor large or small requests. Or, you might want selected streams or devices to handle large requests, and have others handle small requests.

The bias factor allows you to control whether streams or devices favor small or large requests. To favor small requests, set the desired positive bias; to favor large ones, set the desired negative bias.

You can use XBIAS in conjunction with the other selective control commands (LIMIT, QPRIORITY, etc.) to tailor certain streams and devices for specific kinds of requests.

Example

To set EXEC's bias factor for batch stream 1 to 100, and for stream 4 to -100, type

```
) CX  XBIAS  1   100 ⤶
) CX  XBIAS  4  −100 ⤶
) CX  STATUS ⤶
```

...(EXEC displays XBIAS, among other things)...

XHELP
Explains EXEC Commands

XHELP *[command]*

where

command is an EXEC command.

If you omit *command*, XHELP describes all EXEC commands. If you include *command*, XHELP describes the command.

XHELP is the only command to EXEC that doesn't start with CONTROL @EXEC (CX).

Why Use It?

XHELP can be extremely useful if you forget a command or correct command syntax.

Example

```
) XHELP ↵
.
.  (describes all EXEC commands)
.
) XHELP   ALIGN ↵
.
.  (describes ALIGN command)
.
)
```

EXEC Messages

EXEC is a two-part program, serving both the system operator (username OP), and users. It has one set of messages for the operator, another for people who are logged on as users.

This section describes the messages that you — the operator — or users may receive from EXEC. It has two parts, Operator Messages and User Messages.

The message explanations assume the existence of the macro CX.CLI. CX.CLI should contain the characters

CONTROL @EXEC %-%

and be in directory :UTIL (or some easily accessible directory). If it doesn't exist, create it.

Operator Messages from EXEC

Table 8-4 explains the EXEC messages that are not self-explanatory. By default, EXEC sends all messages to the active CLI process that has the username OP. If there are multiple active OP CLIs (for example, if you have logged on as OP to a user console), response messages go to the console from which you typed the EXEC command. Some messages are printed on the batch output file or printed file. MOUNT messages *always go to the system console.*

EXEC also send its messages to the EXEC log file if you have enabled logging with the CX LOGGING command. If you have suppressed EXEC messages with CX SILENCE, no batch or spool informational messages will be sent to console or log file. MOUNT messages always appear, even if you have issued CX SILENCE.

This table includes a few pertinent *system* (not EXEC) error messages.

Table 8-4. Operator Messages from EXEC

Message	Description and Action
ABORT: text	An error message; appears on issuing console or batch output listing. *text* here is another message; check the message column of this table to find it. The text may be a system (not EXEC) error message with instructions; if so, follow them.
ALL CONSOLES ENABLED	Status message from EXEC ENABLE/ALL command; appears on issuing console. EXEC has enabled all user consoles for log on.
ARGUMENT UNKNOWN	Error message from EXEC command, appears on issuing console. Re-enter with correct argument.
ATTEMPT TO CREATE MORE THAN 32 QUEUES	Error message from EXEC CREATE command, appears on issuing console. EXEC allows a maximum of 32 queues, including the three permanent batch queues. If you need the queue, CLOSE and DELETE an old one.
AWAITING ALIGNMENT	Status message, from EXEC STATUS @LPBx command, appears on issuing console. A CX ALIGN command was issued and the printer is waiting. Align printer if needed, then issue `CX ALIGN/CONTINUE @LPBx ↵`.
BAD { $$JOB / $$PASSWORD } FORMAT	Error message from Stacker, after CX STACK command, appears on console that issued STACK. It may appear if the input file is a card reader; the process terminates without an error message if the input file is a disk file. EXEC rejects the job. To fix it, have the user correct the $$JOB-$$PASSWORD card sequence or correct it yourself; then restack the reader and re-enter the EXEC STACK command.
BAD USERNAME	Error message from EXEC's PREMOUNT command, appears on issuing console. Re-enter the PREMOUNT command the with correct username.
BIAS FACTOR = n	Status message from EXEC STATUS (for batch streams) or SPOOLSTATUS command (queues); appears on issuing console. n is 0 if it wasn't set via the XBIAS command.
BINARY MODE NOT ALLOWED	Error message from a user QPRINT/BINARY command, appears on printed file. You must enable binary mode with `CX BINARY @LPx cleanup-filename ↵` before anyone can QPRINT/BINARY.
CANNOT DELETE UNEXPIRED FILE ON LMT	System error message from write to labeled tape; appears on issuing console. The default retention period on files written to labeled tape is 90 days. Choose another tape or write a new label to the tape with the LABEL program; then retry the tape write.
CAN'T TURN OFF -- ACTIVE BATCH REQUEST REQUIRES OP	Error message from EXEC OPER OFF command; appears on issuing console. You can't turn operator off until you cope with the active batch job that was submitted with /OPERATOR switch.

(continues)

Table 8-4. Operator Messages from EXEC

Message	Description and Action
CAN'T TURN OFF -- OUTSTANDING MOUNT REQUEST	Error message from EXEC OPERATOR OFF command; appears on issuing console. You can't turn operator off until you cope with the active mount or dismount requests (use CX MOUNTSTATUS↵ to check).
CLEANUP FILE DOES NOT EXIST	Error message from CX BINARY @LPx cleanup-filename ↵ command; appears on issuing console. File cleanup-filename must be in directory :UTIL:FORMS. Create a cleanup file for the device with interactive program CLEANUP.PR.
COMMAND UNKNOWN	Error message; appears on issuing console. EXEC doesn't know your CX command. You may have miskeyed the command; use XHELP if needed.
CONSOLE ALREADY ENABLED	Error message from EXEC ENABLE command; appears on issuing console. This console *was* enabled. If desired, you can force new log-on parameters by adding the /FORCE switch in your ENABLE command.
CONSOLE DISABLED	Status message from EXEC DISABLE command; appears on issuing console. The console has been disabled. If on, it displays a *CONSOLE DISABLED FROM LOGGING ON* message.
CONSOLE ENABLED	Status message from EXEC ENABLE command; appears on issuing console. The console has been enabled. If on, it displays a *...TYPE NEW LINE TO BEGIN LOGGING ON* message.
CONSOLE JOB TERMINATED	Status message from EXEC TERMINATE command; appears on issuing console. The console process specified has been terminated and the user logged off.
CONSOLE NOT ACTIVE	Error message from EXEC TERMINATE command; appears on issuing console. The console process you tried to terminate didn't exist.
CONSOLE UNKNOWN TO EXEC	Error message from EXEC DISABLE or TERMINATE command; appears on issuing console. The specified console is not ENABLEd.
CONSOLE WILL BE DISABLED	Status message from EXEC DISABLE command; appears on issuing console. EXEC will disable the console as soon as it can. If the console is active, EXEC will disable it when the current user logs off.
CONTROL ENTRY ALREADY EXISTS	Error message from EXEC-creating PROCESS command; appears on system console. An entry named EXEC already exists in :PER. If an EXEC is running, everything is okay. If EXEC is *not* running, try to DELETE :PER:EXEC+↵ and try again.
CO-OPERATIVE INITIATED	Status message from EXEC START or STACK command; appears on issuing console. EXEC created the co-operative process needed to run the device.

(continued)

Table 8-4. Operator Messages from EXEC

Message	Description and Action
CO-OPERATIVE TERMINATED	May be a status message from the Stacker after a short-lived co-op process created by EXEC STACK terminates normally after QSUBMITTING the job. It is an error if a stacker process terminated without QSUBMITTING the job, or if a long-lived co-op like XLPT terminates without an EXEC STOP or TERMINATE command. If XLPT terminates, users won't be able to use the printer; you must recreate the process with EXEC's START command (see UP.CLI macro for syntax). This message generally appears on the system console.
COP LEFT=	Status message from EXEC status @LPBx command, appears on issuing console. Describes how many copies in the current job remain to be printed.
COULD NOT ENABLE CONSOLE	Error message from EXEC ENABLE command, appears on issuing console. EXEC could not enable this console.
COULD NOT RE-OPEN QUEUE	Error message from EXEC startup after PROCESS command; appears on system console. There may be an internal EXEC or system inconsistency; or file QUEUE:Q may have been deleted. Check QUEUE:Q; if it doesn't exist, create it as shown in Chapter 5.
COULDN'T ACCESS CODE FOR MESSAGE	Error message from EXEC command *error*; appears on issuing console. EXEC could not find needed text in error file :ERMES. Perhaps ERMES has been deleted or built incorrectly. Rebuild ERMES as described in Chapter 5.
CURPAGE =	Status message from EXEC STATUS @LPx command, appears on issuing console. The specified page of the current request is being printed.
CURRENT LOG FILE	Status message from EXEC LOGGING command; appears on issuing console; gives the pathname of the the current log file.
CURRENTLY NOT LOGGING	Status message from EXEC LOGGING command; appears on issuing console. EXEC is not logging now; Use CX LOGGING/START *pathname* ↵ to start if you wish.
DEVICE ALREADY IN USE	System error message from EXEC ENABLE command, appears on issuing console. EXEC could not enable this console — probably because the console is "owned" by another DG process, like DATAPREP, TPMS, or DG/SNA.
DEVICE ALREADY MOUNTED	Error message from EXEC MOUNTED command; appears on issuing console. You have already told EXEC that a tape is mounted on this unit.
DEVICE IS NOT STARTED	Error message from EXEC STOP command, appears on issuing console. The device specified has not been started.
DEVICE UNKNOWN	Error message from EXEC START or DISABLE command; appears on issuing console. Perhaps a typo caused this. Or see the UP.CLI macro for correct syntax.
DISABLE CANCELLED, CONSOLE ENABLED	Status message from EXEC ENABLE after DISABLE; appears on issuing console. The DISABLE is cancelled; the user console remains ENABLEd.

(continued)

Table 8-4. Operator Messages from EXEC

Message	Description and Action
DISABLED CONSOLE	Status message from EXEC DISABLED; appears on issuing console. The user console was not in use and has been disabled. If on, the console has a *CONSOLE DISABLED* message.
DRIVE ALREADY MOUNTED -- DISMOUNT FIRST	Error message from EXEC MOUNTED command; appears on system console. You must dismount the tape and type CX DISMOUNTED↵; then retry the MOUNTED command.
DRIVE IN USE - CANNOT DISMOUNT	Error message from EXEC DISMOUNTED command; appears on system console. The tape is mounted and I/O with it is occurring; so EXEC cannot obey. Perhaps you specified the wrong unit. If not, either wait or use SEND to ask the user to interrupt his/her tape I/O. If all else fails, have the user log off (or TERM the user), dismount the tape, and type CX DISMOUNTED↵.
DRIVE NOT MOUNTED -- CANNOT DISMOUNT	Error message from EXEC DISMOUNTED command; appears on system console. A MOUNTED or PREMOUNT request is not outstanding to this tape unit. Perhaps you specified the wrong unit; check with CX MOUNTSTATUS↵.
ENABLED CONSOLE	Status message from EXEC ENABLE command; appears on issuing console. The user console is enabled for user log on. If on, it shows a *TYPE NEW LINE TO START LOGGING ON* message.
ENABLING ALL CONSOLES	Status message from EXEC ENABLE/ALL command; appears on issuing console.
ERROR: text	An error message; appears on issuing console or batch output listing. *text* here is another message; check the message column of this table to find it. The text may be a system (not EXEC) error message with instructions; if so, follow them.
EXEC IS ALREADY RUNNING	Initialization error, from system; appears on system console after EXEC-creating PROCESS command. EXEC is already running. Do nothing.
EXEC LOGGING { *STARTED* / *STOPPED* }	Status message from EXEC LOGGING command; appears on issuing console. It confirms EXEC logging status and file pathname.
EXEC NOT AVAILABLE	Error message from system; appears on issuing console. You tried a Q-series command and EXEC isn't running. Bring EXEC UP↵ and try again.
EXEC'S PROCESS SUB-TREE ONLY	Error message from CLI MOUNT command; appears on system console. You issued a MOUNT command from the system console. Only consoles enabled by EXEC can issue MOUNT commands. Log on as OP on a user console and try again.
FILE ACCESS DENIED	Error message from system; appears on issuing console. You do not have access to this directory or file. If you need access, turn SUPERUSER on and try again.

(continued)

Table 8-4. Operator Messages from EXEC

Message	Description and Action
FILE DOES NOT EXIST:	Error message; appears on issuing console. This device was not identified to the VSGEN program. This can also be a system error from a CLI command.
FILE SPACE EXHAUSTED	Error message from system, appears on issuing console. Generally, this means that tape file space has been exhausted. But it can occur if you issue too many ENABLE or DISABLE commands too quickly; insert more CLI PAUSE commands between the EXEC commands.
FLUSHING CURRENT DECK	Error message from Stacker; appears on issuing console after an EXEC STACK command. The Stacker has found a bad $$JOB or $$PASSWORD card. See BAD $$JOB message for action.
FLUSHING CURRENT JOB AT USER REQUEST	Status message; appears on system console. The user has issued QCANCEL for this request; EXEC is flushing it. (*User* QCANCELs, unlike EXEC's CANCEL, can cancel an active request.)

FORMS ACCESS DENIED	Error message from CLI QPRINT/FORMS= command; appears on issuing console. The person who issued the command does not have R,E access to directory FORMS. With SUPERUSER on, give directory FORMS and all its files +,RE access:
	`DIR :UTIL;   ACL FORMS +,RE )`
	`DIR FORMS;   ACL + +,RE )`

FORMS DO NOT EXIST	Error message from CLI QPRINT/FORMS= command; appears on issuing console. Directory :UTIL:FORMS, or the specified file, does not exist. If directory doesn't exist, create it as described in Chapter 5. If the *file* doesn't exist, have the user specify an existing filename in FORMs.
FROM PID n :(EXEC) text	EXEC preamble, on system console. *text* is another message; check the message column of this table to find it.
FROM SYSTEM: text	System error message; try to find the text message in this table.
GROWING TO n PAGES	Status message on system console, displayed by EXEC as it enables consoles and gives itself more memory.
HARD ERROR	Error message from system; appears on system console. A hardware error prevents further access to the device. Don't retry; see "Hardware Errors" in Chapter 6.
ILLEGAL CHARACTER IN LOGICAL TAPE NAME	Error message from user CLI MOUNT or DISMOUNT command; appears on issuing console. The specified linkname has an illegal character. Have the user re-issue command with valid linkname.
ILLEGAL COOP MESSAGE	Error message from EXEC command; appears on issuing console. An illegal message was passed from one of EXEC's co-operative processes. Bring EXEC down and up again.

(continued)

Table 8-4. Operator Messages from EXEC

Message	Description and Action
ILLEGAL DECIMAL NUMBER	Error message from EXEC commmand; appears on issuing console. The number you specified was not legal; perhaps it contained a decimal point.
ILLEGAL DEFAULT FORMS	Error message from EXEC DEFAULTFORMS command; appears on issuing console. The default forms format given in the file is illegal. EXEC retains the standard forms. Fix the erroneous forms file in directory FORMS.
ILLEGAL DEVICE OR CONSOLENAME FORMAT	Error message from EXEC command; appears on issuing console. The command gave an illegal device or console filename; e.g., @C#N5. It can also occur if you omit the @, which specifies :PER; e.g., CON2 would cause it.
ILLEGAL START OPTION FOR QUEUE TYPE	Error message from EXEC START command; appears on issuing console. The specified queue doesn't allow the option given. Re-issue the command without the option.
ILLEGAL TIME FORMAT FOR STACKER	Error message from Stacker; appears on issuing console after an EXEC STACK command. The Stacker hit a $$JOB card with bad date or time format. For action, see BAD $$JOB.. message.
INCORRECT LABELED TAPE FILE ...	Error message from system; appears on issuing console, from access to a mounted labeled tape. The tape label, or volid sequence, is wrong. Perhaps you tried to start reading (LOAD) in a volume other than the first volume; if so, use the LOAD /SPECIFIC switch or start with the first volume. Or perhaps the volume doesn't belong to the fileset. If the message indicates that the label itself is bad, and you are writing to tape, reLABEL all volumes and try again.
INITIAL DIRECTORY MUST BE :PER	Initialization error; appears on system console after EXEC-creating PROCESS command. EXEC was started from a directory other than :PER. EXEC terminates. Re-issue the PROCESS command, with the /DIR=@ switch.
INSUFFICIENT MEMORY FOR LOGON	Error message from EXEC startup, on system console. EXEC cannot acquire enough memory to log users on or off.
INTERNAL CONSISTENCY ERROR IN *EXEC ....* ****HAVE EVERYONE LOGOFF AND THEN TERMINATE THIS EXEC*	Error message; appears on system console. Use the SEND command to warn users; DISABLE all consoles; then, when users are ready, type DOWN ↵ or TERM OP:EXEC ↵. Then bring EXEC up again. If you want to submit a Software Trouble Report (STR) to DG about this problem, use the /BREAKFILE switch on the TERMINATE command (e.g., TERM/BREAK=EXEC.DUMPFILE OP:EXEC). Then, submit a copy of the breakfile with the STR.
INVALID USERNAME-PASSWORD PAIR	Error message from Stacker; appears on issuing console after EXEC STACK command. The Stacker USERNAME and $$PASSWORD cards submitted were invalid. For action, see BAD $$JOB message. This can also occur interactively if you try to log on as a user and type an invalid username or password.

(continued)

Table 8-4. Operator Messages from EXEC

Message	Description and Action
JOB NOT QSUBMITTED	Status message; appears after EXEC STACK command on issuing console. The Stacker encountered invalid $$JOB and/or $$PASSWORD cards; it has rejected the job. For action, see BAD $$JOB message.
JOB QSUBMITTED	Status message; appears after EXEC STACK command on issuing console. The Stacker encountered valid $$JOB and $$PASSWORD cards; it has accepted and enqueued the job.
JOB RESTARTING	Status message; appears on system console when you bring up EXEC after abnormal termination. Batch and device requests restart automatically, with this message.
LIMITING ENABLED	Status message; appears after EXEC STATUS command, on issuing console. It means that the stream or device was limited via EXEC's LIMIT command.
@LMT ALREADY EXISTS	Error message from EXEC initialization; appears on system console. EXEC tried to create entry :PER:LMT, but the entry already existed. `DELETE :PER:@LMT )` and bring up EXEC again.
LOG ON/OFF IN PROGRESS	Status message after EXEC CONSOLESTATUS command; appears on issuing console. It means that a user is logging on or off with this console.
@LPBx COOPERATIVE TERMINATED	Status message after EXEC STOP; or error message after EXEC START; appears on issuing console. The XLPT co-operative process has terminated. If you want to have it running, see if the line printer is on line; if the printer is not ON LINE, put it ON LINE and start and continue the printer. See the UP.CLI macro for syntax.
@LPBx PHYSICAL UNIT OFFLINE	Error message from EXEC START command; appears on issuing console. The XLPT co-operative process terminates. Put the printer on line and start and continue it. See the UP.CLI macro for syntax.
@LPBx WILL PAUSE AT END OF CURRENT JOB	Status message after EXEC PAUSE command; appears on issuing console. The device will pause after finishing its current job.
MAGTAPE VOLUME ALREADY IN USE BY ANOTHER PROCESS	Error message from user MOUNT request; appears on issuing console. Another user is already using this tape volume. The user must wait until it is dismounted.
MOUNT REQUEST CANCELLED, NO RESPONSE POSSIBLE	Status message after user MOUNT command; appears on system console. The user typed a MOUNT command, but typed DISMOUNT or logged off before you mounted his tape. Do nothing.
MUST BE A FULL PATHNAME	Error message from a program that requested EXEC service via system call ?EXEC. The program must specify a full pathname from the root directory (:). This appears on the console that is running the program or the batch output listing.
NEXT VOLUME ...	Status message from user MOUNT; appears on system console. The user's MOUNT request needs the next tape volume; mount it on a free unit and type `CX MOUNTED @unitname )`

(continued)

Table 8-4. Operator Messages from EXEC

Message	Description and Action
NO MORE VOLIDS IN LIST SPECIFIED IN MOUNT COMMAND	Error message from user MOUNT sequence; appears on issuing console, or batch listing if batched. All file space in the list of volids is exhausted, yet material remains to be written to tape. The labeled tape file is incomplete. The user should resubmit the MOUNT request and specify more volids, or use the MOUNT/EXTEND switch, or both.
NO MOUNT REQUESTS	Status message from EXEC MOUNTSTATUS command; appears on issuing console. There are no requests in the mount queue.
NO OUTSTANDING MOUNT REQUEST	Error message after EXEC MOUNTED, REFUSED, or DISMOUNTED command; appears on system console. There are no requests of the pertinent type in the mount queue. Try CX MOUNTSTATUS).
NO SPOOLERS STARTED	Status message from EXEC SPOOLSTATUS command; appears on issuing console. No spooling device co-operatives (e.g., XLPT) have been started.
NO SUCH CO-OPERATIVE	Error message from EXEC command; appears on issuing console. Your specified co-operative process doesn't exist.
NO SUCH MAGTAPE UNIT	Error message from EXEC MOUNTED command; appears on issuing console. The unit doesn't exist.
NO SUCH QUEUE	Error message from EXEC command; appears on issuing console. There is no such queue. Check the queues with QDISPLAY).
NO SUCH QUEUE TYPE	Error message from EXEC CREATE command; appears on issuing console. The queue types are PRINT, PLOT, PUNCH, HAMQ, FTA, and SNA.
NON-UNIQUE $$ CARD	Error message from Stacker, from EXEC STACK command; appears on issuing console. The Stacker hit a $$JOB card with non-unique switch specification. For action, see BAD $$JOB message.
NOT A CONSOLE DEVICE TYPE	Error message from EXEC ENABLE command; appears on issuing console. The command specified a nonconsole device; correct it.
NOT ENOUGH ARGUMENTS FOR COMMAND	Error message from EXEC command; appears on issuing console. Retry the command with the correct number of arguments.
NOT FOUND	Error message from CX command; appears on issuing console. EXEC could not find the item or file specified. Check; try a corrected command.
NOT PAUSED AND IDLE	Error message from EXEC BINARY, CPL, FORMS, LPP or similar command; appears on issuing console. You must PAUSE the device and wait until it becomes IDLE (or in worst case STOP and FLUSH it), then retry the command.
NOT WAITING TO BE ALIGNED	Error message from EXEC ALIGN/CONTINUE command; appears on issuing console. The printer was not paused via EXEC'S ALIGN command.

(continued)

Table 8-4. Operator Messages from EXEC

Message	Description and Action
NOT YOUR UNIT	Error message from a user DISMOUNT command; appears on issuing console. The user typed DISMOUNT and specified a linkname that someone else is using.
$$PASSWORD CARD MISSING	Error message from Stacker after EXEC STACK command; appears on issuing console. For action, see BAD $$JOB message above.
$$PASSWORD NOT PRECEDED BY $$JOB	Error message from Stacker after EXEC STACK command; appears on issuing console. For action, see BAD $$JOB message above.
PHYSICAL UNIT OFFLINE	Error message from EXEC MOUNTED command; appears on issuing console. The unit that you specified is not on line. Put it on line and type the EXEC MOUNTED command again.
PREVIOUS VOLUME ...	Status message from EXEC, appears on system console as part of tape status message.
PROFILE NOT FOUND	Error message, from Q-series command, appears on issuing console. You have no user profile. There must be a profile for username OP if the operator is to issue Q-series commands from the system console. Create such a profile with PREDITOR.
QUEUE ALREADY EXISTS	Error message from EXEC CREATE command; appears on issuing console. The queue you tried to create already exists. You can OPEN, START, and CONTINUE its device. To reCREATE it, you must first DELETE it.
QUEUE IS ALREADY OPEN	Error message from EXEC OPEN command; appears on issuing console. The queue you tried to open is already open. You can START it on a device and CONTINUE it if you want.
QUEUE IS FULL	See THE QUEUE IS FULL.
QUEUE IS NOT EMPTY	Error message from EXEC DELETE command; appears on issuing console. The queue you tried to DELETE has requests in it. If you must DELETE it, STOP it; then PURGE and DELETE it.
QUEUE IS NOT OPEN	Error message from EXEC CLOSE command; appears on issuing console. The queue you tried to close is not open (but may still have requests in it; use QDISPLAY to check).
QUEUE IS OPEN	Error message from EXEC DELETE or PURGE command; appears on issuing console. The queue you tried to delete or purge is open. Before retrying, STOP it and PURGE it; PAUSE the associated device if needed; then CLOSE and DELETE it.
QUEUE IS STARTED	Error message from EXEC START command; appears on issuing console. The queue is already started.
QUEUE MAY NOT BE DELETED	Error message from EXEC DELETE command; appears on issuing console. You cannot delete a permanent queue, BATCH_INPUT, BATCH_LIST, or BATCH_OUTPUT.

(continued)

Table 8-4. Operator Messages from EXEC

Message	Description and Action
REQUEST ENTRY ALREADY EXISTS	Error message; appears on system console. It may indicate that another EXEC is running. Check with the ? macro; if two EXECs are running, get users off, TERM them both, and restart one. If only one EXEC is running, type DELETE :PER:EXEC+↲
REQUEST IS	User MOUNT message, appears on system console; and specifies EXPLICIT or IMPLICIT LABELED MOUNT, and/or WRITE RING OUT, or other information. Either find the (labeled or unlabeled) tape, mount it, issue the EXEC MOUNTED command; or refuse the request — perhaps using SEND to tell the user why you refused it.
REQUIRES A MAGTAPE UNIT NAME	Error message from EXEC MOUNTED command; appears on issuing console. Include the unitname, form @MTxn, in the MOUNTED command.
RESPOND: CONTROL @EXEC DISMOUNTED	Status message; appears on system console after a user types a DISMOUNT command or logs off with an outstanding MOUNT request. Type CX DISMOUNTED ↲.
RESPOND: CONTROL @EXEC MOUNTED @UNITNAME OR CONTROL @EXEC REFUSED	Mount message; appears on system console after a user issues a MOUNT command. It is preceded by other information. Either find the tape, mount it, and type CX MOUNTED @MTxn ↲; or type CX REFUSED ↲, perhaps sending the user an explanation.
SEARCHING FOR $$JOB	Status message after EXEC STACK command; appears on issuing console. EXEC is looking for the user $$JOB card entry. If $$JOB and $$PASSWORD are okay, the Stacker will accept the job and say JOB QSUBMITTED.
SOFT ERROR	System error message; appears on system console. A correctable media error occurred. If it recurs, see "Hardware Errors" in Chapter 6.
SPECIFIED FORMS FILE DOES NOT HAVE VFU SPECIFICATIONS	Error message from XLPT co-operative process; appears on printer. The forms file given by user with QPRINT/FORMS= does not have Vertical Format information. Have the user fix the file in :UTIL:FORMS with the FCU utility (or fix the file yourself); then have the QPRINT command re-issued.
STACKER REACHED END OF FILE	Status message from Stacker; appears on console that issued STACK command, after Stacker has completed the job(s) in the reader or on disk. It terminates. To stack another job, put it in the reader and issue the STACK command again.
SWITCH IS NOT UNIQUE	Error message from EXEC command; appears on issuing console. The characters you gave for the switch were not unique. Retry command with more switch characters.
SWITCH REQUIRES A VALUE	Error message from EXEC LOGGING command with /MAX= switch; appears on issuing console. Specify a maximum number of blocks or omit the switch.
SWITCH UNKNOWN	Error message from EXEC command; appears on issuing console. The switch is unknown to EXEC. Try XHELP.

(continued)

Table 8-4. Operator Messages from EXEC

Message	Description and Action
TAPE ALREADY MOUNTED	Error message from EXEC MOUNTED command; appears on issuing console. The tape is already mounted.
TAPE SET IN USE - CANNOT DISMOUNT	Error message from EXEC DISMOUNT command, appears on issuing console. You cannot dismount this tape(s) now; they are in use. Check with CX MOUNTSTATUS ↵.
THE QUEUE IS FULL	Error message from Q-series command; appears on system console, user console, or batch output file. The queue needed by your command is full; check with QDISPLAY. The maximum number of queue entries is 256. Have the user try again when there are fewer entries.
TOO { *FEW ARGUMENTS* / *MANY ARGUMENTS* / *MANY SWITCHES* }	Error message from EXEC command; appears on issuing console. You gave an illegal number of arguments or switches for the command. Retry command, using XHELP if needed.
UNABLE TO INITIALIZE THE QUEUE	Error message during EXEC startup; appears on system console. EXEC couldn't open the queue (file :QUEUE:Q). This probably means that QCMP or some other program has QUEUE:Q open. Check and terminate the other program.
UNIT DISMOUNT ...	Mount message; appears on system console after a user has typed DISMOUNT or logged off with an outstanding mount request. EXEC will say NO RESPONSE POSSIBLE or tell you what to do. In any case, you need to remove the tape from the unit.
UNIT MOUNT ...	MOUNT message; appears on system console, followed by user and MOUNT ID= information and perhaps REQUEST IS... information; followed by RESPOND.. instructions. Follow EXEC's advice and/or see REQUEST IS... message above.
UNKNOWN $$ CARD	Error message from Stacker; appears on console that issued EXEC STACK command. For action, see BAD $$JOB message above.
UNKNOWN STREAM NUMBER	Error message from EXEC command; appears on issuing console. Valid batch stream numbers are 1, 2, 3, or 4.
USER SPECIFIED FORM DOES NOT MATCH FORM IN PRINTER	Error message from XLPT co-operative process, appears on printer listing. A user tried to print a file that has form specs in the file's own UDA (not in a different forms file, specified via QPRINT/FORMS=). But the specs in the file's UDA (LPP, CPL, etc.) conflict with the current printer form specs. For the file to be printed, the conflict must be resolved. The file UDA can be changed via FCU (preferably by the user) to eliminate the conflict; or the printer forms file can be changed with EXEC's FORMS or DEFAULTFORMS command (if you know the name of a forms file that can handle the user file's UDA specs).
VALID ONLY FROM OPERATOR	Error message from EXEC command; appears on issuing console. You tried to issue a command to EXEC, but your username is not OP. EXEC accepts commands only from a process with the username of OP. If you are the operator, log on as the operator (or go to the system console) and retry the command.

(continued)

Table 8-4. Operator Messages from EXEC

Message	Description and Action
WAITING TO BE ALIGNED	Status message from EXEC STATUS comand, appears on issuing console. An EXEC ALIGN command is in effect; align the printer(s) if needed and type an ALIGN/CONTINUE command to it.
WARNING:text	An error message; appears on issuing console or batch output listing. *text* here is another message; check the message column of this table to find it. The text may be a system (not EXEC) error message with instructions; if so, follow them.
WRONG QUEUE TYPE FOR...command...	Error message from EXEC OPEN, START, or other queue-oriented command; appears on issuing console. The queue type, as set with the EXEC CREATE command, is wrong for the command just issued. For example, `CX START LPT @PLA ↲` could cause this. If the queue type is the problem, DELETE and reCREATE the queue.
WRONG VOLUME	Error message from EXEC MOUNTED command; appears on system console. The tape volume you just mounted is not the next one specified by the user's volid list. Remove it from the unit, find the correct one, mount it, and type `CX MOUNTED @unitname ↲`

(concluded)

User Messages from EXEC

The messages that EXEC gives *users* differ somewhat from the messages it gives to the operator process. But user messages are important because

- You, as the operator, are treated as a user whenever you issue a Q-series command: QBATCH, QPRINT, and so on. You *are* a user when you log onto a user console — even if your username is OP. So you will probably receive some of these messages.
- Other users will receive these messages and ask you — in your role as operator — what they mean.

Table 8-5 shows and explains the most common EXEC user messages, in alphabetical order. The word "you" applies to the person who received the message — either on a console or, for batch, on the batch output file. The table includes a few pertinent *system* (not EXEC) error messages.

Messages may appear either on a user's console, a batch output listing, or a printed file.

You may want to make copies of this table for users.

Table 8-5. EXEC User Messages

Message	Meaning/Action
ABORT: text	An abort error message; appears on issuing console or batch output listing. *text* here is another message; check the message column of this table to find it. This may be a system-generated (not EXEC generated) message.
ALREADY BEING PROCESSED	Error message; appears on issuing console. You tried to hold this job via a CLI QHOLD command, but the entry was already being processed. Use QCANCEL.
sysid BATCH OUTPUT FILE	Identification message; appears at the beginning of the batch output file, along with logon information, for each batch job.
BINARY MODE NOT ALLOWED	Error message from QPRINT/BINARY command; appears on printed file. The operator must enable binary mode with an EXEC BINARY command before anyone can QPRINT/BINARY.
BINARY MODE NOT ENABLED	Error message; appears in printed file. You included the /BINARY switch on your QPRINT command, but the system operator has not enabled binary mode on the pertinent printer. Have him/her do this with EXEC's BINARY command.
CANCELLED BY OPERATOR	Status message; appears on batch/printed output file. The system operator cancelled your request. If you want to know why, ask with the SEND command.
CANCELLED BY USER	Status message; appears on batch output/printed file. You cancelled your request via the CLI QCANCEL command.
CANNOT DELETE UNEXPIRED FILE ON LMT	System error message from write to labeled tape; appears on issuing console or batch output file if batched. The default retention period on files written to labeled tape is 90 days. Specify another tape volume or have a new label written to the tape with the LABEL program.
CANNOT RESTART, /NORESTART SPECIFIED	Status message; appears in batch output or printed file. The system (as always) tried to restart your batch request; or the operator attempted to restart your printing request, but you specified /NORESTART when you queued the job, so it will not be restarted.
CONSOLE DISABLED FROM LOGGING ON	Appears on a console screen/printer. The system operator has disabled this console for EXEC-supervised user log on. You cannot log on to it.
CONTACT YOUR SYSTEM MANAGER	Error message, usually means EXEC or the system found an error in your user profile. The profile may need to be recreated; see the system operator or manager.
DATA GENERAL AOS text	This is the beginning of the default system banner — appearing on batch output listings and consoles that are ready for user log on. To log on, press ↲ (NEW LINE), then enter your username and password. The banner (DATA GENERAL AOS) can be changed by someone in authority via the CLI command SYSID.
DELETE ACCESS DENIED	Error message from a Q-series command with /DELETE switch, appears on console or batch listing. You do not have delete (Write) privileges to the request source file. The command will be executed without the /DELETE.

(continues)

Table 8-5. EXEC User Messages

Message	Meaning/Action
DIRECTORY ACCESS DENIED	Error message from system; appears on console or batch or printer output file. You do not have the read/execute access privilege to this file. If you really need to read the file, ask the system operator to include your username,RE (or +,RE) in the file's ACL.
ENTER YOUR NEW PASSWORD:	Appears on console after you enter your password but press the ERASE PAGE key instead of NEW LINE. Type in your new password, 3 to 15 alphanumeric characters, For a valid password, the system will then say *new password in effect.*
ERROR: text	An error message; appears on issuing console or batch output listing. *text* here is another message; check the message column of this table to find it. This may be a system-generated (not EXEC generated) message.
ERROR PROCESSING PROFILE	EXEC couldn't process your user profile; see the system operator or manager. The profile may need to be recreated.
FILE ACCESS DENIED	Error message from system; see DIRECTORY ACCESS DENIED.
FLUSHED BY OPERATOR	Status message, appears on batch output or printer listing. The system operator has specifically flushed your request.
FORMS ACCESS DENIED	Error message from QPRINT/FORMS= command, appears on printed file. You do not have at least E access to directory :UTIL:FORMS and R access to the file. The system operator can correct this with the ACL command.
FORMS DO NOT EXIST	Error message from QPRINT/FORMS= command, appears on printed file. Directory :UTIL:FORMS does not exist, or the file does not exist. The system operator should check for the existence of :UTIL:FORMS and create it if necessary; if it exists, check the filename specified.
ILLEGAL CHARACTER IN PASSWORD PASSWORD NOT CHANGED	Error message from system; appears on console. You entered an illegal character in your new password. Log off, then on again, and enter a legal password: 3 to 15 filename characters.
ILLEGAL VFU CHANNEL AFTER VFU NEXT	Error message from QPRINT/FORMS= command, appears on printed file. A channel number was not between 1 and 12. The cause is probably an error in the text source file; if not, check the forms file with the FCU utility.
ILLEGAL VFU LINE SLEW AFTER VFU NEXT CHAR	Error message from QPRINT/FORMS= command, appears on printed file. Probable cause is a syntax error in the text source file or the FCU-created forms file. Check the source; then check the forms file via the FCU utility.
INCORRECT LABELED TAPE FILE.....	Error message from system, from access to a labeled tape volume; appears on console or batch output listing if batched. The tape label, or volid sequence, is wrong. Perhaps you tried to start reading (LOAD) in a volume other than the first volume; if so, use the LOAD/SPECIFIC switch or start with the first volume. Or, the volume may not belong to this fileset. If the message indicates that the label itself is bad, have the operator reLABEL the volume(s) (or do it yourself) and try again.

(continued)

Table 8-5. EXEC User Messages

Message	Meaning/Action
INDECIPHERABLE DUMP FORMAT	Error message from system; appears on console after CLI LOAD command. The cause may be 1) This is an unlabeled tape and you tried to treat it as labeled; e.g., LOAD/V TAPE:FILE ↲; 2) The file was not DUMPed to the tape (it may have been written with COPY or by a user program); 3) the file was DUMPed with a nonstandard buffersize (e.g., /BUFF=8192); if so, try to LOAD it with the original /BUFFERSIZE.
INVALID USERNAME-PASSWORD PAIR	Error message from system; appears on console. EXEC cannot find a user profile with this username and password. Try again.
LAST MESSAGE CHANGE date time	Status message; appears on console after log on. This describes the time at which file :UTIL:LOGON.MESSAGE (whose contents are displayed for each user at log on), was last updated. Usually, the system operator or manager updates this file; but in some systems users can do so.
LAST PREVIOUS LOGON date time	Status message; appears on console or batch output file. Specifies the last time you logged onto this system.
LIST FILE EMPTY, WILL NOT BE PRINTED	Status message; appears on batch output file. You have not specified output to the batch list file with the /L or /QLIST switches.
MAY NOT RUN BATCH JOBS	Error message from system; appears on console. Your user profile does not allow you to submit batch requests. If you need batch, see the system operator/manager.
NEW PASSWORD IN EFFECT	Status message; appears on console. Your new password is now effective, and the old one ineffective.
NO MORE VOLIDS IN LIST SPECIFIED IN MOUNT COMMAND	Error message from MOUNT sequence; appears on console or on batch output if batched. File space in your list of volids is exhausted, yet material remains to be written to tape. The labeled tape file is incomplete. You may want to write it again, specifying more volumes with the MOUNT/VOLID= switch, or telling EXEC to extend the list as needed with the MOUNT/EXTEND switch, or use both switches. One 2400-ft. tape, DUMPed with high density, 1600 bpi, /BUFFER=8192, holds about 39 megabytes.
NO SUCH QUEUE	Error message from system; appears on console or batch/printed output file. The specified queue doesn't exist. Type QDISPLAY ↲ to see the names of existing queues.
NON VFU CTRL CHARACTER AFTER VFU NEXT	Error message from QPRINT/FORMS= command; appears on printed file. There is an invalid character after ↑R. Check the text source file; if it's okay, then check the forms file with the FCU utility.
OPERATOR NOT AVAILABLE	Error message from MOUNT command; appears on console or batch output file if MOUNT command was batched. The EXEC command OPERATOR ON has not been issued by the operator process. MOUNT requests cannot be honored until someone types the EXEC command OPERATOR ON on the system console.

(continued)

Table 8-5. EXEC User Messages

Message	Meaning/Action
PAGE LIMIT EXCEEDED	Error message from QPRINT command; appears on printed file. The system operator enabled page limiting, and the number of pages EXEC estimated for your request exceeds the limit. Append the /PAGES= switch to your QPRINT command (estimating the number of pages); or try another queue.
PASSWORD:	System query message during log on. Type your password followed by ↓ (NEW LINE).
? PASSWORD MUST HAVE 3 TO 15 CHARACTERS *--PASSWORD NOT CHANGED --*	Error message from system; appears on console while you are changing password. Log off and try again, giving a password of 3 to 15 alphanumeric characters.
PHYSICAL UNIT FAILURE	Error message from system; appears on console during I/O. The unit (disk or tape) on which the file resided has returned a HARD ERROR. Do not type anything else; tell the system operator immediately. HARD ERRORS are described in Chapter 6, "Abnormal Shutdown".
PRIORITY TOO HIGH FOR YOU	Error message from Q-series command; appears on console or batch/printed output file. Your highest priority (in your user profile) is too low for the queue; or, the priority you gave with the /QPRIORITY switch is higher (closer to 0) than the profile allows. Consult the system operator.
PROCESS nn TERMINATED [by] *ELAPSED TIME:* *USER 'user' LOGGED OFF*	Status message from system; appears on console or batch output file. The user process that connected your console or batch job has terminated normally, or by operator command, or by the system, or by a CTRL-C CTRL-B sequence.
PROFILE NOT FOUND	Error message after Q-series command; appears on batch output file. You issued a Q-series command (perhaps with an /AFTER switch); but EXEC could not find a profile with your username when it tried to run the request. Your profile may have been changed or deleted; see the system operator.
REQUEST REFUSED BY SYSTEM OPERATOR	Error message from MOUNT command; appears on console or in batch output file. The operator specifically REFUSED your request. There may also be an explanation of why the operator did this.
RESTARTED BY OPERATOR	Status message; appears on printed file. The operator specifically RESTARTed this request.
SKIP TO VFU CHANNEL N GIVEN CHANNEL N NOT PUNCHED	Error message from QPRINT/FORMS= command; appears on printed file. Syntax error in text source file or forms file. Check the source file; if it's okay, check the forms file with the FCU utility.
TAB SENT BEYOND LAST TAB STOP	Error message from QPRINT/FORMS= command; appears on printed file. See previous message for action.
THE QUEUE IS FULL	Error message from Q-series command; appears on console or batch output file. The queue needed by your command was full; check with QDISPLAY; try again when there are fewer entries. The maximum number of entries in a queue is 256.

(continued)

Table 8-5. EXEC User Messages

Message	Meaning/Action
TOO MANY ATTEMPTS, CONSOLE LOCKING FOR n SECONDS	Error message during log on attempt; appears on console. Wait *n* seconds and try again.
TOO MANY ATTEMPTS, DISCONNECTING	Error message during log on attempt via remote console; appears on console. Hang up, redial, and try again.
TOO SLOW - DISCONNECTING	Error message during log on attempt via remote console; appears on remote console. Hang up, redial, and try again.
TOO SLOW - INPUT TIMED OUT	Error message during log on attempt; appears on console. If you still want to log on, start over.
UNABLE TO CREATE YOUR PROCESS	Error message during log on; appears on console or batch output file. EXEC couldn't create a user process for you. Your profile may need to be recreated. See the system operator or manager.
USERNAME:	Query message from system; appears during logon. Type your username and ↓ (NEW LINE).
USER NOT PRIVILEGED TO CHANGE PASSWORD	Error message; appears on console when you try to change your password. Your user profile does not allow you to change your password.
USER SPECIFIED FORM DOES NOT MATCH FORM IN PRINTER	From XLPT process. See this message in Table 8-4, preceding.
USER SPECIFIED @OUTPUT ERROR	Error from Q-series command; appears on batch output file. With the /QOUTPUT= switch, you specified an illegal @OUTPUT file. Try again with another @OUTPUT filename.
VALID ONLY FROM OPERATOR	Error message from CONTROL @EXEC (CX) command; appears on console. Your username is not OP and you issued a CONTROL command. Only processes with username OP can tell EXEC what to do.
WARNING: text	An error message; appears on issuing console or batch/printed output file. *text* is another message; check the message column of this table to find it. This may be a system-generated (not EXEC generated) message.
WAS NOT MOUNTED	Error message from DISMOUNT command; appears on console or batch output file if batched. The tape linkname you specified was not mounted. Perhaps you specified the wrong linkname or this is your second DISMOUNT command.
YOU MAY NOT LOG ON FROM A CONSOLE	Error message during logon attempt; appears on console. Your user profile does not allow you to use a console. You are restricted to batch.

(concluded)

What Next?

This chapter described EXEC — but really much more, because EXEC includes interactive user log on/log off, batch and spooling, queues, and labeled tapes and user mount requests. Other products, like XODIAC, entered the picture because they use EXEC queues. A lot of device-oriented information is also involved: consoles, line printers, and mag tape units. So if you understand even a part of EXEC, and know where to look for more, you've done a lot. Parts of EXEC appear complex because there are two parties involved: users (people logged on to user consoles) and the system operator. You can integrate and clarify what you're read by logging on as user, submitting a batch request or two, then a mount request or two, and watching what happens on the system console.

EXEC is *the* primary multiuser tool. But there are many others: the CLI, display programs, SYSLOG, and backup procedures. The next chapter describes these.

End of Chapter

Chapter 9
Other Runtime Tools: CLI, Lock, PED, Display, Logging, and Backup

Read this chapter:

- when you want to use one or more AOS tools to help operate the system.

There are many tools to help you run your system. They include the ESD, FIXUP, PREDITOR, and EXEC — all explained in previous chapters. The tools described in *this* chapter don't run by themselves — like EXEC — nor are they recovery programs like ESD and FIXUP.

Instead, these tools are for active use during the course of a normal working day. This chapter describes them in the following major sections:

- Using the OP CLI
- LOCK_CLI
- Process Environment Display (PED)
- DISPLAY and File Compare Utilities
- System Logging with SYSLOG and REPORT
- File Backup: DUMP and PCOPY
- What Next?

Other operator-oriented programs, that you may want to use *occasionally*, are the Disk Formatter, Installer, and diagnostics, described in later chapters.

Using the OP CLI

The CLI — AOS' command language — is a very powerful tool, with over 100 commands and a flexible macro facility that allows you to create your own commands.

When AOS comes up, it runs a CLI process on the system console. This CLI has process ID (PID) 2 and it is the father of all lower processes. It has a number of special privileges, including Superuser and Superprocess. This CLI is called the *root* or *master* CLI.

The UP macro issues a number of commands to the master CLI, including: an ACL command that gives the operator control of tape units, a PROCESS command that starts up EXEC, and finally an EXECUTE command that starts a son CLI under the original process. This son CLI is the one that usually runs on the system console.

The son CLI has nearly all the privileges of the father. It has its father's username, OP, which allows it to issue commands to EXEC. And it has the powers that the system operator may need to run the system.

There are several things that the son CLI cannot do. It cannot start or stop the system log; it cannot run the DOWN macro; and, it must have Superprocess on to terminate any user process.

To get back to the father, all a person needs to do is type BYE) on the system console.

Table 9-1 summarizes some process-oriented CLI commands and macros that can be issued by the master CLI, its son, or any user CLI that has the appropriate privileges. It describes the commands briefly, in the context of system operation. For full syntax and all switches, of any CLI command, use HELP↵ or see the CLI manual. For a *session* with the CLI, try *Learning to Use Your AOS System.*

You can abbreviate any CLI command to the shortest string of characters that identifies the command.

Table 9-1. Operator-oriented CLI Commands and Macros

Command or Macro	Description	Example
?.CLI	A macro that describes all processes on the system. Creating it is shown in Chapter 5.	) ? ↵
ACL *[template]*	Displays or sets a file access control list. ACL is the primary access control tool, explained below.	) ACL/V :UTIL:FORMS ↵) ACL/V + OP,OWARE ↵
BROADCAST.CLI	A macro that sends a message to all user consoles. Creating it is shown in Chapter 5.	) BROADCAST Logoff! ↵
BYE	Terminates the current CLI or SED text editor process. The process' father (process from which it was executed) gets control. If there is no father, as with the master CLI, BYE starts a system shutdown. Typed to a user CLI process, BYE logs the user off.	) BYE ↵
CHARACTERISTICS *[char]* *[/DEF [char]]* *[/RESET]*	Displays, sets, or resets the hardware characteristics of a console, overriding the AOSGEN setting if desired. Use it when you want to change the system console's (or any user console's) behavior.	) CHAR/PM ↵ . . .) CHAR/RESET↵) CHAR/D ↵) CHAR/D/605X ↵
CONTROL @process command	Sends an IPC message of command to process in :PER. Use it to control EXEC and other DG products. Also see the CX.CLI macro in this table.	) CONTROL @EXEC ALIGN ↵

(continues)

Table 9-1. Operator-oriented CLI Commands and Macros

Command or Macro	Description	Example
COPY dest-file source-file	Copies a source-file to a destination file (dest-file). COPY is useful when you want to copy a file literally, without the header information that DUMP writes.	) COPY @MTA0 TBOOT ⤶) COPY @NULL @MTA0 ⤶
CREATE *[/DIR]* path *[/LINK link dest]* *[/I path]*	Creates the file named in *path* (pathname). Use /DIR to create a directory file. Use /LINK to create a link entry to a destination file. You may use CREATE pretty often. To insert text in the new file, use CREATE/I.	) CREATE/DIR :UTIL:F77 ⤶) CRE/I FOO ⤶)) TYPE Hi ⤶))) ⤶)
CX.CLI	A macro that makes it easy to issue EXEC commands. Creating it is described in Chapter 5.	) CX ALIGN ⤶
DELETE template	Deletes the file(s) given in template. You will often delete files in the course of system housekeeping.	) DELETE/V +.BRK ⤶) DEL/V DIR:XX+ ⤶
DIRECTORY *[dir]*	Displays the working directory name or sets it to *dir*. The easiest way to work within a directory is to DIR into it.	) DIR ⤶) DIR :UTIL ⤶
DOWN.CLI	This is a DG-supplied macro that brings down EXEC. Tailoring it is described in Chapter 5.	) DOWN ⤶
DUMP file *[template]*	Copies files to tape or disk. DUMP is most useful for file backup (archiving), described later in this chapter.	) DUMP/V TAPE:FOO + ⤶
FILESTATUS *[template]*	Describes files in any directory. It is one of the most useful commands.	) FILES/AS ⤶ *) F/AS/S :+ ⤶
HELP *[command]*	Gives help on CLI topics or any command.	) HELP ⤶) HE/V ACL ⤶
LOAD file *[template]*	Loads dumped files; precise counterpart of DUMP.	) LOAD/V TAPE:FOO + ⤶

(continued)

Table 9-1. Operator-oriented CLI Commands and Macros

Command or Macro	Description	Example
MOVE dir *[template]*	Copies file(s) named in template to another directory. You'll use MOVE often for housekeeping and maintenance.	) MOVE/V/R : ERMES ↵
OFF.CLI ON.CLI	Macros that turn Super modes OFF and ON; shown in Chapter 5. Typing ON or OFF is easier than typing the Super command itself.	) ON ↵ *) OFF ↵) ON/P ↵ +)
POP	Restores the previous CLI environment. After PUSH, it can save a lot of effort. It's explained below.	) PUSH ↵) CHAR/PM ↵) TY XX ↵) POP ↵
PROCESS program	Creates a new process — as XEQ does — but is more versatile. You may want to write PROCESS commands for your applications into the UP macro.	) PROCESS/IOC=@CON2 CLI ↵
PUSH	Descends to a new CLI environment. To ascend, use POP.	See the POP example.
QBATCH command	Posts a batch job; you may use batch fairly often for prolonged operations that can run by themselves. Also, see the BATCH and CHEK macros in Chapter 5.	) QBATCH/AFT=17 DDUMP ↵
QDISPLAY	Describes all queues and their status; this is handy in a multiuser environment.	) QD/V ↵
QPRINT template	Posts a printing job; you'll use this when you want line printer hard copy.	) QPRINT MYFILE ↵) QPR/C=10 C:ZFILE ↵
RENAME old new	Renames a file. Use this to preserve a backup file when you know that the program you're about to run will delete the original.	*) REN ERMES ERMES.OLD ↵ *) NEWERMES ↵

(continued)

Table 9-1. Operator-oriented CLI Commands and Macros

Command or Macro	Description	Example
RUNTIME *[pid]*	Describes a process' statistics; use it when you think a process may be malfunctioning. If it's using all the CPU time, you *know* it's malfunctioning.	) RUN ↵) RUN 1 ↵
SEARCH *[dir] [,...]*	Displays or sets your search list. This is a list of directories the CLI scans when it can't find a file in the working directory. The UP macro sets your search list, but you may want to change it for some system operations.	) SEARCH ↵) SEA [!SEA],:ZDIR ↵
SED.CLI	A macro that executes the SED text editor; described in Chapter 5. It's easier than typing X SED filename.	) SED MYFILE ↵
SEND pid message	Sends a message to a process ID (pid). You'll use this often to communicate with users.	) SEND 20 What?? ↵
SPACE *[cpd]* *[cpd n]*	Tells you how much disk space is used, and how much remains free, in a control point directory (cpd). It's often useful to know this. With a *cpd n*, sets a different maximum size of *n* disk blocks.	) SPACE : ↵) SPA :UDD:OP ↵) SPA MYDIR 1000↵
SUPERPROCESS *[ON][OFF]*	Turns Superprocess, the power to control any process, on or off. You need this to terminate a brother or father process; it's detailed below.	) SUPERPROC ON ↵ +)
SUPERUSER *[ON][OFF]*	Turns Superuser, the ability to access and write any file on the system, on or off. You may need this to create, examine, and delete files outside your OP directory.	) SUPERU ON ↵ *)

(continued)

Table 9-1. Operator-oriented CLI Commands and Macros

Command or Macro	Description	Example
SYSTAPE.CLI	A DG-supplied CLI macro that creates a tailored system tape. Use it for tailored system backup.	*) SYSTAPE @MTB0 SYS.SY ⤶
TERMINATE process	Terminates a subordinate process. This is used in the DOWN macro; you will use it yourself when you need to terminate a process. With Superprocess on, this command can terminate any process.	) TERMINATE 34 ⤶ +) TERM 4 ⤶
TYPE template	Types an ASCII file on the console. Use it whenever you want to read a file or tape label.	) TYPE MYFILE ⤶) TY :UTIL:ERMES.SR ⤶) TY @MTA0 ⤶
UP.CLI	A DG-supplied macro that brings up EXEC and the multiuser environment.	) UP ⤶
WHO *[pid]*	Displays the username associated with PID *pid.* Use it when you want to know who is in the PID.	) WHO ⤶) WH 30 ⤶
WRITE arguments	Displays arguments on the console or listing file. WRITE is most useful in CLI macros.	) WRITE Hi ⤶) WRITE [!OCT 518] ⤶) WRITE/L=FILE [!DATE] ⤶
XEQ program	Executes a program — similar to PROCESS, but easier and less versatile.	) X DISPLAY ZZ.PR ⤶) X PED ⤶ *) X AOSGEN ⤶

(concluded)

File Access Control with ACL

In a multiuser system, users need access to, and privacy for, their own files. System and user files need protection from accidental (or malicious) deletion. AOS file *access control lists* (ACLs) ensure all this.

Each directory, including the root (:), and each file within it, has an ACL. The ACL can give any or all users (or no users) any or all access privileges. The privileges are O, W, A, R, E and they have the following meanings.

Letter	Privilege	Allows
O	Owner	With Owner access to a directory, a user can change its ACL — giving him- or herself other privileges at will. Owner access to a directory also allows a user to change the ACLs of all files in it.
W	Write	With Write access to a directory, a user can create and delete files, and change file ACLs. With Write access to a file, a user can modify the file's contents.
A	Append	With Append access to a directory, a user can add files to it. Append does not apply to files.
R	Read	With Read access to a file, a user can read the file (e.g., with the TYPE command) providing he or she has at least E access to the directory.
E	Execute	E applies primarily to directories and program files. An ACL of +,E (all users, E access) is often given to LDUs. E access to a directory allows a user to execute programs within the directory, if he or she has at least E access to the program file(s). E does not allow a user to read filenames in a directory nor read the text of a file. A user with Execute access to a directory can DIR into the directory and use the directory name in a pathname. In addition, a user with R can read the filenames in the directory (e.g., with the FILESTATUS command). R and E access privileges are often given together.

PREDITOR creates each user directory (in :UDD) with an ACL of username,OWARE. Each file the user creates within this directory gets the same ACL (username,OWARE) by default. The user profile files, in :UPD, have null ACLs, which means that only Superusers, like EXEC, can read them.

Any user can change the ACLs of his own files with the ACL command, or change the default ACLs with the DEFACL command. If you want to run a *really* secure system, in which no one has Superuser privileges, you can make your communication easier by putting a

```
DEFACL  [!USERNAME],OWARE  +,RE
```

command in each user's logon macro. This allows other users to read and execute (but not modify or delete) this user's files. Users can change the DEFACL command at will.

If you desire access to all files in a directory and its subordinates, without turning Superuser on each time, get into the directory and type

```
*) ACL/V  #  [!DEFACL]  OP,OWARE ↵
```

With Superuser on, you can bypass all ACLs. But so can any user, if he or she can turn Superuser on.

The ACL mechanism works automatically; and you don't need to play with ACLs often. But you do need to understand them.

In a multiple-user ACL, give the least privileged group first. For example, the ACL

```
+,RE  SALLY,OWARE
```

gives all users (which includes Sally) read and execute access — and *only* read and execute access. The O, W, and A privileges for Sally are ignored. With this ACL, no one but a Superuser can write to the file or change the ACL. But transposing the user groups:

```
SALLY,OWARE  +,RE
```

gives Sally all privileges — which is what people want from ACLs. When you use templates in multiuser ACLs, type the most specific username or template first, regardless of the specific privileges.

Filename Templates

A filename template includes a special character that specifies a set of filenames. The most common template characters are as follows.

Character	What it Means
*	Match any single character except a period.
−	Match any series of characters not containing a period.
+	Match any series of characters.
\	*Omit* a series of characters.
#	Search the specified directory *and all inferior directories.* Without this template, the search is restricted to the working (or specified) directory.

For example

This command	Searches for filenames of
) FILES/AS ****** ↲	six characters without a period.
) FILES/AS **.CLI ↲	two characters without a period, ending in .CLI
) FILES/AS − ↲	any characters without a period.
) FILES/AS −.CLI ↲	any characters without a period, ending in .CLI
) FILES/AS + ↲	any characters.
) FILES/AS +.CLI ↲	any characters ending in .CLI
) FILES/AS :#:+ ↲	any characters *in all directories in the system, starting at the root.*
) FILES/AS :#:+.SY ↲	any characters ending in .SY in all directories on the system, starting at the root.

Templates work with most CLI commands and are extremely useful. For file searches — especially prolonged ones with # — you may want to apply the /L=@LPT switch, so the listing will show you the pathnames of the filenames found.

Filename Suffixes and Their Meanings

Certain filename suffixes (or extensions) indicate the kind of material in the file. These suffixes are useful because they tell you at a glance what's in a file. You may want to delete files with certain suffixes periodically to conserve disk space. The most common filename suffixes are as follows.

Suffix	Meaning	Example Filename
.BRK	This is a break file created by a program on abnormal termination. These are useful in problem diagnosis, but generally you can delete them.	?013.16_34_22.BRK
.CLI	This is a CLI macro. It's in ASCII and you can type it.	UP.CLI
.ED	This is a SED edit file, created by SED for its own use during an edit session. Such files can be deleted if you need disk space.	UP.CLI.ED
.JOB	This is a batch input file, which the system deletes after the job completes. Such files remain in users' working directories only if jobs don't finish normally. You can type them.	?028.CLI.001.JOB
.LB	This is a library file: a group of .OB files from which Link takes material it needs to build .PR files.	URT.LB
.LPT	This is a temporary print queue file, which the system should delete, as with .JOB files. These files are created in directory :QUEUE.	JACK_0001.LPT
.OB	This is an object file, produced by compilation or assembly of a source file. Link uses .OBs to build .PR files.	SEDERMES.OB
.OL	This is an overlay file, used by the .PR file of some utility programs.	EXEC.OL
.PR	This is a program file, executable under AOS.	CLI.PR
.SR	This is an assembly language source file, in ASCII. It can be a basis for an executable program, or assign parameters. You can type such files.	PARU.SR
.TM .TMP	This is a temporary file, created by a utility program for its own use. The utility usually deletes such files after normal termination. Aside from the AOSGEN .TMP files, you will want to delete these.	?006.MYFILE.TM ?018.SPEED.A.TMP

Other suffixes are used for other DG products. For example, .F77 denotes a FORTRAN 77 source file, .PL1 a PL/I source file, and so on.

Screen Editing

The CLI — like the SED text editor — offers screen editing via console control characters. This can be a big timesaver on CRT consoles. It is detailed in *Learning to Use Your AOS System.*

Pushing and Popping

There are multiple levels of CLI available, each with its own environment. Normally, the CLI runs on level 0, the highest level.

But you can PUSH a level, set up a new CLI environment with new characteristics, directory, searchlist, superuser status, and others (described in the CLI manual). Then, when you're done with the new environment, simply POP to the higher one. The LEVEL command tells you the current level. PUSH and POP can help eliminate confusion and save time. For example:

) DIR ↵	
:UTIL	The working directory is :UTIL.
) PUSH ↵	Push a level.
) DIR :OBS:ZZZ ↵	Change directory and other things.
) SUPERU ON; SEARCH ZZ:XX ↵	
*) MOV/V :UDD:OP FILEX ↵	
FILEX	
*) POP ↵	Pop to restore everything.
) DIR ↵	
:UTIL	

(The semicolon allows you to stack CLI commands.)

Super Powers

Superuser, if on, allows a user to read, write, modify, delete, or change the ACL of any file on the system. Superusers can bypass all file access controls at will. This privilege can imperil all files. The CLI prompt is *) when Superuser is on.

Superprocess, if on, allows a process to block any other process, become resident, or terminate any other process, including the master CLI — which would shut down AOS. The CLI prompt is +) when Superprocess is on. It is #) when both Super powers are on.

The Superuser and Superprocess privileges are options for each user profile. Generally, users should not have these privileges. In some systems, even the *operator* doesn't have them — described in LOCK_CLI, next.

LOCK_CLI

LOCK_CLI is a lockable CLI, that requires a password to unlock or terminate. While locked, it executes only innocuous CLI commands. It is designed to safeguard the system console from unauthorized people. LOCK_CLI has two special commands, LOCK and UNLOCK, and comes up in the locked state. When locked, LOCK_CLI will ignore the following CLI commands.

BLOCK	DELETE	MOVE	RELEASE
BYE	DUMP	PROCESS	RENAME
CHAIN	ENQUEUE	PROMPT	SUPERPROCESS
CONNECT	EXECUTE	QBATCH	SUPERUSER
COPY	INITIALIZE	QFTA	TERMINATE
DEBUG	LOAD	QPLOT	XEQ
		QSUBMIT	

This means that you cannot execute programs (including the LABEL utility, which labels tapes) from a locked CLI. The LOCK_CLI program also will ignore any of the process termination sequences: CTRL-C CTRL-B, CTRL-C CTRL-E, CTRL-D CTRL-D.

While locked, LOCK_CLI allows printing (via QPRINT) of files in the root directory (:) only. And, it ignores the /L switch.

You can unlock LOCK_CLI by typing UNLOCK↵ and password↵; and you can lock it again via LOCK↵.

For example, assume the password XYZZY:

) WHO ↵	Check the process...
PID 20 : OP OP :LOCK_CLI.PR	It's LOCK_CLI.
) SUPERU ON ↵	Try for Superuser.
)	No error, but no Superuser prompt.
) F/AS :+ ↵	Try to check files in the root.
WARNING: FILE ACCESS DENIED	Not allowed.
) BYE ↵	Try to terminate LOCK_CLI.
) WHO ↵	Check again...
PID: 20 OP OP :LOCK_CLI.PR	Still LOCK_CLI.
) UNLOCK ↵	Start to unlock.
XYZZY ↵	Type password (doesn't echo).
) SUPERU ON ↵	See if it's unlocked...
*)	Yes — it now has full CLI powers.
LOCK ↵	Lock it again.
)	LOCK ↵ also turns off Super privileges.

You can type the password in either UPPER- or lowercase. LOCK_CLI is shipped with a password, but this should be changed according to your application. Changing LOCK_CLI's password is described in Chapter 14. Generally, users should not know the password to LOCK_CLI.

To execute LOCK_CLI, simply type X :LOCK_CLI↵. To run it as a matter of course, edit the UP.CLI macro. In this macro, change the line that says EXECUTE CLI to EXECUTE LOCK_CLI. Then insert SUPERUSER ON before the EXECUTE line and insert SUPERUSER OFF after. The three lines should look like this:

```
SUPERUSER   ON
EXECUTE   LOCK_CLI
SUPERUSER   OFF
```

Now LOCK_CLI will run on the system console whenever you come UP.

LOCK_CLI retains the username OP, so it can issue commands to EXEC while locked.

Process Environment Display (PED)

The Process Environment Display (PED) program displays a status report of the processes running on your system. Figure 9-1 shows a sample PED status report.

```
 ID  USER  PROCESS  PROGRAM  WELAPSED  CPU-TIME     WIO  PG-SEC  BS   FP  SH US
  1  PMGR  PMGR     PMGR     12:01:40  0:02:20.191    0  3215
  9  OP    FTA      FTA      12:00:43  0:00:07.259  492   180    B     5  16   7
 15  BARRY 015      SED       0:00:43  0:00:38.561  212  1347    B    18  21  11
 17  MARTY CON22    CLI       0:50:41  0:00:01.782   39    42    B     3  19   2
 19  LEVIT 019      UVTA      0:33:46  0:00:00.547   36     8    B    22   8   5
PED  REV 07.00  9:26:48  SPACE: MAX  370889  CUR  100721  REM  270168
```

Figure 9-1. Sample PED Status Report

To run PED on a model 6012 video display console, set the PAGE switch. You can invoke PED interactively or as a batch job, using the following format.

X PED *[delay][argument[!]] ...*

where:

delay is the number of seconds less than or equal to 65,535 between status report updates (The default is 10 seconds.) (You cannot specify *delay* in batch mode.)

argument is any of the arguments listed in Table 9-2. (The default is to display the first 12 arguments in Table 9-2 for the first 22 processes.)

! is a column separator between arguments

On non-DG consoles, on hard-copy consoles, and in batch mode, PED produces a single status report. If you invoke PED interactively from a video-display console, however, PED continually displays the status report. Also, on video-display consoles, PED periodically updates the status report according to the *delay* that you specify. To restart the delay cycle, type CTRL-C CTRL-A.

To terminate PED, type CTRL-C CTRL-B.

Table 9-2 lists the PED arguments. If you invoke PED with arguments, the status report displays only the requested information. Your console's screen size limits the status report. Therefore, if you specify only a few arguments, PED can display information for a maximum of 64 processes. If you specify too many arguments, PED displays the following message:

REQUESTED DISPLAY TOO WIDE

You can use ! or space as a column separator between arguments. PED returns the octal value of each argument that begins with ?. (Generally, octal values are 6 digit numbers.) PED does not return leading zeros in decimal numbers.

Table 9-2. PED Arguments

Argument	Meaning
ID	Process identification (PID) number (This is a default argument.)
USER	User name (first 5 characters) (This is a default argument.)
PROCESS	Simple process name (first 8 characters (This is a default argument.)
PROGRAM	Program name (This is a default argument.)
WEL	Connect time elapsed (wide)(0 < hours < 1000) (This is a default argument.)
CPU-TIME	CPU time used (This is a default argument.)
WIO	Number of 512-byte blocks transferred through the data channel (wide) (This is a default argument.)
PG-SEC	Memory page usage per second (This is a default argument.)
BS	B means blocked; S means swapped out to disk (This is a default argument.)
FP	PID number of the parent process (This is a default argument.)
SH	Number of shared pages of main memory used (This is a default argument.)
US	Number of unshared pages of main memory used (This is a default argument.)
SONS	PID numbers of sons (PED lists up to 8 sons.)
ELAPSED	Connect time elapsed
I/O	Number of 512-byte blocks transferred through the data channel
PAGE-MSEC	Memory page usage per millisecond
!	Column separator
?PSBK	Number of unshared process pages
?PSEX	Time slice exponent
?PSFL	Process flag word
?PSF2	Process flag word 2
?PSF3	Process flag word 3
?PSF4	Process flag word 4
?PSMB	Number of unshared system pages
?PSMX	Maximum number of pages
?PSNR	Number of tasks suspended on ?IREC system call
?PSPR	Process priority
?PSPS	Number of shared process pages
?PSPV	Process privilege bits
?PSQF	Priority enqueue factor
?PSSF	Starting page of shared process area
?PSSL	Number of sub-slices in time slice
?PSST	Process status word
?PSSW	Number of shared pages loaded on last swap
?PSYS	Number of shared system pages
?PSYT	Starting page of system shared area

DISPLAY and File Compare Utilities

The DISPLAY program can read any binary, ASCII or EBCDIC file — on tape or disk — and display it or write it to a listing file. DISPLAY is useful when you want to read a tape file without loading it; or when you want to see what an EBCDIC file looks like in ASCII; or when you want to see everything (including nonprinting characters) that a program wrote to a disk file. For example

```
) XEQ   DISPLAY   @MTA0:0 ↲
.
.
.
) X   DISPLAY   MYPROG.PR ↲
.
.
.
*) X   DISPLAY   :SYSLOG ↲
.
.
.
) X   DISPLAY/L=REAL_OUTPUT_FILE   OUTPUT_FILE ↲
```

DISPLAY is a help topic. Anyone can get help on it by typing HELP *DISPLAY ↲.

FILCOM and SCOM are file-compare utilities. FILCOM compares binary files; SCOM compares ASCII text files. These programs can be helpful when you want to see if two files differ, or how they differ. For example, if you have two revisions of a program and want to see if they differ, you'd use SCOM for the source file or FILCOM for the program file. For example

```
) DIR   NEW_PROGS ↲
) XEQ   FILCOM   MPROG_1.PR   MPROG_2.PR ↲
.
.  (listing to console)
.
)
```

The CLI manual gives a little more information on FILCOM and SCOM.

System Logging with SYSLOG and REPORT

SYSLOG, the system log function, records information for later playback with the REPORT program.

While on, the system log file automatically records

- user account-related information, as gathered by EXEC and AOS;
- XODIAC network information, as gathered by networking software and AOS;
- software, soft, and hard errors, as gathered by AOS.

People and programs can write additional information to the system log file in two ways:

- with the CLI command LOGEVENT. Any Superuser on the system can write messages into SYSLOG with the LOGEVENT command;
- with the assembly language system call ?LOGEV, described in the *Advanced Operating System (AOS) Programmer's Manual.*

The System Log File (SYSLOG)

The current system log file is always called SYSLOG and is located in the root directory. You start, stop, and rename the log file with the CLI command SYSLOG. SYSLOG is a privileged command; only the master CLI, PID 2, can issue it.

The SYSLOG command has the following variations:

```
        /START [filename]
SYSLOG  /START
        /STOP
        filename
```

SYSLOG, without an argument, displays *OFF* if logging is off and *ON* if logging is on.

SYSLOG/START filename renames the old :SYSLOG file to filename and starts recording in a fresh :SYSLOG file. If you omit *filename,* the system appends new logging information to the current :SYSLOG, creating file :SYSLOG if it doesn't already exist.

SYSLOG/STOP stops recording in file :SYSLOG and closes it. The file must be closed before you can rename it. Shutting down AOS closes the log file, so you don't need to close it before shutting down.

SYSLOG filename simply renames :SYSLOG to filename without starting a new log file. You can then, if desired, start a new :SYSLOG by typing SYSLOG/START ↵.

For effective logging, you will want to rename the old log file periodically, and then start a new :SYSLOG file. You can do this either when you start logging (SYSLOG/START filename ↵) or after you have stopped logging (SYSLOG/STOP ↵; then SYSLOG filename ↵).

One way to identify the system log file is to start it with

) SYSLOG/START ddmmmyy ↵

where ddmmmyy is the date on which the preceding logfile was started; e.g., 18SEP85. This identifies each log clearly and uniquely. For even more clarity, you could add the suffix .LOG to the filename.

For example, a sequence might go

AOS CLI REV date time	Bring up AOS system.
) DATE ↵ *30-NOV-..*	Check date... It's 30 November.
) SYSLOG/START 29NOVLOG ↵	Rename old logfile to previous date and start new :SYSLOG file.
) SUPERU ON ↵	Turn on Superuser and
*) LOGEVENT Running new rev ↵	Log an Event in :SYSLOG.
*) UP ↵	Bring up multiuser environment.
. . .	System runs...
) DOWN ↵	Bring it down...
. . .	
*) BYE ↵ *SYSTEM SHUTDOWN*	
. . .	
AOS CLI REV	Bring it up...
) DATE ↵ *02-DEC-..*	Check date...
) SYSLOG/START 02DEC.LOG ↵	Start new SYSLOG and rename old.
) UP ↵	Bring up the multiuser system again.
. . .	

There are a number of different approaches to this. The easiest way is — as the example shows — to start a new log each time a system is brought up. This allows you to put the SYSLOG commands in the UP macro; all you need is a way to pass the date as a filename to the SYSLOG commands. Such a macro, and a fellow macro that it needs, might look something like Figure 9-2.

File :SYSLOG keeps a running record, which you can check even while SYSLOG is ON.

User Writes and Reads with SYSLOG Files

Any Superuser can post messages into the active :SYSLOG file, under the general-purpose code, with the CLI command LOGEVENT. Or, a user program can do the same thing with the ?LOGEV system call, if it uses event code 1065 (2051 octal). Either of these writes an event record into the log file (but neither returns an error if logging is not on).

SYSLOG files are not pure ASCII, so they are not directly readable. But user programs can read these files if they use the proper record formats. Also, users can read log files (including the active one) via the DISPLAY program. Most important, they can get reports on log files using the REPORT program.

As with all files, a user cannot read or write a log file without read or write access to it — unless he or she has Superuser on.

The REPORT Program

The REPORT program interprets, sorts, and displays the records in system log files. REPORT is in :UTIL, filename REPORT.PR. It is available to any user process. It is a CLI Help topic, so anyone can get help on it by typing HELP *REPORT ⤶.

REPORT can write the logged information to any device or file you name. It can report specific types of information and can report on multiple log files.

REPORT always reports user and device error information. Summary switches for additional information include /C (CPU events) and /X (XODIAC events).

The standard report has two parts. The first page(s) describe users, the last page describes device errors. For each user, information proceeds as follows.

- username (* means that this user has a Super or Access Devices privilege);
- user tape or disk mount requests (if any);
- console connect time. This appears as 0:00 if SYSLOG was started or stopped while this user was logged on;
- number of pages printed;
- CPU time in hours, minutes, seconds and milliseconds;
- I/O blocks: the number of 512-byte disk blocks read or written;
- page-seconds. This relates memory usage and CPU time; it is memory pages used multiplied by the number of CPU seconds;
- number of processes created for this user during his or her connect time;
- hardware errors (on the page that follows the user summary).

Note that the device error report can be very helpful — even essential — to DG engineers whom you may call upon for help. If you are having any kind of stubborn hardware or software problem, you should run SYSLOG, and get and print at least a device and CPU (/C) log regularly.

Text of SYSLOG_UP.CLI Macro

```
[!EQ,%0/%,/H]
     WRITE This macro starts SYSLOG with a SYSLOG command that renames
     WRITE the old logfile to the date it was started -- then moves
     WRITE the old log file into directory :LOGS. If the old logfile
     WRITE was started today the macro simply starts it.
     WRITE You could name this macro SYSLOG_UP.CLI and call it from
     WRITE the UP macro. Before this macro will work, someone must:
     WRITE ,, 1. Create it in the root directory.
     WRITE ,, 2. Create a macro named MAKE_DATE.CLI in the root directory.
     WRITE ,, 3. Create a directory named :LOGS  -- via CREATE/DIR :LOGS
     WRITE ,, 4. Create a file name LAST_LOG_DATE in the root -- this
     WRITE ,,,,, file must contain the string "ddmmmyy.LOG" where
     WRITE ,,,,, ddmmmyy are the date the original :SYSLOG was started:
     WRITE ,,,,, for example - 30NOV81.LOG. This starts things off.
     WRITE After these things have been done, the macro will do log
     WRITE housekeeping simply via the command ,,,SYSLOG_UP. It takes
     WRITE no arguments but the /H switch produces this help message.
[!ELSE]
     PUSH
     STRING [MAKE_DATE [!EXPLODE [!DATE]]].LOG
     [!NEQUAL,([!FILENAMES :LAST_LOG_DATE]),()]
           PUSH
           STRING [:LAST_LOG_DATE]
           [!NEQUAL,[!STRING/P],[!STRING]]
               SYSLOG/START/1=WARNING/2=WARNING  [!STRING]
               PUSH
               SUPERUSER ON
               DIR :
               ACL  [!STRING]  OP,OWARE +,RE
               WRITE Moving log file [!STRING] to directory :LOGS
               MOVE  :LOGS  [!STRING]
               DELETE [!STRING]
               POP
           [!ELSE]
               SYSLOG/START
           [!END]
     POP
    [!ELSE]
      SYSLOG/START
    [!END]

 SUPERUSER ON
 DELETE/2=IGNORE  :LAST_LOG_DATE
 STRING/L=:LAST_LOG_DATE
 ACL :LAST_LOG_DATE   +,R
 POP
 [!END]
```

Text of MAKE_DATE.CLI

```
%1%%2%%4%%5%%6%%8%%9%&
```

Figure 9-2. Example SYSLOG_UP and Companion Macros

For example, a command like

*) X REPORT/L=@LPT :LOGS:15SEP85.LOG ⮐

might produce a report that looks like Figure 9-3.

```
USER SUMMARY FROM FILE(S): :15SEP85.LOG

 USERNAME          CONNECT  TIME   PAGES      CPU TIME        I/O      PAGE NUMBR
                   CONSOLE  UNIT PRINTED                     BLOCKS    SECS PROCS
  BILL                1:14  0:00       5   0:00:05.847        517       437    10
  BRIDGET             4:30  0:00       0   0:10:04.370      34409     79391   170
 *FOO                 0:03  0:00       0   0:00:00.594         68        29     3
 *GAMMA               3:00  0:00       0   0:04:27.916      29761     31903    61
  JEFF                1:34  0:00       0   0:00:38.435        714      2662    19
  JOE                 2:47  0:00       6   0:02:22.956      11084     15046    80
  LUCY                4:27  0:00       0   0:04:28.486      30421     18306    45
  MARC                2:22  0:00       7   0:03:53.311      13029     26748    44
  MASM                2:29  0:00       0   0:02:29.910       6819      8080    43
  NETOP               0:00  0:00       0   0:00:02.451        103       150     2
 *OP                  2:46  0:34      20   0:15:28.560      82321     66996    59
  PAUL                0:01  0:00       0   0:00:00.418         61        23     2
  PETE                1:03  0:00      14   0:12:33.800     148183     53282    50
  STEVE               0:02  0:00       0   0:00:01.045         46        64     2
  SWAT                4:03  0:00      50   0:24:09.367      56109    172285   409
  USER                0:00  0:00       0   0:00:00.492         26        29     2


DEVICE SUMMARY FROM FILE(S): :15SEP85.LOG

 DEVICE  UNIT    ERRORS
 CODE         HARD  SOFT
  22       0     0    31
```

Figure 9-3. A Default Report

Running REPORT

To run a REPORT, use the command form

) XEQ REPORT[switches] [logfile-pathname][...]* ⮐

where

switches — select options as described next. If you omit switches, the report goes to the generic output file, @OUTPUT. By default, @OUTPUT is your console.

logfile-pathname — specifies the SYSLOG-generated logfile from which you want the report. If you omit a *logfile-pathname,* REPORT will use all the logfiles in the working directory. (System logfiles are a special type of file, LOG. REPORT reports on all files of this type.)

When it generates a report from multiple log files, REPORT processes the files in chronological order of creation time — regardless of the order in which you specify them.

REPORT Switches

REPORT *always* produces the default report on users and devices. You can request additional reports via one or more REPORT switches, described below.

REPORT can report only on events whose codes are recorded in a SYSLOG file. These codes are recorded only when AOS is running, with logging on. There are some events that AOS can't record, for the following reasons.

- The event forces AOS to panic. Some events indicate an error condition so serious that the safest course for AOS is to panic.
- The event occurs so often that logging overhead would degrade performance. If an event would occur very frequently, in the course of normal operations, AOS will not log it.
- AOS is not running when the event occurs.

Under any of these circumstances, the pertinent event code will not be written to SYSLOG. And REPORT will not be able to report the event.

Several REPORT switches include the functionality of other switches. For example, the /C switch includes reports on all 30-odd CPU-related switches . So — if you use the /C switch — you will see a report (usually with "NO EVENTS RECORDED") for each of the CPU-related switches.

The following switches are meaningful only for logs created on MV/8000 machines:

/ATU /IC /IOC /SC /SCP

All REPORT switches and descriptions follow. The SYSLOG event code, in form decimal(octal), is part of the description.

Switch	**What It Does**
/AFTER=dd-mmm-yy *[:hh:mm:ss]*	Reports only the events that occurred on or after the specified date or time. You can combine the /AFTER= and /BEFORE= switches to display any time period. For example, to view only those events occurring on December 4, 1985:

```
*) XEQ  REPORT/AFTER=4-DEC-85/BEFORE=4-DEC-85 & )
*&) pathname )
```

To see the action between 8 p.m. and 2 a.m. on this night:

```
*) X  REPORT/AFT=4-DEC-85:20:00:0& )
*&) /BEF=5-DEC-85:2:00:0  pathname )
```

• /ATU	Reports events in which the CPU Address Translation Unit (ATU) accelerator was disabled or enabled by an operator command (detected by the SCP). Normally, the ATU accelerator is enabled. Event codes are 122(172) and 123(173).
/BEFORE=dd-mmm-yy *[:hh:mm:ss]*	Reports entries that occurred on or before the specified date and time.
/C	Reports all CPU-related information. It includes all information gathered by the following switches: /ATU, /FATAL, /HANG, /IC, /IOC, /PW, /PWR, /SC, /SCP, and /TE switches. Most of these reports will be "NO xx EVENTS RECORDED."
/CT	Reports Connect Time (CT). For each console user, this report gives username, connect time, and console name (console name information is not in the default report). An example is shown in Figure 9-5. Event code is 1024(2000).

/DEVICES=dd.u *[+dd.u][...]*	Reports specific devices. The *dd* is the device code and *u* is the unit; e.g., 27.0 for DPF0. The default report always includes this information anyway. Event code is 4(4).
/ERCC	ERCC memory error log.
/EV	Reports all human-defined events (EV). Superusers can write messages to the log file with the CLI command LOGEVENT. Or, they can use the ?LOGEV system call, with general-purpose event code 1065 (2051). REPORT will include human-defined messages only if you include the /EV switch. If there are EV entries, the Event report will look like the one in Figure 9-4, below. As with all of the reports, it will start on its own page. REPORT prints only about 40 characters of the user message. It converts lowercase letters to uppercase and converts each nonprinting character into a space.
/FA	Reports a XODIAC functional level summary. By username, this includes the number of FTA connections, RMA requests, FTA I/O blocks, total packets transmitted and received, and total bytes transmitted and received. Also see the /X switch below. Event code is 1067(2053).
/FATAL	Reports fatal AOS errors (panics). ESD records these when you run it if logging is on. Event code is 41(51).
/FE	Reports XODIAC functional level errors. By username, the report includes the host, virtual circuit number, and error code for each network function that hit an error. Also see the /X switch below. Event code is 1068(2054).
/HANG	Reports on all AOS system hangs (deadlocks). ESD records this when you run it if logging is on. Event code is 42(52).
/I	Tells REPORT to ignore user-defined SYSLOG entries. User programs can write their own codes and messages to SYSLOG, via system call ?LOGEV. (The CLI LOGEVENT command writes only to code 1065 and isn't relevant here.) If the user-defined code is not one of the standard codes (described later on), REPORT will not accept it. So you should include this switch if programmers at your site have used ?LOGEV to write their own codes and messages into a log file on which you want a report.
/IC	Reports events in which the CPU instruction cache was disabled or enabled by an operator command (detected by SCP on MV/8000). Normally, this cache is enabled. Event codes are 117(165) and 118(166).
/IOC	Reports all IOC (input output controller) parity errors, as detected by the SCP on MV/8000 only.
/L*[=pathname]*	Sends the report to the generic @LIST file, set by your CLI LISTFILE command. Or, if you include *=pathname,* sends the report to the file named *pathname;* e.g., /L=@LPT.
/ME	Reports on soft errors in main memory on most systems except for MV/8000s. This log reflects problems in the CPU. (To obtain this log, you may need to issue the /OLD_ERCC switch.)
/MT	Reports on user unit mount requests. For each mount request, this report gives the user name and identifies the tape(s) or disk(s). Event code is 1025(2001).

/NA	Reports XODIAC connection level summary. For each network user, this shows the number of connections, connect time, packets transmitted and received, and number of bytes transmitted and received. (For reports on *all* XODIAC events, use the /X switch.) Event code is 1067(2053).
/NE	Reports XODIAC connection level errors. For each link, this includes each error code, associated diagnostic number, channel, virtual circuit number and transmit or receive status. A -1 for any entry means "not applicable". See also the /X switch. Event code is 1068(2054).
/OLD_ERCC	Append this switch to /ME if you have an S/230 or a C/330.
/PP	Reports pages printed. For each user, this report gives the number of pages printed per file; this information is not part of the default report. Event code is 1027(2003).
/PR	Reports privileged users. This gives the user names of all users who are logged on and who have the Superuser, Superprocess, or Access Devices privilege. A user need not activate the privilege to be listed. Event code is 1026(2002).
/PT	Reports all process terminations. For every process that lived and died during the log period, this describes the date and time of termination, full process name, and other information, shown in Figure 9-5. The /PT switch can produce a very sizable report, since a process terminates after every text editing session, compilation, batch job, and so on. Event code is 3(3).
• /PW	Reports CPU power failure and power restore. These are logged as one event, at the time of power restore. Event codes are 7(7) and 40(50).
• /PWR	Reports CPU power failure and power restore. These are logged as one event, at the time of power restore. Event codes are 100(144) and 101(145).
/RA	Reports a XODIAC RMA agent summary. By username, this gives total connect time and RMA requests. Also see the /X switch. Event code is 1030(2006).
/SA	Reports a DG/SNA accounting summary. By username, it gives connect time, number of logical units, number of request units received and sent, and number of bytes received and sent. Event code is 1066(2052).
• /SC	Reports events in which the CPU's system cache was disabled or enabled by operator command (detected by the SCP on MV/8000 only). This event cannot occur. Event codes are 119(167) and 120(170).
• /SCP	Reports status of SCP logging, as detected by the SCP on MV/8000 only. By default, SCP logging is on. Events are recorded only if someone has turned it off or on with an SCP command. Event codes are 96(140) and 97(141).
/TA	Reports a XODIAC File Transfer Agent (FTA) summary. By username, it gives the connect time, number of FTA connections, number of FTA blocks, packets transmitted and received, and bytes transmitted and received. Event code is 1064(2050).
• /TE	Reports overtemperature state, as detected by the SCP on MV/8000 only. Event code is 103(147).
/USERS=name *[+name][...]*	Reports only the user(s) specified with name(s). For example, for a report on users F77 and SWAT: `*) X REPORT/USERS=F77+SWAT pathname ↵`

/USPC — Reports Universal Power Supply Controller (USPC) events, like fan failures. This is meaningful only on computers that have a USPC (for example, S/280s). Event code is 1023(1777).

/X — Reports a XODIAC summary. This includes the reports from the /FA, /FE, /NA, /NE, /RA and /TA switches described above.

/XA — X25 summary log.

/XE — X25 error log.

SYSLOG and REPORT Examples

```
*) SYSLOG/START ↵
.
.
.
) XEQ   REPORT/USERS=F77+SACKVILLE   :SYSLOG ↵
.
.  (report on console)
.
) X   REPORT/AFT=[!DATE]/L=@LPT   :SYSLOG ↵
```

The preceding command reports on all entries that occurred today.

```
) X   REPORT/L=COBOL_BILL/BEF=17-DEC-85/AFT=12-DEC-85/USER=COBOL&↵
&)  :LOGS:17DEC85.LOG ↵
```

The preceding command reports on user COBOL, between 12 December, 1985 and 17 December 1985, and sends the report to file COBOL_BILL.

```
*) X   REPORT/EV/PT   :LOGS:28SEP85.LOG ↵
```

This command would produce the standard report; and it also produces event and process termination reports (like those shown in Figure 9-4 and Figure 9-5).

```
                              EVENT LOG DUMP

28-SEP-85 07:50:23   *** SYSLOG STARTED ***
28-SEP-85 07:50:30    RUNNING REV 3.00
28-SEP-85 17:32:06    RUNNING 4 BATCH STREAMS
28-SEP-85 20:03:38    REV 3.00 LOOKS GOOD
30-SEP-85 07:53:49   *** SYSLOG STOPPED ***
```

Figure 9-4. User Event Report (/EV Switch)

```
                         PROCESS TERMINATION OUMP

                      PROCESS NAME   ELAPSED  CPU TIME        I/O    PAGE
                                     TIME                  BLOCKS    SECS

28-SEP-85 07:50:23  *** SYSLOG STARTED ***
28-SEP-85 07:51:25  OP:003           0:00:07 0:00:00.077      107       2
28-SEP-85 08:52:45  GAMMA:011        0:00:08 0:00:00.156        9       6
28-SEP-85 08:53:01  GAMMA:011        0:00:10 0:00:00.312        9      12
28-SEP-85 08:53:10  GAMMA:012        0:00:03 0:00:00.047       13       2
28-SEP-85 08:53:16  GAMMA:012        0:00:03 0:00:00.095       13       4
28-SEP-85 08:53:17  GAMMA:STREAM_1   0:00:56 0:00:01.345      163      83
28-SEP-85 08:53:23  OP:CONO          0:00:56 0:00:00.041        0       2
28-SEP-85 08:53:28  LUCY:CON45       0:00:59 0:00:01.345       12      73
28-SEP-85 08:53:35  BRIDGET:CON44    0:03:35 0:00:01.410      184      86
    .        .          .               .        .             .       .
    .        .          .               .        .             .       .
    .        .          .               .        .             .       .
```

Figure 9-5. Process Termination Report (/PT Switch)

System Log Record Format

When AOS or a related program encounters a logable event, it gets the operating system to write the appropriate event *code* into the SYSLOG file. REPORT understands all these codes and translates them into ASCII for people to read.

As mentioned above, A [Super] user program can write any record code it wants into the active logfile via ?LOGEV. Unless the code used is one of the standard record codes, REPORT won't be able to interpret it. But later, application programs may want to read it for their own purposes.

This section details the structure of SYSLOG records for just this reason — so that application programs can read and digest log files in their own way. The programmers who write these will need to understand the SYSLOG record formats.

Programmers should open system log files for dynamic reads. They can declare an integer*2 array for the record header using a 0 base (e,g., in FORTRAN 77, INTEGER*2 HEADER(0:7)). This would allow them to use the 0-base subscripts that we show. They can also declare an integer*2 array for each different record length less 8 elements. At runtime, they can read the header array, check the length from words 0 and 1. Then, if the record is longer than the header, they can read the rest of the record to the array of its (length-8). Then, in the header array, they can check word 5 for the message code. If this is a code they want, they can break down the header and array, and format them for output, and then read the next record. If they don't care about the code, they can simply read the next record.

They can get the record length from the two-word length descriptor by — in FORTRAN 77 — equivalencing element 0 of the descriptor to a 4-byte integer.

The record header, which begins each record, is shown in Figure 9-6. The sixth 16-bit word in the header is a code that indicates the record type. In this and following figures, all subscript/offsets and record codes are decimal. In the records themselves, all numeric values are octal.

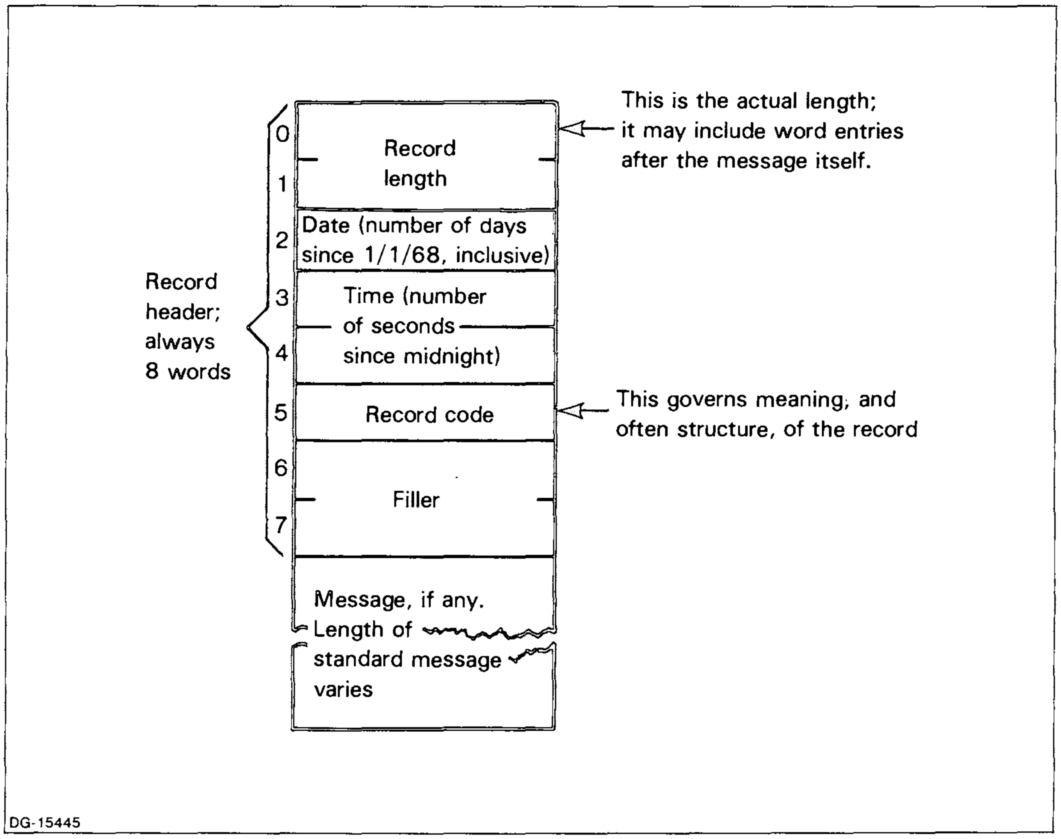


Figure 9-6. SYSLOG Record Header

Record Formats

The following "standard" record types are written by SYSLOG and understood by REPORT. (REPORT does some sorting of these records, depending on the switches you apply in the REPORT command.)

Figure 9-7 describes the records whose length is only the header. Figure 9-8 describes those records that are longer than the header. SYSLOG stores numeric values in octal and ASCII characters as ASCII. Where padding is needed, SYSLOG uses nulls (ASCII 000). The symbol "#" means "number of".

Code (decimal)	Meaning	Message Length (words 8 through n)
0	Unused	0
1	SYSLOG logging on	0
2	SYSLOG logging off	0
3	Process termination	24
4	Device error	4
5	Unused	0
6	ERCC error	3
7	Power fail	3
40	Power restored	0
41	Unused	0
42	Unused	0
43	Unused	0
44	Unused	0
45	Unused	0
73	SCP reset (not recorded) (MV/8000)	0
74	SCP request compl (not recorded) (MV/8000)	0
75	Host-SCP error (not recorded) (MV/8000)	0
76	Host-SCP buffer full error (MV/8000)	0
77	SCP time-out (MV/8000)	0
79	SCP-request-to-host error (MV/8000)	0
80	SCP buffer not cleared (MV/8000)	0
81	Host-request-to-SCP error (MV/8000)	0
96	SCP logging enabled (MV/8000)	0
97	SCP logging disabled (MV/8000)	0
98	Main processor halt	0
99	BOOT issued (MV/8000)	0
100	Power fail	0
101	Power restore	0
102	Air flow fault	0
103	Overtemp fault (not recorded)	0
104	Transfer to battery backup	0
105	Reserved	16
106	ERCC error (MV/8000)	16
107	Microsequencer parity error	0
108	Sys cache parity error	0
109	Cache to Bank controller parity error	0

DG-25612

Figure 9-7. SYSLOG Record Codes and Lengths (Excluding Header) (continues)

Code (decimal)	Meaning	Message Length (words 8 through n)
110	IOC bus parity error	0
111	S-bus timeout	0
112	S-bus parity error	0
113	Operating system error (unused)	0
114	Diskette log error (MV/8000)	0
115	Infinite protection fault	0
116	Infinite page fault	0
117	Instruction cache enabled	0
118	Instruction cache disabled	0
119	Reserved	0
120	Reserved	0
121	System RESET	0
122	ATU accelerator enabled	0
123	ATU accelerator disabled	0
125	XEQ DTOS command (MV/8000)	0
126	Bad return from DTOS (MV/8000)	0
127	HALT command (MV/8000)	0
128	CONTINUE command (MV/8000)	0
129	START command (MV/8000)	0
130	INIT command (MV/8000)	0
131	Bank controller ERCC report disable	0
132	Good return from DTOS (MV/8000)	0
133	Hard interrupt (not recorded)	0
1024	Console connect time	25
1025	Unit mount time	25
1026	Privileged user log on	9
1027	Pages printed	25
1028	Reserved	
1029	Reserved	
1030	RMA accounting	12
1031	Reserved	
1064	FTA accounting	21
1065	General event(LOGEVENT)	varies
1066	DG/SNA accounting	22
1067	X.25 Accounting	38
1068	X.25 Error	14

DG-25327

Figure 9-7. SYSLOG Record Codes and Lengths (Excluding Header) (concluded)

Word	Code 3
8–23	Username, process name (in ASCII, 32 characters, 16 words)
24–25	Elapsed time, seconds(octal)
26–27	CPU time, seconds
28–29	I/O blocks
30–31	Page seconds

Word	Code 4
8	Device code
9	Unit #
10	Status
11	Retry count

Word	Code 7
8	Date reported
9–10	Time reported

Word	Code 41
8–9	Address of panic caller
10–11	Panic subroutine return temporary
12–13	Panic save AC2 temporary
14	Narrow temporary
15–16	Panic code
17–18	Value #1 (AC0)
19–20	Value #2 (AC1)
21–22	Value #3 (AC2)
23–24	Value #4 (AC3)
25–26	Value #5
27–28	Value #6
29–30	Value #7
31–32	Value #8
33–34	Stack pointer
35–36	Frame pointer
37–38	Stack limit
39–40	Stack base

Word	Code 43, 44, or 45
8	Code 43, 44, or 45
9	Cause (CPU I/O access, shift, other)
10	Physical page #
11	Double word on module
12	Syndrome bits

Word	Code 106
8	ERCC code
9	Module #
10	Plane #
11	Bit #

Word	Code 1024 or 1025
8–15	Username (in ASCII, 16 characters, 8 words)
16–31	Device name (in ASCII, 32 characters, 16 words)
32	Minutes used

Word	Code 1026
8–15	Username (in ASCII, 16 characters, 8 words)
16	Privilege bits

Word	Code 1027
8–15	Username (in ASCII, 16 characters, 8 words)
16–31	Device name (in ASCII, 32 characters, 16 words)
32	Pages printed

ID-00011

Figure 9-8. Structure of SYSLOG Records Longer than 8 Words (continues)

Figure 9-10 shows a breakdown of the first record in octal and decimal:

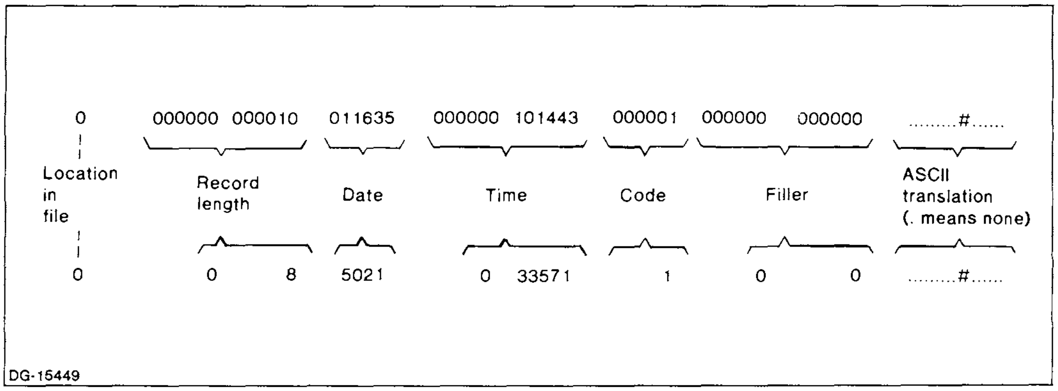


Figure 9-10. Octal and Decimal Versions of a SYSLOG Record

The second record starts at location 10 octal (8). It is 40 octal (32) words long. The code in the fifth word is 3, which means that the record is a process termination message. The ASCII appears in the field off to the right (this field is not in the record, but is a convenience supplied by DISPLAY). Looking at the fields of this record, you can see (in the last line) the elapsed time, CPU time, I/O blocks, and page-seconds.

The third record shown appears later on in the file. It has a record length of 50 octal (40). The fifth word, 2001 octal (1025), describes it as a unit mount. Again, you can see the ASCII off to the right. JACK used the mounted tape for 3 minutes.

The fourth and fifth records appear still later in the file. Each is 24 octal (16) words long. The fifth word in each header is 2051 (1065). This means it is an event record, written by the LOGEVENT command. Once more, you can see the ASCII on the right.

The report that the REPORT program would generate from this log file would have a lot of columnar information, and its numbers would be decimal — as shown in the earlier REPORT figures.

File Backup: DUMP and PCOPY

Your site should regularly make backup copies of its entire disk file system. Then, if files are lost or accidentally deleted, you can restore them from your backup media. (Backup procedures are sometimes called *archiving*, and the media used sometimes called *archives*). How often you back up your system depends on the size of your installation. You can do it daily, on alternate days, or weekly. It's prudent to do it daily, if possible.

There are two approaches to archiving. They are the

- DUMP-oriented approach (uses CLI command DUMP);
- physical copy approach (uses PCOPY program).

There are other backup utilities available with specific DG products, but you need either DUMP or LOAD to do system-level backup (to restore disks to a state where the other products can work).

DUMP and PCOPY are meant for backup on your *local* system: the system that produces the material for backup. Some systems provide for *remote* backup, and use a XODIAC network to move copies of files to a larger central DG system. The file copies on the central system then serve for backup. For details on backup over the network, see the latest version of the *XODIAC™ Network Management Guide for System Managers and Operators.*

Choosing a Backup Program

The DUMP approach is file oriented. It can back up specific files and directories, allowing some or all of them to be restored with the LOAD command. DUMP runs while AOS is up. You can use it with labeled or unlabeled tape, or labeled diskettes. You can use DUMP to copy entire directories (a full backup) or only those files that have changed since a given date (incremental backup).

PCOPY copies all occupied space on an LDU — no more, no less. PCOPY runs as a stand-alone program; i.e., PCOPY that runs when AOS is down; it can copy any LDU.

DUMP and PCOPY can both be used for backup; for example, you can run PCOPY once a week and do incremental backups on the intervening days.

Tradeoffs between the two approaches are as follows.

DUMP and LOAD	PCOPY
Run on any LDU while AOS is up.	Runs on any LDU while AOS is down.
Are selective, allowing you to specify files by date modified and/or a template.	Copies all occupied disk blocks; you cannot specify date modified or a template.
Allow you to dump or load files onto more than one LDU in a single command.	Copies or restores one LDU per run.
Can be run in batch, and allow you to create friendly CLI macros that prompt the system operator.	Requires correct tape or disk mounting and issues its own prompts.
Cannot dump or restore an installed AOS system or SYSBOOT; you must install these from a system tape or diskettes via the Installer if desired. They will dump and restore system .SY files, and you can always bring up any of these without installing it.	Copies all occupied blocks, including those of an installed system and SYSBOOT (but doesn't install a disk bootstrap — you must run the Installer to install this).
Transfer data slower than PCOPY (but since you can be selective, backup with DUMP requires less data transfer).	Transfers data much faster than DUMP. But, PCOPY copies everything, whether or not you need it: DUMP can copy only things that have changed.
Work best in a cycle with one full backup followed by several incrementals.	Works best in sites that have large-scale changes. It copies all occupied blocks and can't do incremental backups.
Work with any disk model; tape or diskettes are the preferred backup media.	Works with any disk model; tape, diskette, or disk work as backup media.

Generally, you might consider the DUMP approach if

- its versatility appeals to you; and/or
- your system cannot be shut down for an hour or so each day to copy the master LDU; and/or
- you don't have very large database files on a separate LDU (built from model 6236 or 6239 disks);
- you must use diskettes for backup; and/or
- you don't have two identical disk units with removable packs (three for a multiple-disk LDU).

You might consider PCOPY if

- you like the idea of doing only full backups of entire LDUs;
- you must use cartidge tape for backup;
- you have two identical disk units, with removable packs (three for a two-disk LDU); and
- your system can be shut down for a while each day to copy the master LDU.

PCOPY can be used with DUMP (with a little care); this has many benefits particularly if you are using cartridge tape.

Preparing for Backup

Before starting a backup, be sure you have enough time and ample backup media. Specifics are given in each backup section. Tapes and diskettes should have paper labels on them — on which you should write the date, volume ID, and other information you might need to restore the backed-up material.

If you use the DUMP approach, all important files must be closed during a dump. (With PCOPY, all files *are* closed, since the LDU to be backed up is not initialized.) If a file is open when dumped, changes may have been made that have not yet been written to disk. This means that the file dumped is not current — and can make the backup useless.

Ideally, during backups, all timesharing users will be logged off. All CEO, INFOS II, or DG/DBMS processes should have been shut down normally. If abnormal shutdown has occurred since the last backup, be sure that recommended verification programs have been run (to ensure the integrity of databases).

Backup Using DUMP — with Tape

This section gives some pointers on using the DUMP approach, with tape, and shows some macros to help you do it. To use DUMP with diskettes, see the next major section.

DUMP and LOAD are very versatile commands, with switches that are fully described in the CLI manual.

DUMP, without a directory-name template, copies all files and directories in the working directory to a dump file. It writes a header before each file, and includes such information as the file's name, date of creation, and ACL. The directory structure on the disk is maintained; later, if you make the directory from which DUMP was issued your working directory, and load the dump file, the system will try to recreate the original structure. So, it's *very important* that the initial directory be the same for LOAD as it was for DUMP. The root directory (:) is the best directory from which to start backups and restorations.

The dump file (which becomes the input file for load) can be a labeled tape file like ROOT or LDU2, or tape file number like @MTB0:3. For tape, we recommend labeled tapes, although they are not mandatory. Labeled tapes allow a dump to span more than one tape volume.

You can label tapes using the LABEL program, from any CLI process, at any time. Use the tape unit name, and then the label (maximum and recommended length six characters). For example

```
) X   LABEL   @MTB0   FULL00 ↲
```

The structure of tape labels and headers is explained in the section, "About User Mount Requests," in Chapter 8. Generally, you don't need to know about label structure to use LABEL for backups.

DUMP Macros for Tape

One approach to full and incremental dumps to tape is shown in the macros in Figures 9-11 and 9-12. A restore macro for tape follows later on, in Figure 9-13. Backup on diskettes is explained in the diskette section, later in the chapter.

The macros in Figures 9-11 and 9-12 are for backup. The FULL_DUMP.CLI macro (see Figure 9-11) does a full backup of all initialized LDUs, excluding system-only directories. The INC_DUMP.CLI macro (see Figure 9-12) does an incremental backup of all initialized LDUs. The INC_DUMP.CLI macro dumps only those files created or modified since the last backup — based on file LAST_DUMP_DATE, which contains the date and time of the last backup. These backup macros set up tape volume IDs for each dump, and prompt the system operator through each step. They specify a volume ID list of 10 tape volumes — and allow the volume ID list to be extended.

The CLI macros shown in Figures 9-11, 9-12, and 9-13 are designed to do a full backup, incremental backup, and restoration of files. DG shipped them with AOS to ease backup. The macros are not executable until you edit them. The macros explain how to do this. Type the name of the macro for instructions; for example type FULL_DUMP ↓.

A dump can archive fewer than 8 Mbytes of material (on one 400-foot tape), or range up to 800 Mbytes at 1600 bpi (on 21 2400-foot tapes). The limiting factor is the volume ID list, which can't exceed 128 characters for a single MOUNT command. Volume IDs of less than 6 characters would allow more volumes to be specified.

A 2400-foot tape, at 1600-bpi density, dumped with a buffer size of 8192 bytes, can hold about 39 Mbytes or 76,000 disk blocks. On an MTB unit, a 2400-foot tape takes roughly 20 minutes to fill.

A 1000-foot tape, at 1600-bpi density, can hold about 16 Mbytes or 31,300 disk blocks. On an MTC unit, model 6125, filling a 1000-foot tape may take about 20 minutes. A cartridge tape, used with a buffer size of 8192, can hold about 15 Mbytes; but takes 45 to 50 minutes to fill.

With a cartridge tape, you can shorten backup dramatically by using PCOPY instead of DUMP. With a buffer size of 2 Kbytes, PCOPY can fill a cartridge in 6 minutes — but stores only 8 Mbytes instead of 15. Thus, with cartridge tape, you may want to use PCOPY for full backups and DUMP for incrementals.

You may want to use these macros — or expanded versions of them — at your own site. Each macro dumps with the /RETAIN=0 switch. This allows the tape set to be reused immediately, if desired, without relabeling. You may want to use a nonzero retention period; for example, /RETAIN=28 for full backups and /RETAIN=7 for incremental backups.

```
[!EQUAL,1,2]
[!equal,comment,]
This macro does a full dump of the entire system, excluding system
directories and files, to labeled tape, in batch. It also sets up the
labeled tape volume IDs to be used. The tape volume IDs are FULLnn,
where nn is the sequence number of the tape in the dump.

The tape fileset name used for the dump is ROOT. The macro also explains
how to specify a dump to a second fileset (like UDD), if your system is
large enough to make a second tape fileset worthwhile.

This template used for the dump excludes DG-supplied directories. You
can restore these from your system SYSTAPE and from the INC_DUMP tapes.

A file named LAST_DUMP_DATE is needed in the root directory to start
things off. If it doesn't exist when you run this macro, the macro will
create it.

To execute the macro, the person acting as system operator must go to a
user console, log on as a Superuser (like OP), and type the macro name.
The macro will then help him/her through the procedure. Tape mounts
and CONTROL @EXEC MOUNTED commands at the system console will be needed.

The command and pseudomacro syntax used is explained in the CLI manual.[!end]

[!equal,comment,] Make sure EXEC Operator mode is on. [!end]

[!nequal,(ON),([!operator])]
  write Error - Operator is not on. Please go to the system console and type
  write [!read CONTROL @EXEC OPERATOR ON newline. Return here and press newline.]
[!end]

[!equal,comment,]  Check for file LAST_DUMP_DATE. If it doesn't exist,
tell user, and then create it. [!end]

push
dir  :
superuser  on

[!nequal,([!filenames LAST_DUMP_DATE]),()]
    [!equal,([!filenames DATE_DUMP_STARTED]),()]
         write  Last dump was done on ,,, [LAST_DUMP_DATE]
         write/l=DATE_DUMP_STARTED   [!date]:[!time]
    [!else]
         write Full dump was started at ,,, [DATE_DUMP_STARTED]
    [!end]
```

Figure 9-11. FULL_DUMP.CLI Macro, for a Labeled Tape Dump (continues)

```
[!else]
    write File LAST_DUMP_DATE doesn't exist! I am creating a LAST_DUMP_DATE
    write file that specifies today's date and the current time as follows:
    write [!date]:[!time]
    write If you don't do a full dump today please delete LAST_DUMP_DATE
    write/l=LAST_DUMP_DATE   [!date]:[!time]
[!end]

write
write  *** ,, Full dump of directory [!dir] on [!date]:[!time] ,, ***
write
[!equal,comment,] Get person's name and write to file DUMPERS_NAME.[!end]

delete/2=ignore  :DUMPERS_NAME
write/l=:DUMPERS_NAME  [!read  Please type your name: ]
pop
dir/i

[!equal comment,] Post the needed mount and dump commands in batch.[!end]
qbatch/m/operator
superuser  on
dir  :

[!equal,comment,]  Issue a mount command for 10 labeled tape volumes.
The volids for full backup are  full00, full01,...full09. (First, to prevent
problems, delete any linkname of the name MYTAPE in user's directory.)[!end]

delete/2=ignore   :udd:[!username]:MYTAPE

mount/extend/volid=full00/volid=full01/volid=full02/volid=full03&
/volid=full04/volid=full05/volid=full06/volid=full07/volid=full08&
/volid=full09   MYTAPE   Please mount the longest tape you can.

[!equal,comment,] Now dump to the labeled tape file. The fileset name
used here is ROOT. The template does not exclude UDD or any nonmaster
LDUs. If you want a dump separate from ROOT (like UDD), exclude the
directory(s) via the dump template. Then insert a "mount/volid=dirnn"
and "dump ...   dir:#" command after the  dismount command. For example,
to exclude UDD from the ROOT dump and include it in a separate dump:
1. Add the text   \UDD:#   to the ROOT dump template.
2. Add commands   mount/volid=UDD00/vol=UDD01... MYTAPE Please   and
                  dump  ... MYTAPE:UDD  UDD:#\?+.BRK\+.ED\+.LS\?+.TMP
   after the dismount command.

The listing file (/l) in batch is the line printer. [!end]

write/l  This is a full backup of directory [!dir] started at
write/l  [!date]:[!time] by [DUMPERS_NAME]
```

Figure 9-11. FULL_DUMP.CLI Macro, for a Labeled Tape Dump (continued)

```
dump/v/l/buffersize=8192/retain=0    &
     :UDD:[!username]:MYTAPE:ROOT               &
     #\HELP\NET\PAGE\PATCH\PER\PROC\QUEUE\SWAP\SWAP.SWAP\SYSGEN\UTIL&
\?+.BRK\+.ED\?+JOB\+.LS\?+.TMP  NET:NETGEN:+

dismount  MYTAPE  Full dump of directory [!dir] is done.
pause 5
send 2 ,, Check the printer -- file [!username].OUTPUT.n  -- for errors.
send 2 ,, A list of files dumped is printed in file [!username].LIST.n

[!equal,comment,] Delete and rename file LAST_DUMP_DATE... but do it
only if file DATE_DUMP_STARTED exists. This prevents LAST_DUMP_DATE
from being deleted if the batch job was aborted. The terminating )
is needed for qbatch/m syntax. [!end]

[!nequal,([!filenames DATE_DUMP_STARTED]),()]
  delete/2=ignore  LAST_DUMP_DATE
  rename/2=ignore  DATE_DUMP_STARTED  LAST_DUMP_DATE
[!end]
)
[!ELSE]
  write This macro is nonexecutable. To make it executable you must use a
  write text editor to change the 2 in line 1 to 1 -- both numbers must be 1.
  write
  write Backup is very important. Before using the edited macro routinely
  write for backup you should understand how it works. Please test this
  write macro. Use it to back up files. Then try restoring files using
  write the RESTORE_TAPE macro -- before relying on this macro for backup.
[!END]
```

Figure 9-11. FULL_DUMP.CLI Macro, for a Labeled Tape Dump (concluded)

Text of Macro INC_DUMP.CLI

```
[!EQUAL,1,2]
[!equal,comment,]
This macro does an incremental dump of the entire system, excluding
system directories and files, to labeled tape, in batch. It also sets up
the labeled tape volume IDs to be used. The tape volume IDs are INCRnn,
where nn is the sequence number of the tape in the dump.

The tape fileset name used for the dump is ROOT. The macro also explains
how to specify a dump to a second fileset (like UDD), if your system is
large enough to make a second tape fileset worthwhile.

This template used for the dump includes DG-supplied directories. You
can restore these from your system SYSTAPE and from dumps produced by
this macro.

A file named LAST_DUMP_DATE is needed in the root directory to start
things off. If it doesn't exist when you run this macro, the macro will
recommend recovery steps and stop.

To execute the macro, the person acting as system operator must go to a
user console, log on as a Superuser (like OP), and type the macro name.
The macro will then lead him/her through the procedure. Tape mounts
and CONTROL @EXEC MOUNTED commands at the system console will be needed.

The command and pseudomacro syntax used is explained in the CLI manual.[!end]

[!equal,comment,] Make sure EXEC Operator mode is on. [!end]

[!nequal,(ON),([!operator])]
  write Error - Operator is not on. Please go to the system console and type
  write [!read CONTROL @EXEC OPERATOR ON newline. Return here and press newline.]
[!end]

[!equal,comment,]  Check for file LAST_DUMP_DATE. If it doesn't exist,
skip to end, give recovery advice, and stop. [!end]

push
dir  :
superuser  on

[!nequal,([!filenames LAST_DUMP_DATE]),()]
  [!equal,([!filenames DATE_DUMP_STARTED]),()]
        write  This dump will back up all dumpable files created or
        write  modified since [LAST_DUMP_DATE]
        write/1=DATE_DUMP_STARTED   [!date]:[!time]
  [!else]
        write The last incremental dump was started at [DATE_DUMP_STARTED]
  [!end]
```

Figure 9-12. INC_DUMP.CLI Macro, for an Incremental Labeled Tape Dump (continues)

```
  write
  write  *** ,, Incremental dump of directory [!dir] on [!date]:[!time] ,, ***
  write

  [!equal,comment,] Get person's name and write to file :DUMPERS_NAME.[!end]
  delete/2=ignore  :DUMPERS_NAME
  write/l=:DUMPERS_NAME  [!read  Please type your name: ]

  pop
  dir/i
  [!equal comment,] Post the needed mount and dump commands in batch.[!end]
  qbatch/m/operator
  superuser  on
  dir  :

  [!equal,comment,] Issue a mount command for 10 labeled tape volumes. The
  volids for incremental backup are INCR00, INCR01, etc. (First, to prevent
  problems, delete any linkname of the name MYTAPE in user's directory.)[!end]

  delete/2=ignore :udd:[!username]:MYTAPE

  mount/extend/volid=incr00/volid=incr01/volid=incr02/volid=incr03&
/volid=incr04/volid=incr05/volid=incr06/volid=incr07/volid=incr08&
/volid=incr09  MYTAPE   Please mount the longest tape you can.

  [!equal,comment,] Now dump to the labeled tape file. The fileset name
  used here is ROOT. The template does not exclude UDD or any nonmaster
  LDUs. If you want a dump separate from ROOT (like UDD), exclude the
  directory(s) via the dump template. Then insert a "mount/volid=dirnn"
  and "dump ...   dir:#" command after the  dismount command. For example,
  to exclude UDD from the ROOT dump and include it in a separate dump:
  1. Add the text   \UDD:#   to the ROOT dump template.
  2. Add commands   mount/volid=UDD00/vol=UDD01... MYTAPE Please   and
                    dump  ... MYTAPE:UDD  UDD:#\+.BRK\+.ED\+.LS
     after the dismount command.

  The listing file (/l) in batch is the line printer. [!end]

  write/l  This is an incremental backup of directory [!dir] started at
  write/l  [!date]:[!time] by [DUMPERS_NAME]
  string [LAST_DUMP_DATE]
  write/l  It dumps all files created or modified since [!string]

  dump/v/l/buffersize=8192/retain=0/after/tlm=[!string]   &
      :UDD:[!username]:MYTAPE:ROOT    &
     #\NET\PAGE\PER\PROC\QUEUE\SWAP\SWAP.SWAP\SYSGEN&
\?+.BRK\+.ED\?+JOB\+.LS\?+.TMP   NET:NETGEN:+
```

Figure 9-12. INC_DUMP.CLI Macro, for an Incremental Labeled Tape Dump (continued)

```
  dismount MYTAPE  Incremental dump of directory [!dir] is done.
  pause 5
  send 2 ,, Check the printer -- file [!username].OUTPUT.n -- for errors.
  send 2 ,, A list of files dumped is printed in file [!username].LIST.n

  [!equal,comment,] Delete and rename file LAST_DUMP_DATE... but do it
  only if file DATE_DUMP_STARTED exists. This prevents LAST_DUMP_DATE
  from being deleted if the batch job was aborted. The terminating
  paren is needed for qbatch/m syntax. [!end].

  [!nequal,([!filenames DATE_DUMP_STARTED]),()]
     delete/2=ignore  LAST_DUMP_DATE
     rename/2=ignore  DATE_DUMP_STARTED  LAST_DUMP_DATE
  [!end]
)
[!else]
     write Error - File LAST_DUMP_DATE doesn't exist! Cannot do incremental
     write dump without it. Suggest full dump. If you can remember the
     write date of the last backup -- full or incremental -- create a
     write file named LAST_DUMP_DATE containing this date and retry %0%
     write The form of the date is dd-mmm-yy -- for example
     write 09-MAY-85
     pop
[!end]
[!ELSE]
  write This macro is nonexecutable. To make it executable you must use a
  write text editor to change the 2 in line 1 to 1 -- both numbers must be 1.
  write
  write Backup is very important. Before using the edited macro routinely
  write for backup you should understand how it works. Please test this
  write macro. Use FULL_DUMP and this macro to back up files. Then try
  write restoring files using the RESTORE_TAPE macro -- before relying on
  write this macro for backup.
[!END]
```

Figure 9-12. INC_DUMP.CLI Macro, for an Incremental Labeled Tape Dump (concluded)

A typical sequence with the backup macros might go as shown in the following example sections.

Full Dump Example

You make sure timesharing users are logged off and that CEO, INFOS II, and/or DG/DBMS processes (if you have them) are shut down.

You log onto a user console with a privileged username and password. Then:

) FULL_DUMP ↲

Last dump was done on 22-MAY-85:17:56:22 (The first time you run it, it displays a *does not exist* and *creating* message about file LAST_DUMP_DATE.)

*** *Full dump of directory : on 23-MAY-85:18:22:40* ***
Please type your name: SAM ↲

(pause)

Go to system console, which beeps and displays

*FROM PID n (EXEC): ** EXPLICIT LABELLED MOUNT ***
FROM PID n : MID=n, USER=xxx, PID=n, EXEC SUB_TREE PID=n
FROM PID n : CURRENT VOLUME: full00, ALL VOLUME(S): full00, full01,
FROM PID n : full02, full03, full04, full05, full06, full07, full08, full09
FROM PID n : REQUEST IS 'Please mount the longest tape that you can.'
FROM PID n : UNIT(S) ARE: NONE
FROM PID n : RESPOND: CONTROL @EXEC MOUNTED @UNITNAME
FROM PID n : OR: CONTROL @EXEC REFUSED

Write-enable the longest tape your largest tape unit can hold. Mount the tape on a unit, say MTB0. Use unit switches to select the highest density (if there's a choice). If you've already used the macro, the tape's already labeled and you needn't relabel it. If you haven't run the macro on this tape, relabel it via

) X LABEL @MTB0 FULL00 ↵

Tell EXEC that the tape is mounted

) CX MOUNTED @MTB0 ↵

The dump begins on this volume. If all specified material fits, the macro issues DISMOUNT and EXEC prompts for a dismount on the system console. But probably, all material won't fit on one volume — and, at the end of tape mark, EXEC will respond

*FROM PID n (EXEC): ** NEXT VOLUME ***
FROM PID n : MID=n, USER=xxx, PID=n, EXEC SUB_TREE PID=n
FROM PID n : CURRENT VOLUME: full01, ALL VOLUME(S): full00, full01,
FROM PID n : full02, full03, full04, full05, full06, full07, full08, full09
FROM PID n : REQUEST IS 'Please mount the longest tape you can.'
FROM PID n : UNIT(S) ARE: @MTB0
FROM PID n : RESPOND: CONTROL @EXEC MOUNTED
FROM PID n : OR: CONTROL @EXEC REFUSED

Remove volume FULL00 from the tape unit, mount volume FULL01 on it, and type CX MOUNTED ↵. (If you have several tape units, you can premount volume FULL01 and maybe FULL02 on other units, saving steps.) If you mount a volume that has the wrong label or file set ID, EXEC will respond with *MOUNT ERROR* and *WRONG VOLUME* messages, and prompt for the correct volume from the MOUNT volume ID list. You can either dismount the current tape and find and mount the correct tape, or type X LABEL ↵ to relabel the tape; then type CX MOUNTED.

This sequence repeats until the system has dumped all files (excluding those excluded in the dump template) to the tape file set.

If 10 volumes aren't enough for the full backup, EXEC prompts for another tape. Mount one, using LABEL to label it if needed (use the same volume ID sequence: FULL11, FULL12, and so on). When you see the following message on the system console

*FROM PID n : (EXEC) ** WAITING TO BE DISMOUNTED ***

.
.
.

FROM PID n : REQUEST IS 'Full dump of directory : done.'
FROM PID n : RESPOND CONTROL @EXEC DISMOUNTED

You can dismount the tape(s) and type CX DISMOUNTED ↵. This completes the full dump. You can file the unused volumes (if any). If you needed extra volumes, note their names on the dump listing and think about adding their volume IDs to the dump macro MOUNT command.

As with any user mount, you can refuse a request to restart (CX REFUSED ↵); or you can cancel the batch job with QCANCEL sequence-number ↵ or CX FLUSH sequence-number ↵. Canceling the batch job will abort a dump in progress.

After the dump completes, make sure each reel has a paper label with its volume ID, the file set name (ROOT), and the date. Store the reels safely, in order. You can use them again for another full dump.

You can reuse the archive tapes as often as desired — for example, every 6 weeks for full dumps, and every 10 working days for incremental dumps. This depends on how many backup sets (full and all incremental) you want to keep.

Incremental Dump Example

As with the full dump, you make sure timesharing users are logged off and that CEO, INFOS II, and/or DG/DBMS processes (if you have them) are shut down.

You log on to a user console with a privileged username and password. Then:

) INC_DUMP ↵

Last dump was done on 23-MAY-85:18:22:40 (If the file with the last dump date doesn't exist, it will tell you to run a full dump and stop.)

This dump will back up all dumpable files created or modified since 23-MAY-85:18:22:40

*** *Incremental dump of directory : on 24-MAY-85:18:10:30* ***

Please type your name: SAM ↵

(pause)

Go to system console, which beeps and displays

*FROM PID n (EXEC): ** EXPLICIT LABELLED MOUNT ***
FROM PID n : MID=n, USER=xxx, PID=n, EXEC SUB_TREE PID=n
FROM PID n : CURRENT VOLUME: incr00, ALL VOLUME(S): incr00, incr01,
FROM PID n : incr02, incr03, incr04, incr05, incr06, incr07, incr08, incr09
FROM PID n : REQUEST IS 'Please mount the longest tape that you can.'
FROM PID n : UNIT(S) ARE: NONE
FROM PID n : RESPOND: CONTROL @EXEC MOUNTED @UNITNAME
FROM PID n : OR: CONTROL @EXEC REFUSED

As with the full backup, write-enable the longest tape your largest tape unit can hold. Mount it on a unit, say MTB0. Use drive switches to Select the highest density (if there's a choice). If you've already used the macro, the tape's already labeled and you needn't relabel it. If you haven't run the macro on this tape, relabel it via

) X LABEL @MTB0 INCR00 ↵

Tell EXEC that the tape is mounted

) CX MOUNTED @MTB0 ↵

The dump begins on this volume. If all modified files fit, the macro issues DISMOUNT and EXEC prompts for a dismount on the system console. Often, all incremental dump material will fit on one volume. But if it won't fit, the system will reach the end of tape and EXEC will respond

*FROM PID n (EXEC): ** NEXT VOLUME ***
FROM PID n : MID=n, USER=xxx, PID=n, EXEC SUB_TREE PID=n
FROM PID n : CURRENT VOLUME: incr01, ALL VOLUME(S): incr00, incr01,
FROM PID n : incr02, incr03, incr04, incr05, incr06, incr07, incr08, incr09
FROM PID n : REQUEST IS 'Please mount the longest tape you can.'
FROM PID n : UNIT(S) ARE: @MTB0
FROM PID n : RESPOND: CONTROL @EXEC MOUNTED
FROM PID n : OR: CONTROL @EXEC REFUSED

If EXEC prompts for another volume, remove the old one from the unit, mount the next (here, volume INCR01), and type CX MOUNTED ↵. (If you have several units available, you can premount volume INCR01 on another unit, saving a step.) If you mount a volume that has the wrong label or file set ID, EXEC will respond *MOUNT ERROR* and *WRONG VOLUME*, and prompt you to mount the correct volume from the MOUNT volume ID list. You can either find and mount the correct tape or type X LABEL to relabel a scratch tape; then type CX MOUNTED ↵.

This sequence repeats until the system has dumped all files (excluding those excluded in the dump template) to the tape file set.

Nearly always, 10 volumes will be enough for an incremental backup. (If not, EXEC will ask for another tape. Mount one and label it if needed using LABEL and the original volume ID sequence — INCR11, INCR12, and so on.) When you see the following message on the system console

*FROM PID n : (EXEC) ** WAITING TO BE DISMOUNTED ***

.

.

.

FROM PID n : REQUEST IS 'Incremental dump of directory : done.'
FROM PID n : RESPOND CONTROL @EXEC DISMOUNTED

You can dismount the tape(s) and type CX DISMOUNTED ↵. This completes the incremental dump. You can file the unused volumes. If you needed extra volumes, note their names on the dump listing and think about adding their volume IDs to the dump macro.

As with any user mount, you can refuse a request (CX REFUSED ↵) to restart; or you can cancel the batch job with QCANCEL sequence-number ↵ or CX FLUSH sequence-number ↵. Canceling the batch job will abort a dump in progress.

After the dump completes, make sure each reel has a paper label with its volume ID, the file set name (ROOT), and the date. Store the reels safely, in order. You can use them again for another incremental dump.

You can reuse the backup tapes as often as desired — for example, every 6 weeks for full dumps, and every 10 working days for incremental dumps. This depends on how many backup sets (full and all incremental) you want to keep.

These dump macros, with the restore macro, can save a lot of time, effort, confusion, and data.

Verifying Dumped Material on Tape

Nearly always, dumped material will load perfectly, restoring your old file structure. But, if you really need to make sure there are no tape errors, you can do so quite quickly via the CLI command COPY. To verify a dumped tape, have it on a tape unit and type the following commands

```
) COPY/IMTRSIZE=80   @NULL   @MTxn:0 ↵        (tape unitname, file 0)
) COPY/IMTRSIZE=8192   @NULL   @MTxn:1 ↵      (tape unitname, file 1)
) COPY/IMTRSIZE=80   @NULL   @MTxn:2 ↵        (tape unitname, file 2)
```

These COPY commands do fast reads on the physical devices. They use a buffer size (IMTRSIZE for COPY) of 80 for the labeled tape leader, 8192 (specified in the DUMP) for the data on tape, and 80 for the trailer. The output file is generic file @NULL, so the data read is not written; this speeds things up. If all tape volumes read without errors, you can be virtually certain that it will load without errors.

You can verify each tape from the system console, if desired, after the entire dump or while EXEC is prompting for the next volume.

Restoration Macro

The RESTORE_TAPE.CLI macro (see Figure 9-13) restores material from either a full or incremental backup. By default, it uses the volume IDs used for incremental backups, but you can tell it to restore from a full backup. The RESTORE_TAPE.CLI macro requires only the original disk structure — as created by the system tape (SYSTAPE) — to restore incremental and full backups.

Text of Macro RESTORE_TAPE.CLI

```
[!EQUAL,1,2]
[!equal,comment,]
This macro restores files from either an incremental or full dump (done
by the FULL_DUMP or INC_DUMP macros), in batch. The macro expects the tape
volume IDs to be of the form FULLnn or INCRnn, as created by the dump
macros. The default is INCRnn.  The macro accepts pathname arguments, so
you can use it to restore individual files. To work, the macro requires
a basic system file structure, which you can restore using your system
SYSTAPE tape if needed.

The tape fileset name used for the restoration is ROOT. The macro also
explains how to specify a dump to a second fileset (like UDD), if you
have dumped files to filesets other than ROOT.

To execute the macro, the person acting as system operator must go to a
user console, log on as a Superuser (like OP), and type the macro name.
Tape mounts and CONTROL @EXEC MOUNTED commands at the system console
will be needed. The most efficient course is to restore the last
incremental backup first, then go backwards through the most recent full
backup.

The command and pseudomacro syntax used is explained in the CLI manual.[!end]

[!equal,comment,] Make sure EXEC Operator mode is on. [!end]

[!nequal,(ON),([!operator])]
  write Error - Operator is not on. Please go to the system console and type
  write [!read CONTROL @EXEC OPERATOR ON newline. Return here and press newline.]
[!end]

write The default restoration assumes you are restoring from incremental
write dump tapes -- with volume IDs INCR00 and INCR01 and INCR02 and
write so on. If you want to restore a full backup -- with volume IDs
write FULL00 and FULL01 and FULL02 and so on -- you must specify FULL.
write
write  For incremental press newline.
string [!read For full type FULL and press newline. ]
write
```

Figure 9-13. RESTORE_TAPE.CLI Macro, to Restore Dumped Files (continues)

```
delete/2=ignore  :UDD:[!username]:?DUMP_TYPE.TMP
[!equal,([!string]),(FULL)]
  write ,, *** Restoring full backup. Need volume IDs FULL01 FULL01 etc. ***
  write/l=:UDD:[!username]:?DUMP_TYPE.TMP  FULL
[!else]
  write  *** Restoring incremental backup. Need volume IDs INCR01 INCR01 etc. ***
  write/l=:UDD:[!username]:?DUMP_TYPE.TMP  INCR
[!end]

[!equal,comment,] Get person's name and write to file :RESTORERS_NAME.[!end]
push
superuser on
dir  :
delete/2=ignore   :RESTORERS_NAME
write
write/l=:RESTORERS_NAME    [!read  Please type your name: ]
pop
dir/i

[!equal comment,] Post the needed mount and dump commands in batch.[!end]
qbatch/m/operator
superuser  on
dir  :

[!equal,comment,] Issue a mount command for 10 labeled tape volumes. The
volids depend on the choice above -- INCRnn or FULLnn. (First, to
prevent problems, delete any linkname of the name MYTAPE in the user's
directory.) [!end]

delete/2=ignore   :udd:[!username]:MYTAPE
string  [:UDD:[!username]:?DUMP_TYPE.TMP]

mount/extend/volid=[!string]00/volid=[!string]01/volid=[!string]02&
/volid=[!string]03/volid=[!string]04/volid=[!string]05/volid=[!string]06&
/volid=[!string]07/volid=[!string]08/volid=[!string]09   MYTAPE   &
Please mount the correct tape volume.

[!equal,comment,] Now load from the labeled tape file. If the person
included pathname arguments 1-n, include them in the load command line.
If he/she omitted arguments, restore the entire dump.
```

Figure 9-13. RESTORE_TAPE.CLI Macro, to Restore Dumped Files (continued)

```
The fileset name used here is ROOT. It does not include any directory
that was excluded from ROOT in the dump macros. If you have a dump
fileset separate from ROOT, you must load it separately, by fileset
name. Use the syntax shown in the mount and load commands above, with
your own custom volid names and fileset name. If the dump file includes
any nonmaster LDUs, be sure they're initialized before you load from
the dump tape. (Otherwise, the load command will recreate the
dump file directory on the master LDU.)

The listing file (/l) in batch is the line printer. [!end]

write/l  This is an [!string] restoration of directory [!dir] started
write/l  at [!date]:[!time] by [RESTORERS_NAME]

load/v/l/buffersize=8192/recent      &
     :udd:[!username]:MYTAPE:ROOT    &
     %1-%

dismount MYTAPE  [!string] restoration of directory [!dir] is done.
pause 5
send 2 ,, Check the printer -- file [!username].OUTPUT.n -- for errors.
send 2 ,, A list of files restored is printed in file [!username].LIST.n
send 2 ,, Don't forget to restore the other dumps, if this applies.
delete/2=ignore  :UDD:[!username]:?DUMP_TYPE.TMP

[!equal,comment,]  The following paren is needed for qbatch/m syntax. [!end]
)
[!ELSE]
  write This macro is nonexecutable. To make it executable you must use a
  write text editor to change the 2 in line 1 to 1 -- both numbers must be 1.
  write
  write Backup is very important. Before using the edited macro routinely
  write for backup you should understand how it works. Please test this
  write macro. Back up some files using FULL_DUMP and INC_DUMP. Then
  write restore files using this macro before relying on the macro set for
  write backup.
[!END]
```

Figure 9-13. RESTORE_TAPE.CLI Macro, to Restore Dumped Files (concluded)

About Restoring Files from Dump Tapes

Restoring falls into two categories: restoring one or more files, and restoring one or more LDUs. The first category is more common, and is easier and faster.

Restoring One or More Files

Usually, people do file restoration when someone has accidentally deleted a file (directory), or group of files. Perhaps someone was careless with DELETE and a template character — or, for whatever reason, you want to restore files that were backed up to tape.

There are two things to consider when you do this kind of restoration — the tape set(s) needed, and the pathname template.

The tape set(s) you use to restore depend on the date that the lost file(s) were last modified. If the files were created since the last backup, then they weren't backed up and cannot be restored. Otherwise, use the backup that occurred soonest after the files were modified (incremental or full).

If you can't determine when the files were last modified, check the backup listings. If the name of a lost file appears in any listing, then you know the file is in that backup. In the worst case, without a listing or dates, you must restore the last full backup set; then the earliest incremental backup; then next incremental backup; and so on. This is a good reason to keep your backup listings — especially for incremental backups.

After deciding on the tape set, you must choose one or more pathname templates (unless you want to restore the entire backup). The RESTORE macro allows template arguments in which you can specify a directory, a specific file, or a directory and pathname template.

You can restore all files in and below a directory (including subordinate directories), with the template

```
pathname-from-root:#
```

You can restore all files in and below a user's directory with the template

```
UDD:username:#
```

NOTE: Because the pathname starts in the root directory, it does not start with :, the name of the root directory.

For DG's Comprehensive Electronic Office (CEO) files, the directory structure and restoration procedure differs from that for standard AOS files. To restore CEO files, see *Managing the CEO® System.*

File Restoration Example

As an example, assume Andy accidentally deleted two files named REPORT.MAY and SUMMARY. He last modified them a week ago. (If the lost files were last modified on different dates, it may be most efficient to work from the last full backup). First, you must find and get the appropriate tape set.

You need a template for the restoration. The two filenames have the letters MA in common. So, you could use the template

```
UDD:ANDY:#:+MA+
```

This would work, but it might restore many matching, unwanted files. You can be more specific. The two names each have at least one character — and no period — following the A. So you could refine the template to

```
UDD:ANDY:#:+MA*-
```

Or, if you know which directory the files were in, you could use the specific pathnames:

```
pathname-from-root:REPORT.MAY    pathname-from-root:SUMMARY
```

Let's say you decide on the most general course — the +MA+ template, which covers many possibilities.

You log on to a user console with a privileged username and password. Then:

) :RESTORE_TAPE UDD:ANDY:#:+MA+ ↵ (Start the macro, specifying the desired template.)

The default restoration assumes you are.... (Macro describes the two kinds of restorations.)

.

.

.

For incremental press newline. (Press ↵ to select incremental.)
For full type FULL and press newline. ↵

Restoring incremental backup -- expect volume IDs INCR00...

Please type your name: SAM ↵

(pause)

Go to system console, which beeps and displays

*FROM PID n (EXEC): ** EXPLICIT LABELLED MOUNT ***
FROM PID n : MID=n, USER=xxx, PID=n, EXEC SUB_TREE PID=n
FROM PID n : CURRENT VOLUME: INCR00, ALL VOLUME(S): INCR00, INCR01,
FROM PID n : INCR02, INCR03, INCR04, INCR05, INCR06, INCR07, INCR08, INCR09
FROM PID n : REQUEST IS 'Please mount the correct tape volume.'
FROM PID n : UNIT(S) ARE: NONE
FROM PID n : RESPOND: CONTROL @EXEC MOUNTED @UNITNAME
FROM PID n : OR: CONTROL @EXEC REFUSED

Mount the first tape in the file set on a unit, say MTB0.

Tell EXEC that the tape is mounted

) CX MOUNTED @MTB0 ↵

The restore begins from this volume. If this is the only volume in the file set, or if the system has restored all files you specified (you must have specified entire filenames, not templates), then the macro issues DISMOUNT and EXEC prompts for a dismount on the system console. For the sake of this example, assume another volume is needed. At the end of tape mark on volume INCR00, EXEC will respond

*FROM PID n (EXEC): ** NEXT VOLUME ***
FROM PID n : MID=n, USER=xxx, PID=n, EXEC SUB_TREE PID=n
FROM PID n : CURRENT VOLUME: INCR01, ALL VOLUME(S): INCR00, INCR01,
FROM PID n : INCR02, INCR03, INCR04, INCR05, INCR06, INCR07, INCR08, INCR09
FROM PID n : REQUEST IS 'Please mount the longest tape you can.'
FROM PID n : UNIT(S) ARE: @MTB0
FROM PID n : RESPOND: CONTROL @EXEC MOUNTED
FROM PID n : OR: CONTROL @EXEC REFUSED

Remove volume INCR00 from the tape unit, mount volume INCR01 on it, and type CX MOUNTED ↵. (If you have several tape units, you can premount volume INCR02 on another units, saving a step.) If you mount a volume that has the wrong label or file set ID, EXEC will respond *MOUNT ERROR* and *WRONG VOLUME*, and prompt you to mount the correct volume in the MOUNT volume ID list. Dismount the tape; find and mount the correct tape; then type CX MOUNTED ↵.

This sequence repeats until the system has restored the files or read the entire file set of tapes. When you see the following message on the system console

*FROM PID n : (EXEC) ** WAITING TO BE DISMOUNTED ***

.

.

.

FROM PID n : REQUEST IS 'INCR restoration of directory : is done.'
FROM PID n : RESPOND CONTROL @EXEC DISMOUNTED

you can dismount the tape(s) and type CX DISMOUNTED ↵. This completes the restore. Check the printer for verification of files restored; and store the tape reels safely.

As with any user mount, you can refuse a request (CX REFUSED ↵) to restart; or you can cancel the batch job with QCANCEL sequence-number ↵ or CX FLUSH sequence-number ↵. Canceling the batch job will abort a restoration in progress.

With the files restored, tell Andy to check them — and to check his directories for unwanted, restored files. He can then delete the unwanted files.

(If Andy had created the files in CEO (the full product, not Word Processor - Independent), and had accidentally deleted "documents" named REPORT.MAY and SUMMARY, you could suggest that he check the CEO Wastebasket. In the full CEO, documents are not actually deleted until a program called Janitor has been run. Until then, documents that people delete can be retrieved from the Wastebasket. If Andy's documents are not in the Wastebasket, you can restore them from the last backup — as described in *Managing the CEO® System.*)

Shortening a Restoration

When you mention a specific filename (like UDD:CHRIS:MYFILE) in a restoration, the restoration will end, and the CLI prompt will return *as soon as the file has been copied to disk.*

The restoration will take longer if you use a template like UDD:CHRIS:MYF+ because the restore will continue through the last tape, even after the desired file has been copied. This happens because AOS can't tell, until it reaches the end of the last tape, that there is no matching filename in the tape set.

Generally, this means that it's desirable to give a specific filename, if you can. But if you do, be sure it's the correct one — if you make a mistake with a specific filename, the system will take you all the way through the tape set, and not restore the file you want restored. If you think this may be true, cancel the batch job via QCANCEL n ↵ (n is the batch job sequence number, returned by the QDISPLAY command. Then start again.

Restoring an Entire LDU Using Dump Tapes — When and How

The time may come when you need to restore all backed-up material to an LDU. This can happen when a disk wears or fails in such a way that the system can no longer read it.

If you have checked with DG, and either acquired a new disk or decided to rebuild the old one, follow these steps.

1. To restore the master LDU, continue. To restore a nonmaster LDU, skip to step 7.
2. Get the tailored system tape you made with SYSTAPE after testing your tailored AOS system. (If you don't have a tailored system tape, use the lastest system tape you received from DG; later, you'll need to generate a tailored system.)
3. Return to Chapter 3 and execute all the numbered steps there, using your tailored system tape or the DG tape. (If you're not using a tailored system tape, generate a tailored AOS system, patch it, start it, and make a system tape as described in Chapter 4.)
4. If you have any DG software products that were not backed up (those under :UTIL, for example), install these products as described in product documentation.

5. Use PREDITOR to create an operator profile; then create the line printer/batch queues and bring up EXEC (as shown in Chapter 5).
6. Get all your incremental backup sets and your last full backup set of tapes.
7. To restore a nonmaster LDU: make sure this LDU is formatted with the Disk Formatter and has the desired name. Run FIXUP on it for good measure. Then initialize it into your system as usual.

   If the nonmaster LDU needs DG software installed (for example, FORTRAN 77) and that software was not backed up, make the LDU your working directory, and install the software.

   Follow this step for all nonmaster LDUs you want to restore.
8. Log onto a user console that's physically close to the system console. Restore your most recently dumped incremental backup tape set. Proceed backwards through the incrementals until you have restored them all. Then restore the most recent full backup.

   After each restoration ends, check the printed batch output file for error messages. If you see a *CONTROL POINT DIRECTORY MAX SIZE EXCEEDED* message, the LDU is full; you, (or other users) must delete some previously deleted and backed-up files on the disk, to free some space, before you can continue the restoration process.

   The files to delete may be in user directories, where they can be deleted with the CLI DELETE command, or in CEO, where they can be deleted with the "Delete" menu choice. In CEO, you will also need to run the Janitor to complete the deletions.

   When you have some disk space free, say 1000 to 5000 blocks or more (type SPACE :↵), proceed with the next restoration.
9. Make sure you return all tapes to their covers and store them safely.
10. You're done! You've recreated the entire LDU(s), having lost only a little work (the files created or changed since the last backup).

If you have XODIAC networking software, the :NET:NETGEN files were backed up and have been restored. You can regenerate host and FMA files by running NETGEN and specifying the network spec filename.

NOTE: The RESTORE macro restores files with their original creation times, but it changes the time last modified to the time you restore the files. This means that your next backup must be a full backup, via the FULL_DUMP macro, since the dates last modified aren't real. It also means that the FILESTATUS command with the /BEFORE/TLM= switches won't help identify "old" files after the restoration.

Backup using Dump — with Diskettes

This section gives some pointers on using diskettes for backup: handling diskettes, using dump with diskettes, and some macros to help you do it. (To use DUMP with tape, see the section "Backup Using DUMP — with Tape."

Backup Sets of Diskettes

One hard disk can hold 71.2 Mbytes (about 139,000 disk blocks); a diskette can hold 368 Kbytes (about 720 disk blocks). Thus, over a hundred diskettes are needed to back up the biggest disk available on a diskette-only system. For smaller disks (38.6 or 15 Mbytes), you need proportionately fewer diskettes. At the beginning, you can get by with very few diskettes because your disk will not have been filled up. To get the approximate number of diskettes needed, type SPACE :↵ and divide the CUR figure by 720; then subtract 10 diskettes for system files (which won't be backed up, since you can restore them from system diskettes).

Assume 3 to 10 diskettes for each incremental backup, and multiply this number by the number of incremental backups you plan between full backups. Add the total incremental backup number to the full backup number. This gives the approximate number of diskettes needed for one backup set.

You *can* get along with one backup set — but ideally, you should have alternate sets. This allows you to keep the last backup set intact, and use the *previous* backup set for the new backup.

Whatever schedule and plan you come up with, be sure you have enough diskettes on hand when you start a full backup. If you run out of diskettes and can't complete a FULL_BACKUP normally, the backup will be incomplete. To make it complete, you'll need to start again from the beginning, with enough diskettes. (This caution also applies to incremental backups, but with less force, since only a few minutes are wasted if you need to restart.)

Handling and Storing Diskettes

Diskettes are important — and fragile. Some handling cautions and hints follow.

- Store diskettes in their outer envelopes; remove a diskette from its outer envelope just before you use it.
- Hold a diskette by the edges of the envelope only. Avoid touching the diskette surface (exposed in oval cutout on the inner envelope). The oil on your finger could make that part of the diskette unreadable.
- If a diskette has no paper label, apply one — properly labeled diskettes make life easier. Labels go on the smooth, seamless side of the inner envelope, at the top (with the write-enable notch to the right). *Avoid the oval cutout.* Remove the sticky-backed label from its backing and apply it to inner envelope.
- To write on a diskette label, use only a felt-tipped pen. A pencil or ball-point pen can score the diskette surface — destroying some or all data on it.
- A diskette must be properly inserted — the system can't read one that's improperly inserted. On a DESKTOP GENERATION system, when you insert a diskette, hold it by the edge, with label (seamless) side facing right and the write-enable notch up. For other diskette drives, insert the diskette with the label side up. In either case, the diskette should slide in smoothly and come to a firm stop.
- Diskettes wear. If you see the message *SOFT ERROR, DEVICE 20 n,* this is a warning. Consider subsituting a new diskette for the one in unit *n.*

  A *HARD ERROR* or *PHYSICAL UNIT FAILURE* means that the rest of the diskette is unreadable or unwritable. If this message appears during backup, replace the diskette. If it appears during a restoration, restart the restoration. If the hard error recurs, this probably means you cannot restore data from the rest of the diskettes in this set. (Perhaps there is another, earlier set you can use.) When the restoration is complete, you should bring down AOS and run FIXUP on your LDU, since inconsistent information may have been copied to it.
- Don't bend or twist a diskette. A crease on the surface means data loss.
- Cold and heat can harm diskettes. Keep them temperate (between 10 and 50 degrees Centigrade, between 50 and 125 degrees Fahrenheit).
- A magnetic field can erase part or all data on a diskette. Keep them away from electric motors, magnets, and transformers.
- Don't turn computer power off while a diskette is inserted; remove diskettes first. Turning off power while a diskette is inserted can lead to data loss.
- Diskettes must be hardware formatted (different from software formatting, which is done by the Disk Formatter). Diskettes you get from DG are shipped formatted, but diskettes from another vendor are *not* hardware formatted.

A diskette that is not hardware formatted will produce a *HARD ERROR* or *PHYSICAL UNIT FAILURE* message. To hardware format a diskette, run the hardware formatting program described in the hardware documentation. It's a good idea to hardware format non-DG diskettes immediately after you purchase them.

- Generally, you should not write-protect diskettes. AOS can't access a write-protected diskette as a directory, or write to it in any way.

Dumping to Labeled Diskettes

DUMP, without a directory-name template, copies all files and directories in the working directory to a dump file. It writes a header before each file, and includes such information as the file's name, date of creation, and ACL. The directory structure on the disk is maintained; later, if you make the directory from which DUMP was issued your working directory, and load the dump file, the system will try to recreate the original structure. So, it's *very important* that the initial directory be the same for LOAD as it was for DUMP. The root directory (:) is the best directory from which to start backups and restorations.

The CLI can read, write, and label diskettes — allowing you to dump and load material using a sequence of labeled diskettes. This kind of sequence, with multiple diskette volumes, is essential for backup if your system doesn't have a tape unit. Labeled diskettes are needed in any situation where someone wants to write to diskette and the material involved exceeds the capacity of one diskette.

The command that enables the CLI to access labeled diskettes is OPERATOR. It is available to any process, on any console. The CLI OPERATOR command is very different from EXEC's command of the same name (described in Chapter 8).

The CLI OPERATOR command has the form

$$\text{OPERATOR} \quad \begin{bmatrix} ON \\ OFF \end{bmatrix}$$

If you omit arguments, the CLI displays the current status of operator mode. If you include ON, the CLI turns on operator mode (if not already on). If you include OFF, the CLI turns off operator mode (if on).

When operator mode is on, the CLI can dump to, label, and load from labeled diskettes. Operator mode can be turned on by any CLI process, on any console. It stays on until turned off or until the CLI process terminates.

When operator mode is off, the CLI cannot access a diskette by label. An attempt to do so provokes a *NO OPERATOR AVAILABLE* error message. The CLI can access a diskette by *physical* unit name; for example, via the DUMP or LOAD commands, as in

```
) DUMP/V  @DPM0  MYFILE ↵
```

The OPERATOR /LABEL Switch

The /LABEL switch tells the CLI to label diskettes — during a dump operation — without warning you before doing so.

On any dump to a labeled diskette, the CLI checks the diskette before starting to write. If the diskette has the volume ID (volid) expected, the CLI starts writing to it.

If the diskette does not have the volid expected, the CLI's action depends on whether you included the /LABEL switch when you turned the Operator mode on. If you omitted /LABEL, the CLI displays an error message and asks if you want it to relabel the diskette. If you included /LABEL (for example, OPERATOR/LABEL ON ↵), the CLI relabels the diskette without asking for your okay.

The /LABEL switch is useful for dumps when the diskettes you want to use are not labeled, or when you don't care about their labels. It eliminates the label specification step on a label conflict.

Whether or not you include /LABEL, the CLI will create volids for you as described in the next section, under "volid."

Labeled Diskette Access

After you turn operator mode on (with or without the /LABEL switch), you can use labeled diskettes. To access a labeled diskette, use the form

```
command   @LFD:volid:filename
```

where:

command — is the DUMP command (to write to), or the LOAD command (to read from), one or more labeled diskettes.

On a dump, if you omitted /LABEL from the OPERATOR command, the CLI checks to see if the retention period specified in the previous dump has elapsed. If not, the CLI asks if you want to relabel the diskette. The default retention period is 90 days. With the DUMP /RETAIN= switch, you can specify a different period, including 0 days (which allows the diskette set to be reused immediately).

@LFD — is the filename that indicates a labeled diskette (labeled floppy disk). You must include @LFD for labeled diskette access.

volid — is the volid that's on the first diskette (on a load) or the volid you want written on the first diskette (on a dump). A volid can be written to a diskette either by the CLI or by the LABEL utility. The CLI's labeling mechanism is more convenient. Volids are limited to six legal filename characters.

On a load, the volid you specify must match the volid on the first diskette. If not, the CLI will signal an error when it checks the diskette label.

On a dump, the CLI checks the diskette label. If the label is correct, the dump proceeds. If the label is wrong, the CLI's message depends on whether you included the /LABEL switch when you turned operator mode on. If you omitted /LABEL, the CLI warns you that the label doesn't match the one expected. Then it gives you the choice of relabeling the diskette or inserting another diskette. If you included /LABEL, the CLI relabels the diskette with a new volid.

For access to the second and subsequent diskettes, the CLI creates default volids, based on the original volid you specified. It creates a numeric sequence by adding 1 to each volid. If the volid you originally specified doesn't end with a number, the CLI will add a number, space permitting. If there isn't room for the CLI to add a number, it will drop as many characters as needed to create the next volid. For example

Volid you specify	Default volids created by CLI
VOL1	VOL1 (for first diskette) VOL2 (for second diskette, if needed) . . . VOL9 (for ninth diskette, if needed) VOL10 (for tenth diskette, if needed)
SYSTEM	SYSTEM (for first diskette) SYSTE1 (for second diskette, if needed) . . . SYSTE9 (for ninth diskette, if needed) SYST10 (for tenth diskette, if needed)

When you choose a volid, we suggest that you end it with a number, for consistency with the CLI's label creating sequence (VOL1, VOL2, VOL3, and so on).

filename is the filename of the diskette fileset (on a LOAD command); or it is the filename you want to create for the diskette fileset (on a DUMP command).

This filename applies to the entire fileset of one or more diskettes. The same filename used for a labeled diskette dump must be used to load the diskette set; if not, the CLI will report a *FILE DOES NOT EXIST* message when it checks the label of the first diskette. The filename is limited to 17 legal filename characters.

With operator mode on, after you type a command of the form @LFD:volid:filename, the CLI will prompt you to insert a diskette in a specific unit, like this:

PLEASE INSERT A DISKETTE IF NOT ALREADY INSERTED
UNIT [@DPM0] VOLUME ID [volid] ? [Y]

The default diskette unit is @DPM0. The displayed volid is the one you specified (first diskette) or the next sequential number *after* the preceding volid. If you want to override the default unit, type N ↲. The CLI then allows you to specify a different default unit.

The volid and filename you specify cannot be changed during a dump or load command. To use a different volid or filename, you must abort the command and restart it. For example, assume you type

) DUMP/V @LFD:XVOL1:MYFILE ↲

The CLI displays the *PLEASE INSERT* message shown above. But then you decide that you prefer VOL1 to XVOL1 as a first volid. To change the volid, abort the dump by typing CTRL-C CTRL-A; then retype the command with the new volid (in this case, DUMP/V @LFD:VOL1:MYFILE ↲).

For any dump, each diskette must be hardware formatted, but need not be software formatted with the Disk Formatter. During any load operation, the first diskette must have the volid you specify, and each of the following diskettes must have the volid that the CLI expects. If you make a mistake, like typing the wrong volid or filename, abort the command via CTRL-C CTRL-A and restart it, as shown above.

Diskette Access Control

The default ACL for any diskette unit does not allow user access. If you want users who don't have the Superuser privilege to be able to use the unit, we suggest you set the unit ACL to allow for this. The best place to do this is in the system UP macro, with a command like

ACL @DPM0 +,WARE

While you are loading from (or dumping to) diskette, the system does not automatically protect your diskette from access by other users. For example, user JACK can write to a diskette that you (OP) are trying to read. With Superuser on, or if you have owner (O) access to the diskette unit, you can change the unit ACL to prevent other users from accessing it. For example, you might type

*) ACL/V @DPM0 ↲	
DPM0 +,WARE	(It displays the ACL.)
*) ACL @DPM0 OP,OWARE ↲	(Set ACL for your use.)

Then use the unit. When you're finished with the unit, restore its original ACL so others can use it.

Labeled Diskette Example

The following example shows a full backup of the entire system. The CLI macros to make such backups easier are shown in the next section. We're including a full backup example, without macros, here to show how it works.

In this example, the diskettes have not been used for a system backup. They have not been labeled and the person doing the backup chooses to include the /LABEL switch for labeling without the extra step of typing the new labels.

. . .	(Ensure that all users have logged off and that all server processes — like CEO and INFOS II, if present — are shut down.)
) DIR : ⤶	(Get to the root directory.)
) OPERATOR/LABEL ON ⤶	(Turn operator mode on, with /LABEL.)
) SUPERUSER ON ⤶	(Turn Superuser on for file access.)
*) ACL @DPM0 OP,OWARE ⤶	(Give yourself sole access to diskette unit.)
*) DUMP/V/L=DFILE @LFD:FULL01:BACKUP ⤶	(Start the dump...)
PLEASE INSERT A DISKETTE IF NOT ... INSERTED *UNIT [@DPM0] VOLUME ID [FULL01]? [Y]* ⤶	(CLI prompts for diskette. Insert one and press ⤶.)
	(Since /LABEL was included, the CLI labels the diskette without asking for confirmation. The dump proceeds, with filenames listed to file DFILE. Each diskette takes about 2 minutes to fill.)
PLEASE INSERT NEXT DISKETTE *UNIT [@DPM0] VOLUME ID [FULL02]? [Y]* ⤶	(After the diskette has been filled, the CLI prompts for the next one, incrementing the volid. You remove the diskette, insert the ⤶ next one; press ⤶.
	(Again, the CLI labels the diskette and the system continues dumping files. The dump proceeds through other diskettes.)
PLEASE INSERT NEXT DISKETTE *UNIT [@DPM0] VOLUME ID [FULL19]? [Y]*	(Again, after a number of diskettes, The CLI prompts for the next one...)
⤶	(You remove the diskette, insert the next one; press ⤶.)
	(Again, the CLI labels it and the system starts dumping files. Then...)
PLEASE REMOVE THE DISKETTE	
*) QPRINT DFILE ⤶ *QUEUED...*	(The full backup is done. You print the listing file...)
*) DELETE/V DFILE ⤶ *DELETED DFILE*	(Delete the listing file.)
*) ACL @DPM0 +,WARE ⤶ *)	(Restore the old ACL.)

The full backup is done. You can now remove the diskette from unit DPM0 and store all diskettes safely. Later on, if needed, all files could be restored to the disk via the commands:

```
) DIR  : ↲
) OPERATOR  ON ↲
) SUPERUSER  ON ↲
*) ACL  @DPM0  OP,OWARE ↲
* ) LOAD/V/R  @LFD:FULL1:BACKUP ↲
```

Diskette Backup Macros

The CLI macros shown in Figures 9-14, 9-15, and 9-16 are designed to do a full backup, incremental backup, and restoration of files via diskettes. DG shipped them with AOS to ease backup on systems without tape. The macros are not executable until you edit them. The macros explain how to do this. Type the macro name for instructions; type FULL_BACKUP ↲.

The backup macros do not turn Operator mode on, so you will need to turn it on before using a macro. The first time you create any full or incremental diskette file set, we suggest that you turn Operator mode on with the /LABEL switch, to avoid extra keystrokes. Since you need to label each diskette the first time through, you can save typing effort by using /LABEL.

To reuse a diskette set, you should use Operator without /LABEL, so the CLI will check labels and allow you to change diskettes if you make a mistake and insert the wrong diskette. If you need more diskettes (as you probably will), the CLI will let you label and use them as needed.

The DUMP command in the macros specifies a retention period of 0 days, allowing you to reuse the diskette set immediately. You may want to use a nonzero retention period for the macros. For example, you might use /RETAIN=7 for full backup and /RETAIN=2 for incremental backup. (If you ever want to reuse a diskette set before its expiration date, you will need to have the CLI relabel all diskettes.) If desired, you can use a text editor to change the /RETAIN= numbers.

Generally, it's best to use Operator without /LABEL whenever you can. With the OPERATOR command and /LABEL switch, the CLI will label any diskette without asking for confirmation. Labeling a diskette effectively destroys data stored on that diskette and all subsequent diskettes. So, it's very important to avoid mixing up diskette sets. Keep the sets separate, and make sure their paper labels are current and accurate.

The backup macros use the following volids for diskettes.

FULL_BACKUP	**INC_BACKUP**
FULL01	INC01
FULL02	INC02
.	.
.	.
.	.
FULLn	INCn

These are good, descriptive general-purpose volids. If you use them in the macros, you can restore using name FULL01 (to restore all files) or INC01 (to restore an incremental backup.). If you want different volids, use a text editor to change the name FULL01 or INC01 within the macro file. The CLI will then create volids based on your volid, and you can specify this volid — instead of FULL01 or INC01 — when you start the RESTORE macro.

The *filename* used by all the macros is BACKUP, and you should not change this name.

The incremental backup macro depends on a file named LAST_BACKUP — which is created by the last macro to run (full or incremental). If an error occurs and the backup is not valid, this file will contain an invalid last backup date. So, if ever a backup is invalid (for example, if you abort it and don't restart it that day), you should delete the file LAST_BACKUP. Then copy file LAST_BACKUP.BU to LAST_BACKUP; for example, type COPY LAST_BACKUP LAST_BACKUP.BU ↲.

Generally, you should standardize backup procedures. This means reusing volids, keeping each backup diskette set discrete, and having each diskette in the set labeled with its filename and volid/sequence number. Also, on the first diskette in a fileset, you should write the date it was last written to.

Users without Superuser privilege can use these macros to back up and restore their own directories. For this to work, the unit ACL must be something like +,WARE (perhaps set via the UP macro, as described above.)

```
[!equal,1,2]
comment This macro does a full labeled diskette backup from the
comment working directory. To do a full system backup, it requires
comment the process that runs it to be PID 2.
push
prompt pop
comment Check for arguments -- none is allowed.
[!nequal,%1-%,]
  write
  write This macro backs up copyable files in and below [!DIRECTORY] --
  write excluding DG-supplied files in the root and excluding directories
  write HELP,,PATCH,,and,,UTIL.
  write It doesn't allow arguments. Please try again via ,, %0% ,, when ready.
  write
[!else]
  string ok_to_proceed
  comment Check for the root directory -- if so the PID must be 002.
  class(1 2) ignore
  [!equal,[!directory],:]
    [!equal,[!pid],002]
      superuser on
    [!else]
      string non_master
    [!end]
  [!else]
    comment Not in the root - check for write access. If we can create
    comment and read from a file, assume we have needed access.
    permanence =?[!pid].[!username].tmp off
    delete =?[!pid].[!username].tmp
    write/l==?[!pid].[!username].tmp test
    string [=?[!pid].[!username].tmp]
    delete =?[!pid].[!username].tmp
    [!equal,[!string],test]
      string ok_to_proceed
    [!else]
      string no_access
    [!end]
  [!end]
  comment If it's not ok to proceed, describe error and stop.
  [!nequal,[!string],ok_to_proceed]
    [!equal,[!string],non_master]
      Write Only the master CLI can back up from the root.
    [!else]
      [!equal,[!string],no_access]
        write
        write Error - [!username] does not have write access to [!directory].
        write ,,,,,,,,,You cannot back up this directory.
        write
      [!end]
    [!end]
```

Figure 9-14. FULL_BACKUP Macro, for a Full Labeled Diskette Backup (continues)

```
    pop
  [!else]
    comment Check if CLI Operator mode is on. If not, stop with message.
    permanence =?[!pid].[!username].tmp off
    delete =?[!pid].[!username].tmp
    operator/l==?[!pid].[!username].tmp
    string [=?[!pid].[!username].tmp]
    delete =?[!pid].[!username].tmp
    [!nequal,([!string]),(OFF)]
      comment Operator mode is on -- proceed.
      class1 error
      class2 warning
      write
      write ** Full backup from directory [!DIRECTORY] at [!TIME] on [!DATE] **
      write
      write Please insert the first diskette to receive backup material in
      write the primary -- rightmost -- unit. The default first-volid for
      write full backups is FULL01. The CLI can label diskettes if needed.
      write The first diskette and any others used for backup will be
      write overwritten -- so don't use diskettes that have material you
      write want to keep.
      write
      write Please number the paper label of each diskette as it is filled so
      write that -- if needed -- the diskettes can be restored in correct order.
      write
      write ,,,,,,,,,,,,,,,,,,,,  -- Beginning file backup --
      [!equal,[!directory],:]
        comment Backup from root directory. Set diskette unit ACL
        comment for exclusive access, and then do the backup and reset ACL.
        acl   @dpm0   op, oware
        DUMP/RETAIN=0/V%0/L%    @LFD:FULL01:BACKUP    &
            #\HELP\NET\PATCH\SYSGEN\UTIL\SWAP\PAGE\PER\PROC\QUEUE&
\SWAP.SWAP\?+.BRK\+.ED\?+.JOB\+.LS\?+.TMP   NET:NETGEN:+
        comment  Note: If you want to back up files in HELP, SYSGEN, or UTIL,
        comment  insert the pathname(s) in the macro BEFORE the #\HELP...
        comment  exclusion. For example, the template UTIL:MYDATA+ before &
        comment in the DUMP command line will back up all :UTIL:MYDATA+ files.
        acl @dpm0 +,ware
      [!else]
        DUMP/RETAIN=0/V%0/L%    @LFD:FULL01:BACKUP  #
      [!end]
      write
      write ** Full backup of [!DIRECTORY] complete at [!TIME] **
      permanence/2=ignore  =LAST_BACKUP.BU  OFF
      delete/2=ignore  =LAST_BACKUP.BU
      rename/2=ignore  LAST_BACKUP  LAST_BACKUP.BU
      write/l==LAST_BACKUP   [!DATE]:[!TIME]

      write
      write This backup has created file LAST_BACKUP in this directory for
      write future backups. Don't delete this file.
    [!else]
      write Operator mode is not on. Type OPERATOR ON and retry the command.
    [!end]
  [!end]
[!end]
```

Figure 9-14. FULL_BACKUP Macro, for a Full Labeled Diskette Backup (concluded)

```
[!equal,1,2]
comment working directory -- based on the last backup, stored in file
comment LAST_BACKUP. To back up from the root directory, the macro
comment requires the process that runs it to be PID 2.
push
prompt pop
comment Check for arguments -- none is allowed.
[!nequal,%1-%,]
  write
  write This macro doesn't allow arguments. Please try again by typing
  write ,, %0% ,, when ready.
  write
[!else]
  string ok_to_proceed
  comment Check for the root directory -- if so the PID must be 002.
  class(1 2) ignore
  [!equal,[!directory],:]
    [!equal,[!pid],002]
      superuser on
    [!else]
      string non_master
    [!end]
  [!else]
    comment Not in the root - check for write access. If we can create
    comment a file, assume write access.
    permanence =?[!pid].[!username].tmp off
    delete =?[!pid].[!username].tmp
    write/l==?[!pid].[!username].tmp test
    string [=?[!pid].[!username].tmp]
    delete =?[!pid].[!username].tmp
    [!equal,[!string],test]
      string ok_to_proceed
    [!else]
      string no_access
    [!end]
  [!end]
  comment If it's not okay to proceed, describe error and stop.
  [!nequal,[!string],ok_to_proceed]
    [!equal,[!string],non_master]
      Write Error - only the master CLI can back up from the root.
    [!else]
      [!equal,[!string],no_access]
        write
        write Error - [!username] does not have write access to [!directory].
        write ,,,,,,,,,You cannot back up this directory.
        write
      [!end]
    [!end]
```

Figure 9-15. INC_BACKUP Macro, for an Incremental Labeled Diskette Backup (continues)

```
  pop
 [!else]
  [!equal,[!pathname =LAST_BACKUP],]
   Write
   Write Error - Incremental backup requires file LAST_BACKUP -- which does
   Write ,,,,,,,,, not exist. Suggest either doing FULL_BACKUP or retry
   Write ,,,,,,,,, incremental backup from directory in which you have
   Write ,,,,,,,,, done your most recent backup.
   Write
  [!else]
   comment Check to see if CLI Operator mode is on. If not, stop with message.
   permanence =?[!pid].[!username].tmp off
   delete =?[!pid].[!username].tmp
   operator/1==?[!pid].[!username].tmp
   string [=?[!pid].[!username].tmp]
   delete =?[!pid].[!username].tmp
   [!nequal,([!string]),(OFF)]
    comment Operator mode is on -- proceed.
    class1 error
    class2 warning
    write
    write ** Incremental backup from directory [!DIRECTORY]&
            at [!TIME] on [!DATE] **
    string [=LAST_BACKUP]
    write This backup will dump all files created or modified since [!string].
    write
    write Please insert the first diskette to receive backup material in
    write the primary -- rightmost -- unit. The default first-volid for
    write full backups is INC01. The CLI can label diskettes if needed.
    write The first diskette and any others used for backup will be
    write overwritten -- so don't use diskettes that have material you
    write want to keep.
    write
    write Please number the paper label of each diskette as it is filled so
    write that -- if needed -- the diskettes can be restored in correct order.
    write
    write ,,,,,,,,,,,,,,, -- Beginning file backup --
    [!equal,[!directory],:]
     comment Backup from root directory. Set the diskette unit ACL
     comment for exclusive access, and then do backup and reset ACL.
     acl   @dpm0   op, oware
     DUMP/RETAIN=0/AFTER/TLM=[!STRING]/V%0/L%   @LFD:INC01:BACKUP &
      #\NET\PAGE\PER\PROC\QUEUE\SWAP\SWAP.SWAP\SYSGEN\?+.BRK\+.ED&
\?+.JOB\+.LS\?+.TMP NET:NETGEN:+
     acl  @dpm0  +,ware
    [!else]
     DUMP/RETAIN=0/AFTER/TLM=[!STRING]/V%0/L%   @LFD:INC01:BACKUP  #
    [!end]
```

Figure 9-15. INC_BACKUP Macro, for an Incremental Labeled Diskette Backup (continues)

```
        write
        write ** Incremental backup of [!DIRECTORY] complete at [!TIME] **
        permanence/2=ignore  =LAST_BACKUP.BU  OFF
        delete/2=ignore  =LAST_BACKUP.BU
        rename/2=ignore  LAST_BACKUP  LAST_BACKUP.BU
        write/1==LAST_BACKUP   [!DATE]:[!TIME]

        write
        write This backup has created file LAST_BACKUP in this directory for
        write future backups. Don't delete this file.
      [!else]
        write Operator mode is not on. Type OPERATOR ON and retry the command.
      [!end]
    [!end]
  [!end]
[!end]
[!else]
  write This macro is nonexecutable. You can make it executable by
  write changing the 2 to a 1 in line 1 -- make both numbers 1.
  write
  write Backup is very important. Before using the edited macro routinely
  write for backup you should understand how it works. Please test this
  write macro. Use FULL_BACKUP and this macro to back up some files.
  write Then try restoring files using the RESTORE macro before relying
  write on this macro set for backup.
[!end]
```

Figure 9-15. INC_BACKUP. Macro, for an Incremental Labeled Diskette Backup (concluded)

```
[!equal,1,2]
comment This macro restores files from either a full or incremental
comment labeled diskette backup. To restore into the root directory,
comment the macro requires the process that runs it to be PID 2.
push
prompt pop
comment Check for at least one argument.
[!equal,%1%,]
  write
  write This macro requires at least one argument: the volid of the first
  write diskette in the fileset. The volid used by the FULL_BACKUP macro
  write is FULL01 and the first volid used by the INC_BACKUP macro is
  write INC01. After the volid argument, you can specify one or more
  write templates. If you omit templates, all files in the fileset will
  write be restored.
  write Run the macro via  ,, %0% ,, volid ,, {template} ...,, when ready.
  write
[!else]
  string ok_to_proceed
  comment Check for the root directory -- if so the PID must be 002.
  class(1 2) ignore
  [!equal,[!directory],:]
    [!equal,[!pid],002]
      superuser on
    [!else]
      string non_master
    [!end]
  [!else]
    comment Not in the root - check for write access. If we can create
    comment and read from a file, assume we have needed access.
    permanence =?[!pid].[!username].tmp off
    delete =?[!pid].[!username].tmp
    write/l==?[!pid].[!username].tmp test
    string [=?[!pid].[!username].tmp]
    delete =?[!pid].[!username].tmp
    [!equal,[!string],test]
      string ok_to_proceed
    [!else]
      string no_access
    [!end]
  [!end]
  comment If it's not ok to proceed, describe error and stop.
  [!nequal,[!string],ok_to_proceed]
    [!equal,[!string],non_master]
      Write Only the master CLI can restore the root directory.
    [!else]
      [!equal,[!string],no_access]
        write
        write Error - [!username] does not have write access to [!directory].
        write ,,,,,,,,,You need Superuser on to restore this directory.
        write
      [!end]
    [!end]
    pop
```

Figure 9-16. RESTORE Macro, to Restore Files from Labeled Diskettes (continues)

```
  [!else]
    comment Check if CLI Operator mode is on. If not, stop with message.
    permanence =?[!pid].[!username].tmp off
    delete =?[!pid].[!username].tmp
    operator/l==?[!pid].[!username].tmp
    string [=?[!pid].[!username].tmp]
    delete =?[!pid].[!username].tmp
    [!nequal,([!string]),(OFF)]
      comment Operator mode is on -- proceed.
      class1 error
      class2 warning
      write
      write ** Restoration within directory [!DIRECTORY] at [!TIME] on [!DATE] **
      write
      write Please insert the first diskette of the backup fileset in
      write the primary -- right -- unit.
      write
      write Later, you'll insert diskettes in the same order in which
      write these diskettes were originally dumped.
      write
      write ,,,,,,,,, -- Beginning file restoration --
      comment  Restore the fileset, using the supplied volid, filename
      comment  BACKUP, and any supplied template(s).
      [!equal,[!pid],002]
        acl @dpm0 op,oware
      [!end]
      LOAD/V/RECENT%0/L%     @LFD:%1%:BACKUP   &
           [!equal,%2%,]#[!else]%2-%[!end]
      write
      write ** Restoration of [!DIRECTORY] complete at [!TIME] **
      write
      [!equal,[!pid],002]
        acl @dpm0 +,ware
      [!end]
    [!else]
      write Operator mode is not on. Type OPERATOR ON and retry the command.
    [!end]
  [!end]
[!end]
[!else]
  write This macro is nonexecutable. You can make it executable by
  write changing the 2 to a 1 in line 1 -- make both numbers 1.
  write
  write Backup is very important. Before using the edited macro routinely
  write for backup you should understand how it works. Please test this
  write macro. Use FULL_BACKUP and INC_BACKUP to backup some files.
  write Then try restoring files using this macro before relying on this
  write macro set for backup.
[!end]
```

Figure 9-16. RESTORE.CLI Macro, to Restore Files from Labeled Diskettes (concluded)

Diskette Backup Example

For this example, assume that an AOS system is built in October. Information starts accumulating on it immediately. The person acting as system manager decides on a weekly full backup and daily incremental backups.

The full backup will occur each Friday (with provision for Friday holidays). Incremental backups will be done the following Monday through Thursday. All backups will be done from the root directory (:).

The first full backup, in October, takes only 14 diskettes; and the related incremental backups take only 2 diskettes each. The November full backups average 20 diskettes and incrementals take 2 or 3 diskettes each.

The next sections show the actions and dialog at the system console for the first Friday in December and for the first following incremental backup, on Monday.

All diskette sets have been labeled — with the default names shown in the sample macros (FULL01 or INC01), filename BACKUP.

The Full Backup

At the sample site, Friday afternoon arrives — time for the full backup.

1. Joan, the person who does backups, prepares to bring the multiuser environment down, making sure that no users will lose work when this happens (described in Chapter 6, section "Shutdown").

2. She returns to the master CLI, PID 2, on the system console

) WHO ↵
   PID: 13 OP CON0 :CLI.PR (This CLI is PID 13, not PID 2.)

) BYE ↵ (Log off this CLI.)
   AOS CLI TERMINATING ...

   You are now
   PID: 2 OP OP :CLI.PR (PID 2 — the master.)

3. She brings the multiuser environment down

) DOWN ↵

   .
   . (termination messages)...
   .

4. She makes the working directory the root:

) DIR : ↵

5. She turns operator on:

) OPERATOR ON ↵

   Since the diskettes have already been labeled, and the expiration date (default 0 days from the last backup) has been reached, Joan omits the /LABEL switch from the OPERATOR command.

6. She starts the full backup, including the /L switch to specify a listing file of filenames dumped. This is your decision, but we recommend it — the listing tells which files were backed up. (She ignores the access control issue, since the multiuser environment is down.)

) FULL_BACKUP/L=FILES_BACKED_UP ↵

   .

   .

   .

   ** *Full backup from directory : at 16:45:05 on 07-DEC-85* **

   PLEASE INSERT A DISKETTE IF NOT ALREADY INSERTED
   UNIT [@DPM0] VOLUME ID [FULL01] ? [Y]

7. She inserts the first FULL BACKUP diskette, FULL01, in the primary (right) unit.

8. She presses ↵.

   Beginning file backup

   . (It writes filenames backed up to the disk file ... diskette fills up.)
   . (Each diskette takes about 2 minutes to fill.)
   .

   PLEASE INSERT NEXT DISKETTE
   UNIT [@DPM0] VOLUME ID [FULLn] ? [Y]

9. She replaces the diskette with the next one and presses ↵.

10. She repeats steps 8 and 9 until she's out of previously labeled diskettes. Let's say there are 22 of these, and she has used them all. It displays

    PLEASE INSERT NEXT DISKETTE
    UNIT [@DPM0] VOLUME ID [FULL23] ? [Y]

11. She removes the diskette, inserts an unlabeled hardware-formatted diskette, and presses ↵.

    AN UNLABELED DISKETTE HAS BEEN INSERTED
    DO YOU WANT TO RELABEL THIS DISKETTE ? [N]

12. She types Y ↵.

    *** *RELABELED DISKETTE* *** *NEW VOLUME ID: FULL23*

13. She repeats steps 11 and 12 as long as the CLI prompts for another diskette. Let's say that only one additional diskette (FULL23) is needed. Some files are copied to the diskette. Then

    PLEASE REMOVE THE DISKETTE

    ** *Full backup of directory : complete at ..... on 07-DEC-85*

)

    The full backup is done. The listing file (shown here as FILES_BACKED_UP) remains in the root directory. You can print it (when the multiuser environment is up) and delete it as desired. Each backup listing should be stored in the same place as its diskette fileset. If you specify a listing file each time you do a full backup, simply delete the old one before starting the next next full backup.

14. Joan stores all FULL BACKUP diskettes in order, in their outer covers, and safely away from strong magnetic fields.

After a backup, AOS can be shut down or the multiuser environment can be brought back via UP ↵. In this case, Joan would probably shut down, turn everything off, and leave.

The Incremental Backup

At the sample site, Monday afternoon rolls around, time for the incremental backup.

1. As she did last week, Joan prepares to bring the multiuser environment down, making sure that no users will lose work.

2. She returns to the master CLI, PID 2, on the system console:

) WHO ↵
   PID: 13 OP CON0 :CLI.PR (PID 13, not PID 2.)

) BYE ↵ (Log off this CLI.)
   AOS CLI TERMINATING ...

   You are now
   PID: 2 OP OP :CLI.PR (PID 2 — the master.)

3. She brings it down.

) DOWN ↵

   .
   . (termination messages)
   .

4. She makes the working directory the root.

) DIR : ↵

5. She turns operator on

) OPERATOR ON ↵

   Again, since the diskette set has been labeled, and the retention period (0 days in the INC_BACKUP macro shown) has expired, Joan omits the /LABEL switch.

6. She starts the incremental backup, using the /L switch to specify another listing file of filenames copied. This can be *very* helpful when restoring incrementals — we recommend it.

) INC_BACKUP/L=FILES_BACKED_UP.DEC.10 ↵

   .
   .
   .

   ** *Incremental backup from directory : at .... on 10-DEC-85.*

   This backup will dump all files created or modified since 07-DEC-85....
   PLEASE INSERT A DISKETTE IF NOT ALREADY INSERTED
   UNIT [@DPM0] VOLUME ID [INC01] ? [Y]

7. She gets the first backup diskette, volid INC01, and inserts it in the primary (right) unit.

8. She presses ↵.

   Beginning file backup .
   . (It writes filenames backed up to the disk file.)
   .
   .
   PLEASE INSERT NEXT DISKETTE
   UNIT [@DPM0] VOLUME ID [INCn] ? [Y]

9. She removes the diskette, writes the date on its paper label (using a felt-tipped pen to avoid scoring the diskette surface). Then she replaces the diskette in its envelope and inserts the next diskette in the unit.

10. She repeats steps 8 and 9 as long as the CLI prompts for another diskette. Let's assume she's on the third diskette. Some files are dumped to the diskette. Then

    PLEASE REMOVE THE DISKETTE

    ** *Incremental backup of directory : complete at ... on 10-DEC-85*
)

    This incremental backup is done. Additional diskettes (as shown for the full backup) were not needed. The listing file (shown here as FILES_BACKED_UP.DEC.10) remains in the root directory. It can be printed (when the multiuser environment is up) and deleted as desired. AOS can be shut down or it can stay up.

11. Joan stores all INC BACKUP diskettes in order, in their outer covers, safely away from strong magnetic fields.

The next day, Tuesday, Joan does another incremental backup. The steps are the same as the last incremental backup, except that the date differs and she uses a different listing filename.

As you can see, the procedure is methodical — and repetitive. But it can be extremly important.

Restoring Dumped Files from Labeled Diskettes

Restoring falls into two categories: restoring one or more files, and restoring one or more LDUs. The first category is the most common, and the easiest and fastest.

Restoring One or More Files

Usually, you will restore files when someone has accidentally deleted a file (directory), or group of files. There are two things to consider when you restore files: the diskette set(s) needed, and the pathname template.

The diskette set(s) you use to restore depend on the date that the lost file(s) were last modified. If the files were created since the last backup, then they weren't backed up, and cannot be restored. Otherwise, use the backup (incremental or full) that occurred soonest after the files were modified.

If you can't determine when the files were last modified, check the backup listings. If the name of a lost file appears in any listing, then you know the file is in that backup. In the worst case, without a listing or dates, you must restore the last full backup set; then the earliest incremental backup; then next incremental backup; and so on. This is a good reason to keep your backup listings — especially for incremental backups.

After deciding on the diskette set, you must choose one or more pathname templates (unless you want to restore the entire backup). The RESTORE macro allows template arguments, in which you can specify a directory, a specific file, or a directory and pathname template.

You can restore all files in and below a directory (including subordinate directories), with the template

```
pathname-from-root:#
```

You can restore all files in and below a user's directory with the template

```
UDD:username:#
```

NOTE: Because the pathname starts in the root directory, it does not start with : — the name of the root directory.

For DG's Comprehensive Electronic Office (CEO) files, the directory structure and restoration procedure differs from that for standard AOS files. To restore CEO files, see *Managing the CEO® System*.

File Restoration Example

As an example, assume Andy accidentally deleted two files named REPORT.MAY and SUMMARY. He last modified them a week ago. (If lost files were last modified on different dates, it may be most efficient to work from the last full backup). You find and get the appropriate diskette set — which happens to be from an incremental backup.

You need a template for the restoration. The two filenames have the letters MA in common. So, you could use the template

UDD:ANDY:#:+MA+

This would work, but it might restore many matching, unwanted files. You can be more specific. The two names each have at least one character — and no period — following the A. So you could refine the template to

UDD:ANDY:#:+MA* –

Or, if you know which directory the files were in, you could use the specific pathnames.

pathname-from-root:REPORT.MAY   pathname-from-root:SUMMARY

Let's say you decide on the most general course — the +MA+ template, which covers many possibilities.

You go to the system console, get back to PID 2 (perhaps by typing BYE ↵, if needed.) Then

) RESTORE INC01 UDD:ANDY:#:+MA+ ↵	(Start macro, specifying an incremental backup and the desired template.)
Restoration with in directory : at 18:33 ... *Please insert the first diskette of the ...*	(Macro prompts for next steps...)
PLEASE INSERT A DISKETTE IF NOT ALREADY ... *UNIT [@DPM0] VOLUME ID [INC00] ? [Y]* ↵	(CLI prompts for diskette.) (Make sure diskette is inserted and press ↵.)
	(CLI runs through the diskette, looking for matching files. Since you specified no listing file, it will display matching pathnames that it restores on the screen.)
PLEASE INSERT NEXT DISKETTE. *UNIT [@DPM0] VOLUME ID [INC01] ? [Y]* ↵	(CLI prompts for next diskette; insert it and press ↵.)
	(Again, the CLI spins through the diskette, looking for files.)
UDD:ANDY:SUMMARY *UDD:ANDY:BDIR:SUMMARIES* *UDD:ANDY:BDIR:SUMMARIES.ED*	(It restores one of the files you want, and some unwanted files...)
PLEASE INSERT NEXT DISKETTE. *UNIT [@DPM0] VOLUME ID [INC02] ? [Y]* ↵	(CLI prompts for next diskette; insert it and press ↵.)
	(Time passes...
:UDD:ANDY:REPORTS:REPORT.MAY *:UDD:ANDY:REPORTS:REPORT.MAY.BU* *:UDD:ANDY:REPORTS:REPORT.MAY.ED*	(It restores the other lost file, and some others...)
PLEASE INSERT NEXT DISKETTE. *UNIT [@DPM0] VOLUME ID [INC02] ? [Y]*	(CLI prompts for next diskette.)

CTRL-C CTRL-A (Since all lost files were restored, you can interrupt the restore; you're done.)

ERROR
CONSOLE INTERRUPT

.

.

.

Restoration of : complete at 18:48:06
)

With the files restored, tell Andy to check them — and to check his directories for unwanted, restored files. He can then delete the unwanted files.

(If Andy had created the files in CEO (the full product, not Word Processor - Independent), and had accidentally deleted "documents" named REPORT.MAY and SUMMARY, you could suggest that he check the CEO Wastebasket. In the full CEO, documents are not actually deleted until a program called Janitor has been run. Until then, documents that people delete can be retrieved from the Wastebasket. If Andy's documents are not in the Wastebasket, you can restore them from the last backup — as described in *Managing the CEO® System.*)

Shortening a Restoration

When you mention a specific filename (like UDD:CHRIS:MYFILE) in a restoration, the restoration will end and the CLI prompt will return *as soon as the file has been copied to disk.*

The restoration will take longer if you use a template like UDD:CHRIS:MYF+ (as shown in the example) because the restore will continue through the last tape, even after the desired file has been copied. This happens because AOS can't tell, until it reaches the end of the last tape, that there is no matching filename in the tape set. When you see that the files of note have been restored, there's no need to restore others; you can interrupt the restore as shown above.

Generally, this means that it's desirable to give a specific filename, if you can. But if you do, be sure it's the correct one — if you make a mistake with a specific filename, the system will take you all the way through the tape set, and not restore the file you want restored. If you think this may be true, cancel the restoration with CTRL-C CTRL-A. Then start again.

Restoring an Entire LDU Using Diskettes — When and How

The time may come when you need to restore all backed-up material to an LDU. This can happen when a disk wears or fails in such a way that the system can no longer read it.

If you have checked with DG, and either acquired a new disk or decided to rebuild the old one, follow these steps.

1. To restore the master LDU, do the following. To restore a nonmaster LDU, skip to step 7.
2. Get the tailored system diskette you made after testing your tailored AOS system. (If you don't have a tailored system diskette, use the latest system diskette you received from DG; later, you'll need to generate a tailored system.)
3. Return to Chapter 2 and execute all the numbered steps there, using your tailored system diskette or the DG diskette. (If you're not using a tailored system diskette, generate a tailored AOS system, patch it, start it, and make a system diskette as described in Chapter 4.)
4. If you have any DG software products that were not backed up (those under :UTIL, for example, install these products as described in product documentation.
5. Use PREDITOR to create an operator profile; then create the line printer/batch queues and bring up EXEC (as shown in Chapter 5).

6. Get all your incremental backup sets and your last full backup set of diskettes.

7. To restore a nonmaster LDU: make sure this LDU is formatted with the Disk Formatter and has the desired name. Run FIXUP on it for for good measure. Then initialize it into your system as usual.

   If the nonmaster LDU needs DG software installed (for example, FORTRAN 77) and that software was not backed up, make the LDU your working directory, and install the software.

8. Log on to a user console that's physically close to the system console. Restore your most recently dumped incremental backup diskette set. Go backwards through the incrementals until you have restored them all. Then restore the most recent full backup.

   If you see a *CONTROL POINT DIRECTORY MAX SIZE EXCEEDED* message, the LDU is full; you, (or other users) must delete some previously deleted and backed up files on the disk, to free some space, before you can continue the restoration process.

   The files to delete may be in user directories, where they can be deleted with the CLI DELETE command, or in CEO, where they can be deleted with the "Delete" menu choice. In the full CEO, you will also need to run the Janitor to complete the deletions.

   When you have some disk space free, say 1000 to 2000 blocks or more (type SPACE :⤶), proceed with the next restoration.

9. Make sure you return all diskettes to their covers and store them safely.

10. You're done! You've recreated the entire LDU(s), having lost only a little work (the files created or changed since the last backup.)

If you have XODIAC networking software, the :NET:NETGEN files were backed up and have been restored; you can regenerate host and FMA files by running NETGEN and specifying the network spec filename.

NOTE: The RESTORE macro restores files with their original creation times, but it changes the time last modified to the time you restore the files. This means that your next backup must be a full backup, via the FULL_DUMP macro, since the dates last modified aren't real. It also means that the FILESTATUS command with the /BEFORE/TLM= switches won't help identify "old" files after the restoration.

Using PCOPY

PCOPY runs as a stand-alone program — directly bootable from disk. It is file PCOPY in the root directory.

PCOPY is ideal if you have two or more identical disk units, with removable disks, and LDUs of one physical disk. It is also ideal if you have twice as many disk units as there are disks in your largest LDU. And, it also works if you have three or more identical disk units and LDUs of two or more physical disks. If you PCOPY disk-to-disk, you can simply insert the copy LDU if something happens to the original's file structure (but don't do this if you suspect disk surface damage).

PCOPY can also copy to tape. It requires labeled tapes — which you can produce easily with the LABEL command or with PCOPY.

If You Make a Mistake

If you make a typing error and notice it before pressing ⤶, press the DEL key to erase the bad characters one by one. The system echoes an underscore (_) for each character erased. Then type the correct characters and press ⤶. Or, you can press CTRL-U to delete the entire line.

If you have already pressed ⤶, and PCOPY has not yet written to the destination medium, you can restart it by typing CTRL-C CTRL-A when PCOPY asks the next question. If PCOPY has started writing, you can stop it only via the break sequence and RESET ⤶ (for MV/8000s), the break sequence and P⤶ (for other programmed consoles), or by lifting and releasing STOP and RESET (data switches).

If PCOPY displays an error message that you can't understand, see the PCOPY error messages in Table 9-3, at the end of the PCOPY description. You can restart PCOPY if it aborts by typing CONTINUE ⤶ (for MV/8000s), and P⤶ (for other programmed consoles), or by pressing the CONTINUE switch (for data switches).

Before Running PCOPY

Make sure your destination medium is ready.

For a disk to disk copy, the LDUs must be identical. Each LDU's bitmap must be in the same place; each corresponding disk's remap area must be in the same place; and, if the source LDU was formatted as a system disk, the destination LDU must have an overlay area in the same place. The destination LDU's remap area must be free of bad blocks. (The LDU IDs and names can be the same, or different; PCOPY doesn't care about LDU IDs and names.)

All this sounds formidable, but PCOPY error messages will tell you what to do if the LDUs differ. Usually, it's simply a matter of running a Disk Formatter Partial format on one of the LDUs.

For a disk to tape copy, have enough tapes ready. Theoretically, a 2400-foot tape, on an MTB unit set at high density, can hold about 90,000 disk blocks (46 Mbytes) of information. An MTC unit, with a 1000-foot tape, can hold about 38,000 blocks (20 Mbytes). Type SPACE :⤶ to find the number of blocks used on the LDU; then divide by the pertinent number of blocks to see about how many tapes you need. The tapes may already be labeled (perhaps obsolete archive tapes that you no longer want for backup). PCOPY will accept the existing labels as you specify them. Or, PCOPY can label the tapes as you wish when it starts writing to them. (From AOS, you can always tell the contents of a tape label by typing TYPE @unitname⤶ with the tape mounted. The first four characters are the volume header; the following six characters are the volume ID.)

When ready, bootstrap PCOPY as in the following sections, from disk or tape.

Bootstrapping PCOPY from Disk

Shut down AOS normally (typing DOWN ⤶, BYE ⤶, etc.) Then bootstrap PCOPY. This is identical to bootstapping AOS, except that you type PCOPY ⤶ instead of a system name when the console asks *SYSTEM PATHNAME?* For example

...bootstrap sequence...

SPECIFY EACH DISK IN THE LDU
DISK UNIT NAME? DPF0 ⤶ (or whatever disk you wish)
DEVICE CODE? ⤶ (or device code if nondefault)

SYSTEM PATHNAME? PCOPY ⤶

Skip to the desired "PCOPY Dialog" section below.

Bootstrapping PCOPY from Tape

Booting PCOPY is easier from disk. But you might want to boot it from tape if you can't boot it from disk.

Before doing it, you need to make a stand-alone PCOPY tape. This is easy to do when AOS is running. Mount a 400-foot tape, ring in, on a tape unit (e.g., MTA0); put the unit on line; and type

```
) SUPERUSER ON ⤶
*) DIR :⤶
*) COPY/V @MTA0:0 TBOOT ⤶        (Use correct tape unit name)
TBOOT
*) COPY/V @MTA0:1 PCOPY ⤶        (Use correct tape unit name)
PCOPY
*) REWIND @MTA0 ⤶        (Use correct tape unit name)
*)
```

This gives you a stand-alone PCOPY tape. When this tape is mounted on unit 0 on the first tape controller, you can boot it via device code code 22 and specify the tape file number you want. You can do this with *any* stand-alone program, if you put TBOOT on file 0 of the tape.

Now, bring AOS down. Then, with the PCOPY tape on unit 0, bootstrap from tape (device code 22 octal, described in Chapter 2).

FROM MT-0: 1 ↵ (the PCOPY file)

Go to the desired "PCOPY dialog" section below. The sections proceed as follows

PCOPY Dialog, Disk-to-Disk, All Disks On-Line

Using PCOPY, Disk-to-Disk, All Disks Not On-Line

PCOPY Dialog, Disk-Tape

PCOPY Dialog, Tape-Disk

PCOPY Dialog, Disk-to-Disk, All Disks On-Line

Have all disks in your source and destination LDUs ready. For a multiple-disk LDU, you can put the source LDU disks in the first group of disk units, and the destination disks in the second group of disk units. For example, the source LDU might go in units DPF0 and DPF1, and the destination LDU in units DPF10 and DPF11. (If all disks in the source and destination LDUs will not be mounted throughout the copy, go to the next section.)

As a safeguard, you may want to write-disable the unit(s) that hold your source LDU before you start. This will prevent an accidental wrong answer from overwriting the source LDU.

Bootstrap PCOPY from disk or tape. The disk-to-disk dialog goes as follows. If PCOPY reports an error, see Table 9-2.

AOS DISK COPIER REV n
ENTER TODAY'S DATE (MM DD YY)?

1. Enter today's date, with numbers separated by one space; e.g.,

   8 4 85 ↵

   WILL THIS BE A DISK TO DISK PCOPY [N] ?

2. Type Y ↵.

   WILL ALL DISKS IN EACH LDU BE ON-LINE AT ALL TIMES [Y]

3. Press ↵ for the default (shown in brackets) or type Y ↵. (To PCOPY without all disks on line, go to the next section.)

   SPECIFY SOURCE LDU
   ENTER UNIT NAMES FOR EACH DISK IN LDU (<NL>) WHEN DONE
   DISK UNIT NAME?

4. Type the name of the first disk in the LDU; e.g., DPF0 ↵.

   DEVICE CODE?

   Unless you know that this disk is connected to a nondefault device code, press ↵ for the default. Otherwise, type the device code of the unit's controller and press ↵.

   DISK UNIT NAME?

PCOPY cycles the *DISK UNIT NAME?/DEVICE CODE?* questions until you respond ↵ to *DISK UNIT NAME?*. Be sure to specify all disks in the LDU, if a multiple-disk LDU was originally created with the Disk Formatter. When you've specified all disks, press ↵ at *DISK UNIT NAME?* PCOPY then describes the source LDU and asks for the destination LDU:

LDU UNIQUE ID = 'id'
LDU NAME = 'name'

SPECIFY DESTINATION LDU.
ENTER UNIT NAMES FOR EACH DISK IN LDU (<NL>) WHEN DONE
DISK UNIT NAME?

5. Type the unit name of the first disk in the *destination* LDU; e.g., DPF10 ↵.

   DEVICE CODE

6. If this disk is on the default device code for its name, press ↵; otherwise type the device code and ↵.

   DISK UNIT NAME?

   As with the source LDU, PCOPY cycles the unit and device code questions until you answer *DISK UNIT NAME?* with ↵. Then it displays the destination LDU ID and name and asks for confirmation:

   LDU UNIQUE ID = 'id'
   LDU NAME = 'name'

   COPY TO DISK FROM DISK. PLEASE CONFIRM (N/Y)?

7. If there are any files you want on the destination LDU, type N ↵ and begin again. A PCOPY overwrites all files on the destination medium. PCOPY asks this question to give you a chance to reconsider. To proceed, type

   Y ↵

   CONFIRMED.

   PCOPY now copies the whole disk structure from the source to the destination LDU. A full 190-Mbyte LDU takes roughly 15 minutes to copy. If PCOPY hits an error, it will display one of the error messages shown in Table 9-2. (But if either LDU is not ready, PCOPY will wait in silence until you make the LDU ready.) When PCOPY is done, it says

   DONE!

8. The PCOPY is done. If you write-disabled any disk units, you should write-enable them. Then, if the source disk includes an an AOS system, you might try booting the system on the destination disk. Give *its* name to the bootstrap program.

   To copy another LDU, place its disks in their units (if applicable); then type CONTINUE ↵ (for MV/8000s), C ↵ (for other programmed consoles), or press the CONTINUE switch (for data switches), and return to step 1 in this section.

Disk-to-Disk PCOPY Example, All Disks On-Line

SYSTEM PATHNAME? :PCOPY ↵

AOS DISK COPIER REV n

ENTER TODAY'S DATE (MM DD YY)? 8 4 85 ↵

WILL THIS BE A DISK TO DISK PCOPY [N] ? Y ↵
WILL ALL DISKS IN EACH LDU BE ON-LINE AT ALL TIMES [Y] ↵

```
SPECIFY SOURCE LDU
ENTER UNIT NAMES FOR EACH DISK IN LDU (<NL>) WHEN DONE
DISK UNIT NAME?     DPF0 ↵
DEVICE CODE?     ↵
DISK UNIT NAME?     ↵

LDU UNIQUE ID = 'ROOT1'
LDU NAME = 'ROOT1'

SPECIFY DESTINATION LDU.
ENTER UNIT NAMES FOR EACH DISK IN LDU (<NL>) WHEN DONE
DISK UNIT NAME?     DPF10 ↵
DEVICE CODE     ↵
DISK UNIT NAME?     ↵

LDU UNIQUE ID = 'ROOT1A'
LDU NAME = 'ROOT1A'

COPY TO DISK FROM DISK. PLEASE CONFIRM (N/Y)?     Y ↵
CONFIRMED...
(time passes)

DONE!
```

Using PCOPY, Disk-to-Disk, All Disks Not On-Line

Do this only if you want to PCOPY a multiple-disk LDU, have units with removable packs, and you don't have twice as many identical disk units as there are disks in the LDU.

To PCOPY disk-disk without all disks in each LDU on-line, you need three (or more) identical disk units with removable packs. The disk with the LDU's bitmap must be in the first disk unit, and will stay in this unit throughout the LDU copy. (The bitmap disk is usually the first disk in the LDU, since the Disk Formatter puts the bitmap on this disk by default.) The sequence (with three disk units) goes like this:

- The bitmap disk is in the first unit (unit 0). You might want to write-disable this unit so that you won't accidentally overwrite it. Unit 1 holds the copy disk for the bitmap disk. Unit 2 (the third unit) doesn't matter for this step.
- PCOPY asks about source and destination units, and you specify unit 0 as the source and unit 1 as the destination. PCOPY copies the bitmap disk to the disk in unit 1, then prompts you to dismount unit 1.
- The bitmap disk stays in unit 0. Remove the disk from unit 1 and put the second source disk in unit 1. Write-disable unit 1 so that, if you accidentally specify unit 1 as the destination, the disk in unit 1 won't be overwritten. Put the second destination disk in unit 2 (the third unit). PCOPY copies the second source disk (unit 1) to the second destination disk (unit 2).
- If you're copying a two-disk LDU, you're done; remove the second disk copy from unit 2. The source LDU remains in units 0 and 1. Write-enable all write-disabled units.

If there are more disks in the LDU, remove the source disk from unit 1 and the copy disk from unit 2. Insert the next source disk in unit 1 and the next destination disk in unit 2. PCOPY copies the source in unit 1 to the destination in unit 2.

Repeat this step until you have copied all the disks in the source LDU. Then remove the last copy disk from unit 2. The bitmap source disk remains in unit 0; the last source disk remains in unit 1. Put the desired disks in units 1 and 2 to resume system operations. Write-enable all write-disabled disks.

(If you have four or more identical disk units with removable packs, you'll be able to think up other disk/unit placements for the PCOPY. This section just gives the *approach*. The only restriction is that the bitmap disk must remain on-line and ready at all times.)

PCOPY Dialog, Disk-to-Disk, All Disks Not On-Line

Have the LDU source and destination disk ready as described above.

Bootstrap PCOPY from disk or tape. The disk-to-disk dialog goes something like the following. If PCOPY reports an error, see Table 9-2.

AOS DISK COPIER REV n

ENTER TODAY'S DATE (MM DD YY)?

1. Enter today's date, separating numbers by one space; e.g.,

   8 4 85 ↵

   WILL THIS BE A DISK TO DISK PCOPY [N] ?

2. Type Y ↵

   WILL ALL DISKS IN EACH LDU BE ON-LINE AT ALL TIMES [Y]

3. Type N ↵. (To PCOPY with all disks on-line, see the previous section.)

   SPECIFY SOURCE LDU
   ENTER FIRST PHYSICAL DISK NAME (MUST CONTAIN BITMAP):

4. Respond with the unit name that holds the first (bitmap) disk in the LDU; e.g., DPF0 ↵.

   DEVICE CODE?

   Unless you know that this disk is connected to a nondefault device code, press ↵ for the default. Otherwise, type the device code of the unit's controller. PCOPY gives the source LDU ID and name, then asks about the second LDU:

   LDU UNIQUE ID = 'id'
   LDU NAME = 'name'

   SPECIFY DESTINATION LDU.
   ENTER FIRST FIRST PHYSICAL DISK NAME:

5. Type the unit name of the first disk in the *destination* LDU; e.g., DPF1 ↵.

   DEVICE CODE

6. If this disk is on the default device code for its name, press ↵ otherwise type the device code. PCOPY displays the destination LDU ID and name, then asks for confirmation:

   LDU UNIQUE ID = 'id'
   LDU NAME = 'name'

   COPY TO DISK FROM DISK. PLEASE CONFIRM (N/Y)?

7. If there are any files you don't want overwritten on the destination disk, type N ↵ and begin again. A PCOPY overwrites all files on the destination medium. PCOPY asks this question to give you a chance to reconsider. To proceed, type

   Y ↵

   CONFIRMED.

   PCOPY now copies the bitmap disk to the destination disk. A full 190-megabyte disk takes roughly 15 minutes to copy. If PCOPY hits an error, it will display one of the error messages explained in Table 9-2. (But if either disk is not ready, PCOPY will wait in silence until you make it ready.) When PCOPY is done, it says

   PROCESSING COMPLETED ON CURRENT DISKS.
   DISMOUNT DESTINATION DISK #1

   ENTER SOURCE DISK #2 NAME:

8. Dismount the destination disk (the one in the second unit, unit 1, if you followed the procedures given above). Mount the next source disk and the next destination disk. (Leave the bitmap disk alone.) Then type the source disk name; for example, DPF1 ⤶.

   DEVICE CODE?

9. As above, press ⤶ for the default code; or for a nonstandard code, type the code and ⤶.

   ENTER DESTINATION DISK #2 NAME:

10. Type the name of the second destination disk; e.g., DPF2 ⤶.

    DEVICE CODE?

    Press ⤶ for the default or type the device code.

11. PCOPY now copies the new source disk to the new destination disk. If the new source disk is the last disk in the LDU, the LDU copy is done and PCOPY says *DONE!*. Go to step 17. (PCOPY knows the number of disks in the LDU from the Disk Information Block (DIB) in the first disk.)

12. If the source LDU contains more disks, PCOPY prompts you to dismount the disks and asks for the new source and destination names, as follows:

    PROCESSING COMPLETED ON CURRENT DISKS.
    DISMOUNT SOURCE DISK #n
    DISMOUNT DESTINATION DISK #n

    ENTER SOURCE DISK #n+1 NAME:

13. Dismount the last source and destination disks (leaving the bitmap disk alone). Put the next source and destination disks in available units (unit 1 and 2, if you're following the procedures given above.) Then type the unit name of the next source disk; e.g., DPF1 ⤶.

    DEVICE CODE?

14. Press ⤶ for the default code, or type the code.

    ENTER DESTINATION DISK #n+1 NAME:

15. Type the unit name of the next destination disk name; e.g., DPF2 ⤶.

    DEVICE CODE?

16. Press ⤶ for the default code, or type the code. PCOPY proceeds to copy the source disk to the destination disk. If this is the last source disk in the LDU, PCOPY says *DONE!*; proceed to step 17. If there are more disks in the source LDU, return to step 12.

17. The PCOPY is done. You can dismount the last copy disk and mount the disks needed for normal AOS operations. Be sure to write-enable all disk units.

    To copy another LDU, ready the source and destination disk(s). Then type CONTINUE ⤶ (for MV/8000s), C ⤶ (for other programmed consoles), or press the CONTINUE switch (for data switches), and return to step 1 in this section.

Disk-to-Disk PCOPY Example, All Disks Not On-Line

This example shows a PCOPY of a two-disk LDU.

SYSTEM PATHNAME? :PCOPY

AOS DISK COPIER REV n

ENTER TODAY'S DATE (MM DD YY)? 8 4 85 ⤶

WILL THIS BE A DISK TO DISK PCOPY [N] ? Y ⤶
WILL ALL DISKS IN EACH LDU BE ON-LINE AT ALL TIMES [Y] N ⤶

SPECIFY SOURCE LDU
ENTER FIRST PHYSICAL DISK NAME (MUST CONTAIN BITMAP): DPF0 ↵
DEVICE CODE? ↵

LDU UNIQUE ID = 'SYS'
LDU NAME = 'SYS'

SPECIFY DESTINATION LDU.
ENTER FIRST FIRST PHYSICAL DISK NAME: DPF1 ↵
DEVICE CODE ↵

LDU UNIQUE ID = 'SYS1'
LDU NAME = 'SYS1'

COPY TO DISK FROM DISK. PLEASE CONFIRM (N/Y)? Y ↵
CONFIRMED.

.
. (time passes)
.

PROCESSING COMPLETED ON CURRENT DISKS.
DISMOUNT DESTINATION DISK #1

(Dismount the destination disk; mount the new source and destination disks.)

ENTER SOURCE DISK #2 NAME: DPF1 ↵
DEVICE CODE? ↵
ENTER DESTINATION DISK #2 NAME: DPF10 ↵
DEVICE CODE? ↵

.
. (time passes)
.

DONE!

PCOPY Dialog, Disk-to-Tape

Have all the tape units set to DENSITY HIGH (if this applies), to conserve tape. And have all the tapes that you can mount mounted and on-line. Having multiple tapes mounted will reduce the number of tape changes you must make. Next, bootstrap PCOPY from disk, tape, or diskette as described above.

AOS DISK COPIER REV n

ENTER TODAY'S DATE (MM DD YY)?

1. Enter today's date, separating numbers by one space; e.g.,

   8 4 85 ↵

   WILL THIS BE A DISK TO DISK PCOPY [N]?

2. Press ↵ for the default answer, No, shown in square brackets.

   IS SOURCE A LOGICAL DISK (D), LABELED TAPE (T), OR FLOPPY (F)?

3. Type D ↵

   SPECIFY SOURCE LDU
   ENTER UNIT NAMES FOR EACH DISK IN LDU (<NL>) WHEN DONE
   DISK UNIT NAME?

4. Type the name of the first disk in the LDU; e.g., DPF0 ↵.

   DEVICE CODE?

5. Unless you know that this disk is connected to a nonstandard device code, press ↵ for the default. Otherwise, type the device code of the unit's controller.

   DISK UNIT NAME?

   PCOPY cycles *DISK UNIT NAME?/DEVICE CODE?* questions until you respond ↵ to *DISK UNIT NAME?*. Be sure to specify all disks in the LDU, if a multiple-disk LDU was originally created with the Disk Formatter.

   When you've specified each disk, press ↵ at *DISK UNIT NAME?;* PCOPY then describes the source LDU and asks for the destination medium:

   LDU UNIQUE ID = 'id'
   LDU NAME = 'name'

   COPY TO DISK (D), LABELED TAPE (T), OR FLOPPY (F)?

   Type T ↵.

6. *TAPE UNIT NAME?*

   Type the name of the tape unit that holds the first tape (or next tape); e.g., MTA0 ↵.

7. *SPECIFY VOLUME ID*

   Type the volume ID (volid) that you want on the tape. If you tell PCOPY to label the tapes, it will write this volid to the tape. If you tell PCOPY not to label the tapes, you should type the correct volid here. (Each tape volid should be written somewhere on a paper label on the tape reel or cover.) Tape volids are one to six characters long and can be written to tape via either the LABEL program or PCOPY. For example, if you want the volid PC0000, type PC0000 ↵.

   TAPE UNIT NAME?

8. PCOPY cycles *TAPE UNIT NAME?/VOLUME ID?* questions until you respond ↵ to *TAPE UNIT NAME?*.

   For example, if the second tape is mounted on MTA1 and you want its volid to be PC0001 (or you know that its volid *is* PC0001), you'd type MTA1 ↵, then PC0001 ↵.

   When you've specified all the tape units and volids mounted, press ↵ to *TAPE UNIT NAME?*.

   EXPIRATION DATE (DEFAULT is mm dd yy)

9. The default expiration date is three months from the current date. If anyone tries to PCOPY to a tape whose expiration date has not been reached, PCOPY will ask for confirmation before proceeding. Thus, the expiration date can be a useful tool — a time limit that may be overridden. You can select the three-month default by pressing ↵. Or you can type the date after which you want the tape to be reused; e.g., 9 14 85 ↵. Or, you can type 0 ↵, which labels the tape as already expired, so that it can be reused at any time. If you enter a date (not 0), it must be at least one day later than today's date. When you answer the question, PCOPY says

   SHOULD PCOPY LABEL THE TAPES? [Y]

10. You can have PCOPY label all tapes that will be used to copy the LDU. Before it writes to each tape, it will ask for the volume ID (volid), then write the volid you specify to the tape. To have PCOPY label the tapes, answer Yes by pressing ↵ or typing Y ↵. The PCOPY labels will overwrite current labels (if any). You will need to specify the tape volids if you ever use PCOPY to restore the LDU from these tapes, so keep track of them — perhaps by writing each volid on a paper label attached to the tape reel.

    Having PCOPY label the tapes is a convenience, so you will probably want to answer Yes.

If you want to retain the current tape labels, type N ↵. If you type N ↵, the tapes must have valid labels, with volids that match the volids you specify in *SPECIFY VOLUME ID?* questions. If a label or volid is invalid, PCOPY will ask you to respecify the tape unit and label, and ask again if you want it to label the tapes.

COPY TO DISK FROM TAPE. PLEASE CONFIRM (N/Y)?

11. PCOPY gives you this chance to confirm the tape write. To confirm, type

    Y ↵

    CONFIRMED.

    GENERATION NUMBER? [current]

12. The generation number is part of the tape label and distinguishes successive copies on the tape. It is an indicator of tape "mileage." It is optional; you need not enter it or use it. If you want to use it, and the tape is new, type 0001 ↵. If the tape isn't new, increment the old number by 1; for example, if the number is 0123, type 0124 ↵. To skip the generation number, press ↵; this will retain the current number (or none, if there is none).

    PCOPY now starts copying the LDU to labeled tape. A 2400-foot tape takes 6 to 9 minutes; an 800-foot tape on an MTC unit takes 5 to 6 minutes. If PCOPY hits an error, it will display one of the error messages shown in Table 9-2. (But if a disk or tape unit is not ready, PCOPY will wait in silence until you make everything ready.)

13. After PCOPY fills each tape, it rewinds the tape, then writes to the next tape (if mounted). If PCOPY needs more tape volumes, it will return to the *TAPE UNIT NAME?,* in step 6 above. You'll mount the next tape volume(s), specify the unit names(s) and volume ID(s), and proceed. For the second and subsequent tapes, PCOPY skips the *SHOULD TAPES BE LABELED* and *GENERATION NUMBER* questions.

    When PCOPY has copied the entire LDU to tape, it says

    PCOPY TO TAPE COMPLETE. DO YOU WANT TO VERIFY THE TAPE? [N]

14. PCOPY can verify the tapes that it wrote by reading them back. If all tapes read back without errors, you can be nearly certain that they will PCOPY properly later. Verification doesn't take too long, and, generally, it's a good idea to verify.

    When you verify, PCOPY must read the tapes in the same order that it wrote them. So, for a multiple-tape dump, you will need to dismount the last tape(s) and mount the first tape(s). PCOPY then reads the label and tape and prompts for others as needed. When done, it will confirm the verification and proceed to step 15. (If PCOPY hits a fatal read error, replace the bad tape with another, restart PCOPY if needed and redo the PCOPY dump from the beginning, at step 6.)

    If you decide to verify the tapes, type Y ↵ and go to step 6.

    If you don't want to verify the tapes, press ↵ or type N ↵.

15. PCOPY is done. It says

    DONE!

    Remove the write-enable rings from the tapes, make sure they have paper labels with the correct volume IDs, and store them in a safe place.

    To copy another LDU, place its disks in their units (if applicable); then type CONTINUE ↵ and return to step 1 in this section.

Disk-to-Tape PCOPY Example

```
SYSTEM PATHNAME?    :PCOPY ↵

AOS DISK COPIER REV n

ENTER TODAY'S DATE (MM DD YY)?    8  4  85 ↵
WILL THIS BE A DISK TO DISK PCOPY [N] ?    ↵
IS SOURCE A LOGICAL DISK (D), LABELED TAPE (T), OR FLOPPY (F)?    D ↵

SPECIFY SOURCE LDU
ENTER UNIT NAMES FOR EACH DISK IN LDU (<NL>) WHEN DONE
DISK UNIT NAME?    DPF0 ↵
DEVICE CODE?    ↵
DISK UNIT NAME?    ↵

LDU UNIQUE ID = 'UDD1'
LDU NAME = 'UDD1'

COPY TO DISK (D), LABELED TAPE (T), OR FLOPPY (F)?    T ↵
TAPE UNIT NAME?    MTA0 ↵
SPECIFY VOLUME ID    UDD1X1 ↵
TAPE UNIT NAME?    MTA1 ↵
SPECIFY VOLUME ID    UDD1X2 ↵
TAPE UNIT NAME?    ↵
EXPIRATION DATE (DEFAULT IS 11 12 85)    ↵
SHOULD PCOPY LABEL THE TAPES [Y]?    ↵

COPY TO TAPE FROM DISK. PLEASE CONFIRM (N/Y)?    Y ↵
CONFIRMED.

GENERATION NUMBER? [0002]    0003 ↵

.
.  (time passes as PCOPY writes to first tape, then second tape)
.
PCOPY TO TAPE COMPLETE. DO YOU WANT TO VERIFY THE TAPE? [N]    Y ↵

TAPE UNIT NAME?    MTA0 ↵
SPECIFY VOLUME ID    UDD1X1 ↵
TAPE UNIT NAME?    MTA1 ↵
SPECIFY VOLUME ID    UDD1X2 ↵
TAPE UNIT NAME?    ↵

.
.  (time passes as PCOPY reads first tape, then second tape)
.
TAPE VERIFICATION COMPLETE.

DONE!
```

PCOPY Dialog, Tape-to-Disk

As with disk-tape, have all the labeled tapes that you can mount mounted and on line. Having multiple tapes mounted will reduce the number of tape changes you must make.

Next, bootstrap PCOPY from disk or tape.

```
AOS DISK COPIER REV n

ENTER TODAY'S DATE (MM DD YY)?
```

1. Enter today's date, separating numbers by one space; e.g.,

   8 4 85 ↓

   WILL THIS BE A DISK TO DISK PCOPY [N]?

2. Press ↓ to select the default answer, No, shown in square brackets.

   IS SOURCE A LOGICAL DISK(D), LABELED TAPE (T), OR FLOPPY (F)?

3. Type T ↓.

   TAPE UNIT NAME?

4. Type the name of the tape unit that holds the *first* tape; e.g., MTA0 ↓.

   SPECIFY VOLUME ID

5. Type the tape volume ID (volid), as assigned by PCOPY or with the LABEL utility. (If you can't determine the volid, make one up; later, when PCOPY reads the label, the error message will tell you the actual volid. But remember that, to restore the LDU properly, the tapes must be copied to the LDU in the same order that they were copied *from* the original LDU.)

   For example, if the volid were UDD1X1, you'd type UDD1X1 ↓.

   TAPE UNIT NAME?

6. PCOPY cycles the *TAPE UNITNAME?/VOLUME ID?* questions until you respond ↓ to *TAPE UNITNAME?*.

   For example, if the second tape is mounted on MTA1 and is labeled UDD1X2, you'd type MTA1 ↓; then UDD1X2 ↓.

   When you've specified each unit and volid mounted, press ↓ to *TAPE UNIT NAME?*.

   SPECIFY DESTINATION LDU
   ENTER UNIT NAMES FOR EACH DISK IN LDU (<NL>) WHEN DONE
   DISK UNIT NAME?

7. Type the unit name of the first disk in the LDU; e.g., DPF0 ↓.

   DEVICE CODE?

8. Unless you know that this disk is connected to a nondefault device code, press ↓ for the default. Otherwise, type the device code of the unit's controller.

   DISK UNIT NAME?

9. PCOPY cycles *DISK UNIT NAME?/DEVICE CODE?* questions until you respond ↓ to *DISK UNIT NAME?*. Be sure to specify all disks in the LDU, if a multiple-disk LDU was originally created with the Disk Formatter.

   When you've specified each disk in the LDU, press ↓ at *DISK UNIT NAME?*. PCOPY displays the LDU unique ID and name, then asks for confirmation.

   LDU UNIQUE ID = 'id'
   LDU NAME = 'name'

   COPY FROM TAPE TO DISK. PLEASE CONFIRM (N,Y)?

10. If there are any files you want on the destination LDU, type N ⤶ and begin again. A PCOPY overwrites all files on the destination medium — and PCOPY asks this question to give you a chance to reconsider. To proceed, type

    Y ⤶

    CONFIRMED.

    PCOPY now starts copying the tape(s) to the LDU. A 2400-foot tape takes 7 to 10 minutes. If PCOPY hits an error, it will display one of the error messages explained in Table 9-2. (But if an LDU or tape unit is not ready, PCOPY will wait in silence until you make it ready.)

    As PCOPY reaches the end of each tape, it will rewind the tape. Then, if it needs more tape volumes, it will return to *TAPE UNIT NAME?*, as shown in step 3 above.

    When PCOPY is done, it says

    DONE!

11. Now, if you restored an AOS system LDU, you might try booting the system on the destination LDU. Type the copy LDU disk unit name(s), then device code(s) as SYSBOOT asks *DISK UNIT NAME?* and *DEVICE CODE?*.

    To restore another LDU, place its disks in their units (if applicable); then type CONTINUE⤶ (MV/8000), C⤶ (other programmed consoles), or press the CONTINUE switch (data switches) and return to step 1 in this section.

Tape-to-Disk PCOPY Example

SYSTEM PATHNAME? :PCOPY ⤶

AOS DISK COPIER REV n

ENTER TODAY'S DATE (MM DD YY)? 8 4 85 ⤶

WILL THIS BE A DISK TO DISK PCOPY [N]? ⤶
IS SOURCE A LOGICAL DISK (D), LABELED TAPE (T), OR FLOPPY? T ⤶
TAPE UNIT NAME? MTA0 ⤶
SPECIFY VOLUME ID UDD1X1 ⤶ (type the tape's volid)
TAPE UNIT NAME? MTA1 ⤶
SPECIFY VOLUME ID UDD1X2 ⤶ (tape's volid)

SPECIFY DESTINATION LDU
ENTER UNIT NAMES FOR EACH DISK IN LDU (<NL>) WHEN DONE
DISK UNIT NAME? DPF0 ⤶
DEVICE CODE? ⤶
DISK UNIT NAME? ⤶

LDU UNIQUE ID = 'UDD1'
LDU NAME = 'UDD1'

COPY TO DISK FROM TAPE. PLEASE CONFIRM (N/Y)? Y ⤶
CONFIRMED.

.
. (time passes as PCOPY copies to the LDU)
.

TAPE UNIT NAME? MTA0 ↵
SPECIFY VOLUME ID UDD1X3 ↵ (tape's volid)
TAPE UNIT NAME? MTA1 ↵
SPECIFY VOLUME ID UDD1X4 ↵ (tape's volid)

.
. (time passes as PCOPY copies to the LDU)
.

DONE!

PCOPY Dialog, Disk-to-Diskette

Before you start, you must have enough write-enabled diskettes to hold the LDU's material. These diskettes need not be formatted with the Disk Formatter, but running a Formatter full format on each one identifies bad blocks, so that PCOPY can avoid them. If you do run the Formatter on the diskettes, answer N↵ to the *SYSTEM DISK* question.

PCOPYing disk to diskette will require varying numbers of diskettes depending on the size of the disk and the capacity of the diskettes. We suggest that you number the diskettes by writing on their envelopes or labels; for example, 1, 2, 3, 4, etc. If you ever need to restore an LDU from these diskettes, PCOPY will require the diskettes in the original dump sequence.

When you're ready, bootstrap PCOPY from disk, tape, or diskette.

AOS DISK COPIER REV n

ENTER TODAY'S DATE (MM DD YY)?

1. Enter today's date, separating numbers by one space; e.g.,

   8 4 85 ↵

   WILL THIS BE A DISK TO DISK PCOPY [N]?

2. Press ↵ for the default answer, No, shown in square brackets.

   IS SOURCE A LOGICAL DISK (D), LABELED TAPE (T), OR FLOPPY(F)?

3. Type D ↵.

   SPECIFY SOURCE LDU
   ENTER UNIT NAMES FOR EACH DISK IN LDU (<NL>) WHEN DONE
   DISK UNIT NAME?

4. Type the name of the first disk in the LDU; e.g., DPI0 ↵.

   DEVICE CODE?

5. Unless you know that this disk is connected to a nonstandard device code, press ↵ for the default. Otherwise, type the device code of the unit's controller.

   DISK UNIT NAME?

6. PCOPY cycles *DISK UNIT NAME?/DEVICE CODE?* questions until you respond NEW LINE to *DISK UNIT NAME?*. Be sure to specify all disks in the LDU, if a multiple-disk LDU was originally created with the Disk Formatter.

   When you've specified each disk, press ↵ at *DISK UNIT NAME?* PCOPY then describes the LDU's ID and name, and asks for the destination medium:

   COPY TO DISK (D), LABELED TAPE (T) OR FLOPPY (F)?

7. Type F ↓

FLOPPY UNIT NAME? [DPI1]

Type the name of the diskette unit that holds the diskette (e.g., DPM0); or press ↓ if the diskette is in the default unit, DPI1.

DEVICE CODE?

8. If the diskette is connected to the default device code (20 for DPM0), press ↓. Otherwise, type the device code.

SPECIFY VOLUME ID

9. Type the name of the label that you want on the first diskette. This label will identify the entire diskette dump. It must be between one and six characters long. You can use the date, or any other name that will help identify the dump. For example, for 5 May 1985, you could type 051285 ↓.

The label name will be required to restore the LDU, it should be written on the envelope or label of diskette number 1.

COPY TO FLOPPY FROM DISK. PLEASE CONFIRM (N/Y)?

10. PCOPY gives you this chance to confirm the diskette write. To confirm, type

Y ↓

CONFIRMED.

PCOPY now starts copying the LDU to diskette. Each diskette takes 1-2 minutes. If PCOPY hits an error, it will display one of the error messages shown in Appendix C, Table C-7. (But if the disk or diskette unit is not ready, PCOPY will wait in silence until you make it ready.)

11. After PCOPY has filled the diskette, it says

MOUNT NEXT FLOPPY, STRIKE ANY KEY WHEN READY.

Remove the diskette from its unit, replace it in the envelope, and write the diskette sequence number on the diskette envelope if you have not done so (use a felt-tipped pen). Insert the next diskette in the unit.

12. Strike any key.

PCOPY now copies from the LDU to the new diskette. When done, it displays the message shown in step 11. Repeat steps 11 and 12 until PCOPY says:

DONE!

Remove the last diskette from its unit, and store all diskettes safely, away from strong magnetic fields.

To copy another LDU, place its disks in their units (if applicable); then press the CONTINUE panel switch (or, with an MV/8000, type CONTINUE ↓; with other programmed front panels, type P ↓), make sure you have enough write-enabled diskettes, and return to step 1 in this section.

Making a PCOPY Diskette – If you don't have a tape unit, you should make a PCOPY diskette so that later — if you cannot boot from disk — you can bring up PCOPY from diskette. You can make a PCOPY diskette from AOS easily, using the stand-among Disk Formatter and Installer. Put a new diskette, with write hole covered, in a diskette unit (say DPI1). Execute the Disk Formatter (XEQ :UTIL:DFMTR↓), and specify a full format and the diskette unit name. Use PCOPY (or any valid name) as the LDU ID and name. Default other Disk Formatter questions, but specify a zero length for the overlay area, and be sure to run at least one pattern on the diskette. By defaulting the SYSTEM DISK question, you reserve space for a system bootstrap and system.

When the Disk Formatter is done, execute the stand-among Installer. Install a system bootstrap and PCOPY as in the following example:

*) XEQ :UTIL:INSTL ↵

AOS INSTALLER REV xx.xx

ENTER ALL UNITS IN LDU
DISK UNIT NAME? DPI1 ↵ (Correct diskette unit name)
DISK UNIT NAME? ↵

DISK BOOTSTRAP INSTALLED
INSTALL A SYSTEM BOOTSTRAP? Y ↵
PATHNAME? :SYSBOOT ↵

SYSTEM BOOTSTRAP INSTALLED
INSTALL A SYSTEM? Y ↵
PATHNAME? :PCOPY ↵

SYSTEM INSTALLED
—DONE!

*)

Remove the diskette, label it "PCOPY Diskette" or equivalent, and store it safely, with your other PCOPY diskettes.

If you ever need to bootstrap PCOPY from diskette, you will put the PCOPY diskette in unit 0 (use the toggle switch or thumbwheel to make the diskette unit 0), then bootstrap as you would AOS. When SYSBOOT asks *SYSTEM PATHNAME,* press NEW LINE to select the installed "system." PCOPY will then run from the diskette. Remove the PCOPY diskette from the unit, put dump diskette number 1 in the unit, and proceed as described in the next section.

Disk-to-Diskette PCOPY Example –

SYSTEM PATHNAME? :PCOPY ↵

AOS DISK COPIER REV n

ENTER TODAY'S DATE (MM DD YY)? 8 4 85 ↵
WILL THIS BE A DISK TO DISK PCOPY [N] ? ↵
IS SOURCE A LOGICAL DISK (D) LABELED TAPE (T), OR FLOPPY(F)? D ↵

SPECIFY SOURCE LDU
ENTER UNIT NAMES FOR EACH DISK IN LDU (<NL>) WHEN DONE
DISK UNIT NAME? DPI0 ↵
DEVICE CODE? ↵
DISK UNIT NAME? ↵

LDU UNIQUE ID = CACTUS
LDU NAME = CACTUS

COPY TO DISK (D), LABELED TAPE (T), OR FLOPPY (F)? F ↵
FLOPPY UNIT NAME? [DPI1] ↵
DEVICE CODE? ↵
SPECIFY VOLUME ID VOL1 ↵

COPY TO FLOPPY FROM DISK. PLEASE CONFIRM (N/Y)? Y ↵
CONFIRMED.

.
. (time passes)
.

MOUNT NEXT FLOPPY, STRIKE ANY KEY WHEN READY.

(Mount next diskette and strike key.)

.

. (time passes)

.

MOUNT NEXT FLOPPY, STRIKE ANY KEY WHEN READY.

(Mount next diskette and strike key.)

.

. (time passes)

.

DONE!

PCOPY Dialog, Diskette-to-Disk

Before you start, get the set of diskettes that PCOPY copied to. The label should be written on the first diskette envelope; and all diskettes should be labeled with their dump sequence numbers.

When you're ready, bootstrap PCOPY from disk if you can.

If you *cannot* bootstrap PCOPY from the disk (for example, if the disk structure has been damaged), you'll need to use the PCOPY diskette (that you made earlier). To bootstrap, you must make the diskette number 0 and disk number 1, as described in Chapter 3, step 3. Then bootstrap as usual and answer PCOPY↵ to *SYSTEM PATHNAME?* When PCOPY comes up, remove the PCOPY diskette from the unit.) Next, make the diskette number 1 and disk number 0, described in Chapter 3, step 27. Then proceed as follows.

AOS DISK COPIER REV n

ENTER TODAY'S DATE (MM DD YY)?

1. Enter today's date, separating numbers by one space; e.g.,

   8 4 85↵

   WILL THIS BE A DISK TO DISK PCOPY [N]?

2. Press ↵ for the default answer, No, shown in square brackets.

   IS SOURCE A LOGICAL DISK (D), LABELED TAPE (T), OR FLOPPY(F)?

3. Type F↵

   FLOPPY UNIT NAME [DPI1]?

   Type the name of the diskette unit that holds the diskette (e.g., DPM0); or press ↵ if the diskette is in the default unit, DPI1.

   DEVICE CODE?

4. If the diskette is connected to the default device code (20 for DPM0), press ↵. Otherwise, type the device code.

   SPECIFY VOLUME ID

5. Type the volume ID that was specified for the diskette dump. This should be written on the paper envelope. (If you can't determine the volume ID, make one up; later, when PCOPY reads the volume ID, it will tell you what the label is in the error message. Then type CTRL-C CTRL-A to restart PCOPY and type the actual volume ID. But remember that, to restore the LDU properly, the diskettes must be copied to the LDU in the same order that they were copied *from* the original LDU.)

   For example, if the label were VOL1, you'd type VOL1 ↵.

   SPECIFY DESTINATION LDU
   ENTER UNIT NAMES FOR EACH DISK IN LDU (<NL>) WHEN DONE
   DISK UNIT NAME?

6. Respond with the name of the first disk in the LDU; e.g., DPI0 ↵.

   DEVICE CODE?

7. Unless you know that this disk is connected to a nonstandard device code, press ↵ for the default. Otherwise, type the device code of the unit's controller.

   DISK UNIT NAME?

8. PCOPY cycles *DISK UNIT NAME?/DEVICE CODE?* questions until you respond ↵ to *DISK UNIT NAME?*. Be sure to specify all disks in the LDU, if a multiple-disk LDU was originally created with the Disk Formatter.

   When you've specified each disk, press ↵ at *DISK UNIT NAME?*.

   COPY TO DISK FROM FLOPPY. PLEASE CONFIRM (N/Y)?

9. If there are any files you want on the destination LDU, type N ↵. PCOPY overwrites all material on the destination medium. PCOPY gives you this chance to confirm the write. To confirm, type

   Y ↵

   CONFIRMED.

   PCOPY now checks the volume ID and sequence number. If these are correct, it starts copying the diskette to the LDU. If PCOPY hits an error, it will display one of the error messages shown in Appendix C,Table C-7. (But if the disk or diskette unit is not ready, PCOPY will wait in silence until you make it ready.)

10. After PCOPY has finished copying from the diskette, it says

    MOUNT NEXT FLOPPY, STRIKE ANY KEY WHEN READY.

    Remove the diskette from its unit and replace it in the envelope. Insert the next diskette in the unit.

11. Strike a key.

    If the sequence number is correct, PCOPY now copies from the new diskette to the LDU. (If the sequence number is wrong, PCOPY will display an error message and prompt for the correct diskette in sequence. Try to find and and mount the correct diskette. If this succeeds, fine; if not, you will have to restart the PCOPY from the first diskette). When PCOPY has finished copying from this diskette, it displays the message shown in step 9. Repeat steps 9 and 10 until until PCOPY says:

    DONE!

12. Now, if you restored an AOS system disk, you might try booting the system on the destination disk. Type the copy LDU disk unit name(s), then device code(s) as SYSBOOT asks *DISK UNIT NAME?* and *DEVICE CODE?*.

    Remove the last diskette from its unit and store all diskettes safely.

    To copy another LDU, place its disks in their units (if applicable); then press the CONTINUE panel switch (or, with an MV/8000, type CONTINUE⤶; with other programmed consoles, type P⤶). Make sure you have enough formatted, labeled diskettes, and return to step 1 in this section.

Diskette-to-Disk PCOPY Example –

SYSTEM PATHNAME? :PCOPY ⤶

AOS DISK COPIER REV n

ENTER TODAY'S DATE (MM DD YY)? 8 4 85 ⤶
WILL THIS BE A DISK TO DISK PCOPY [N] ? ⤶
IS SOURCE A LOGICAL DISK (D) LABELED TAPE (T), OR FLOPPY(F)? F ⤶

FLOPPY UNIT NAME? [DPI1] ⤶
DEVICE CODE? ⤶
SPECIFY VOLUME ID VOL1 ⤶
SPECIFY DESTINATION LDU
ENTER UNIT NAMES FOR EACH DISK IN LDU (<NL>) WHEN DONE
DISK UNIT NAME? DPI0 ⤶
DEVICE CODE? ⤶
DISK UNIT NAME? ⤶

LDU UNIQUE ID = CACTUS
LDU NAME = CACTUS

COPY TO DISK FROM FLOPPY. PLEASE CONFIRM (N/Y)? Y ⤶
CONFIRMED.

.
. (time passes)
.

MOUNT NEXT FLOPPY, STRIKE ANY KEY WHEN READY.

(Mount next diskette and strike key.)

.
. (time passes)
.

MOUNT NEXT FLOPPY, STRIKE ANY KEY WHEN READY.

(Mount next diskette and strike key.)

.
. (time passes)
.

DONE!

Table 9-3. PCOPY Error Messages

Message	Meaning and Action
(nothing)	A pertinent tape or disk unit may not be ready or on-line. Check; if any is not ready, make it ready and PCOPY will continue.
BAD SEQUENCE NUMBER, INCORRECT FORMAT FOR PCOPY	You have mounted the wrong tape or disk; or you mounted a tape or disk not written by PCOPY. Dismount or replace the tape or disk and replace it with the correct one.
BAD TAPE - TAPE ERROR DURING RECOVERY OF TAPE ERROR	PCOPY encountered an error and — while trying to fix it, encountered another error. If you are dumping to tape, mount another tape and continue. If you are verifying or loading from tapes, PCOPY aborts. If you are verifying tapes, try another tape (or unit) and restart the original PCOPY. If you are loading from tape, and you must load this tape, try cleaning the tape unit heads, or try the tape on another unit.
BITMAP DISK HAS BEEN ALTERED RESTORE BITMAP DISK TO DEVICE CODE nn UNIT NUMBER n	For a disk-to-disk PCOPY with all disks not on-line, the first disk to be copied must remain in the same unit, on-line, throughout the copy. Put the bitmap disk in this unit and restart; or specify the unit that holds the bitmap disk. Restore the bitmap disk to its original unit. You may need to restart; or try typing CONTINUE ↵.
BITMAP NOT ALIGNED, SHOULD BE AT n	The bitmap address on the destination LDU does not match the bitmap address on the source LDU. The Disk Formatter default address for the bitmap (and thus for the overlay area and remap area) changed between AOS revisions, so you may see this message for a destination LDU formatted with a post-change Formatter. To retain the destination LDU's bitmap/overlay area/remap area addresses, run a Disk Formatter Partial format on the destination LDU, and note the bitmap, overlay, and remap area addresses. Then, again with a Formatter Partial format, try to move the *source* LDU areas to the same addresses as on the destination LDU. If this succeeds, try PCOPY again. If it doesn't succeed, try to find some other way to get the source LDU's files onto the destination LDU, then run a Full format on the source LDU and default the bitmap and overlay area addresses. Future PCOPYs with the source LDU will then work without this error. If you *cannot* get the source LDU's files onto the destination LDU without using PCOPY, use a Disk Formatter Partial format to move the destination LDU's bitmap, overlay area, and remap area to the same addresses as on the source LDU. Unfortunately, this may mean moving the bitmap to the beginning of the LDU, which is not an ideal place for performance.
BITMAP NOT ON THIS DISK	For a disk-to-disk PCOPY with all disks not on-line, the first disk to be copied must have the bitmap on it. This disk must be on-line throughout the copy. Put the bitmap disk in this unit and restart; or specify the unit that holds the bitmap disk.
CHECK FOR PROPER DENSITY	PCOPY encountered an error that may indicate incorrect density; check the unit for correct density.

(continues)

Table 9-3. PCOPY Error Messages

Message	Meaning and Action
DISK ERROR,DEVICE n u STATUS= s	An unrecoverable disk error occurred. See if the disk is ready and write-enabled. If not, write-enable it and type CONTINUE ⤶ or press CONTINUE and retry PCOPY. If the disk is write enabled, check the error status *s* in the *Programmer's Reference Manual for Peripherals*, or disk hardware manual. It may be possible to run a Disk Formatter Partial format on the disk; if not, the disk may need hardware formatting.
FATAL DISK ERROR, STATUS= s	See *DISK ERROR* message.
FATAL TAPE ERROR, STATUS= s	See *TAPE ERROR* message.
FATAL TAPE READ ERROR, RETRIES = 15	PCOPY cannot read the tape, although it has retried 15 times. PCOPY aborts. If you must load the tape, try another unit, or clean the unit's read/write heads.
FILE ID DOES NOT MATCH LDU UNIQUE ID FILE ID =file-id LDU ID =ldu-name *IF YOU WISH TO CONTINUE, PLEASE CONFIRM (Y/N)?*	The fileset ID (filename) label on the tape does not match the LDU unique ID, as assigned with the Disk Formatter. The LDU does not have the same ID as the one that was copied with PCOPY; it may have valuable files on it. Do not confirm unless you are certain that you want to write this tape set to this LDU. (You would confirm, for example, if you knew that the LDU had been Full formatted since it was copied. You would also confirm if you knew that the LDU had been given a new unique ID via a Partial format run since it was copied.) If you confirm with Y ⤶, PCOPY ignores the LDU ID and writes the tape to the LDU. If there are more tapes in the original dump, PCOPY will repeat this message for each. If you type N ⤶, PCOPY aborts; and you may want to find the tape set with the correct LDU name as its file ID. From AOS, you can discover the file ID on any tape by typing TYPE @unitname ⤶; the file ID is in bytes 22-27 of the label.
INCONSISTENT BLOCK COUNT ON EOF1 (EOV1) LABEL	The number of blocks read or written from tape is not the same as the block count in the end-of-file (or end-of-volume) label. If this is a dump, tell PCOPY Yes to the *SHOULD THE TAPES BE LABELED* question and restart from the beginning. If this is a load, at least *some* files were loaded — although they may not be complete. All you can do, on a load, is retry from the beginning. In either case, you may need to restart the tape via tape panel switches.
INCONSISTENT DIB INFO	The Disk Information Block on one of the disks indicates that you may have made an error when you specified the disk unit name. You may have specified a disk that doesn't belong to the LDU; or you may have specified the disks in the wrong order; or the DIB itself may be invalid (perhaps the disk wasn't formatted with the Disk Formatter). Verify that the correct disks in each LDU are mounted and that you specified them correctly. Then restart PCOPY by typing ⤶ or by pressing CONTINUE and rerun from the beginning.

(continued)

Table 9-3. PCOPY Error Messages

Message	Meaning and Action
INCORRECT VOLUME ID	The tape volume ID you specified does not match the ID of the tape mounted. PCOPY repeats its volume ID request. You can type the actual ID (as given by PCOPY in this message), or you can try to locate the volume with thc ID you typed. On a dump to tape, you can get PCOPY to label the tapes with your specified names. To do this, answer Y) to the *SHOULD THE TAPES BE LABELED* question.
INCORRECT VOLUME SEQUENCE NUMBER	The tape volume is out of order. Put the proper volume on a tape unit and/or respecify as asked.
INVALID DATE	You entered an invalid expiration date or today's date. Retry.
INVALID OR MISSING HDR1 LABEL	The HDR1 portion of the label is invalid or missing; perhaps the label was not written by the LABEL program or PCOPY. The tape is unusable as is. On a load, try to find and mount the correct tape. On a dump, get PCOPY to label the tapes by answering Yes to the *SHOULD THE TAPES BE LABELED* question.
INVALID OR MISSING VOL1 LABEL	The VOL1 portion of the label is invalid or missing. See the preceding message.
LDU SIZE MISMATCH	The destination LDU is not the same size as the source LDU. The two LDUs must include identical disks, aside from the IDs and names. If they *are* identical, run a Disk Formatter Partial format and make sure that the octal addresses (DISK NUMBER n, m through n) are consistent on the LDUs.
OVERLAY AREA NOT ALIGNED - SHOULD BE AT n	The destination LDU's overlay area does not match the source LDU's overlay area. See *BITMAPS NOT ALIGNED* message above.
OVERLAY AREA SIZE MISMATCH	The destination LDU's overlay area is not the same size as the source LDU's. Run a Disk Formatter Partial format on the destination LDU. Whenever anyone full-formats an LDU, he or she should default the overlay area size; this will prevent many problems, including this one.
REMAP AREA NOT ALIGNED - SHOULD BE AT n	The destination LDU's remap area address does not match the source LDU's remap area address. See *BITMAP NOT ALIGNED* message above.
REMAP AREA SIZE MISMATCH	The destination LDU's remap area size differs from the source LDU's. See *BITMAP NOT ALIGNED* message above.
RESPECIFY TAPE UNITS STARTING AT REEL n TAPE UNIT NAME?	This usually follows a labeled tape error message. You may not need to restart the dump or load operation from the beginning. If you can correct the error (e.g., by typing the correct volume ID or having PCOPY label the tape), you can resume the operation at reel *n*.
SET ID DOES NOT MATCH 1ST VOLID	The tape volume ID does not identify it as the first volume in the dump. (The tape's HDR1 file ID doesn't match the first tape volume's HDR1 fileset ID). Try to find the first tape of the dump. PCOPY describes both the *actual* volume ID on the tape and the whole *label* of the tape it wants.

(continued)

Table 9-3. PCOPY Error Messages

Message	Meaning and Action
TAPE ERROR, status=s	An unrecoverable tape error has occurred. This can occur if the tape is not a labeled tape. It can also occur if the tape heads are dirty (try cleaning them) or if the heads are out of alignment. Restart PCOPY if needed. If you are verifying tapes, replace the tape; you must redo the whole PCOPY from the beginning. If the error recurs, check the *Programmer's Reference Manual for Peripherals*, or appropriate tape manual. Try another unit if you have one.
TAPE HAS NOT REACHED ITS EXPIRATION DATE DO YOU WANT TO OVERRIDE IT? (Y/N)	The expiration date given or defaulted when PCOPY last wrote to this tape set has not been reached. If you want to override, and write to the tape set, type Y ↵. To preserve the the set, type N ↵ and mount an expired set.
TAPE IS WRITE LOCKED INSERT A RING AND...	Do what it says.
TAPE WRITE ERROR, STATUS= s	PCOPY cannot write to the tape, although it has retried 15 times. It cannot recover from the error. Mount another tape and continue. If the error recurs, check the error status in the *Programmer's Reference Manual for Peripherals.*
TOO MANY PHYSICAL UNITS IN LDU	You tried to specify more than eight physical disk units for the LD. Retry, specifying the correct number of physical disks.
WRONG DISK FORMAT REV	The LDU you have mounted is of an obsolete revision. Run a Disk Formatter Partial format on the LDU. This may fail; if so, try to DUMP the material (if there is any you care about); then run a Full format.
WRONG NUMBER OF PU'S IN DESTINATION LDU	The destination and source LDUs do not contain the same number of physical disks. Probably you mounted the wrong destination (or source) LDU.

(concluded)

What Next?

In this chapter, you've read about many of the runtime tools other than EXEC: the master CLI, templates, ACLs, and filename suffixes; LOCK_CLI, for security; PED, for an overview of system activity; SYSLOG and REPORT, for a *record* of system activity; and DUMP and PCOPY, to back up and restore all your vital disk-based material.

Other tools include the Disk Formatter, described in Chapter 11. Or, you might want to read about system management considerations, in Chapter 14.

End of Chapter

Chapter 10
Unusual System Conditions

Read this chapter

- if software-based abnormal shutdowns recur and you want to submit a Software Trouble Report to DG.

An unusual system condition is a situation in which you are having recurrent problems (or suspect problems) with your system hardware or software.

This chapter explains some tools to help you identify the source of actual and potential problems. The major sections proceed

- Hard, Soft, and Software Errors
- Software Problems — How to Submit a Software Trouble Report
- What Next?

Hard, Soft, and Software Errors

A hard error is a noncorrectable hardware error; if one occurs in the main processor, it halts the main processor. A soft error is a correctable hardware error; it doesn't halt the main processor. A software error is an AOS operating system or other software flaw.

Any of these errors produces an error message on the system console. If the error causes an AOS panic *(FATAL AOS ERROR)*, the panic values describe conditions when AOS detected the error.

Aside from panic and other system console messages, AOS has SYSLOG to help identify current (and latent) errors. Also, the AOS Release Notice has a "Notes and Warnings" section that often contains useful information. (All serious error messages and their meanings are summarized in Appendix A.)

The hardware itself can help diagnose errors. There are disk status lights and codes (for example); and some computers (like S/280s and MV/8000s) can detect hardware problems and display descriptive codes (also described in Appendix A). MV/8000s have a diskette-based SCP log, described in the *ECLIPSE MV/8000 System Control Processor* manual.

And there are human resources, including your DG support organization, and DG itself, accessible by a *Software Trouble Report*. DG support people are often available for help, but you can assist them, and your installation, by taking a few steps of your own — as described in this chapter.

A precautionary note: If a hard error occurs on disk, FIXUP aborts, and you suspect media damage, do not insert another pack in the unit (if this applies). Cut power to the unit and consult your DG support organization.

Software Problems — Submitting a Software Trouble Report (STR)

If you are having serious system problems, and your organization is *outside* the continental United States, you should generally consult your local DG systems engineer about them.

If your site is inside the continental US, and you have a contract with the Atlanta support center, call the support center for help. If you don't have a contract with the support center, you will probably need to submit a Software Trouble Report (STR) on your own. The *Software Trouble Report Guide,* 012-1407, explains this. But, for your convenience, we'll review the steps involved.

Wherever you are, it's important to discover whether the problem is software or hardware. SYSLOG and diagnostic programs (if recommended by a DG engineer) can help in this area. You should be sure that you are running a current version of AOS and that all generally-known patches and hardware fixes have been applied to your system. You may also be interested in the *AOS Monthly Newsletter;* DG publishes STR responses in this newsletter.

The best way (often the *only* way) to identify a software bug is to reproduce the environment in which the bug appeared. To do this, DG requires a copy of the software *and* information on your environment. If you don't provide these, DG may not be able to help you. It's very frustrating to receive STRs that say simply "EXEC HANG" or "SYSTEM TRAPPED", and it may be impossible to provide useful answers to such STRs. So, it's in your interest to provide all the background information that you can.

Before you decide to submit an STR, be sure that all current patches have been installed. This includes peripheral manager, Ghost, CLI, AOSGEN, and other patches. If you patch AOSGEN or any associated library, run AOSGEN again to generate a new system; then apply all AOS patches to this system. In many cases, patching will clear up an error situation — allowing your applications to continue normally, and eliminating the need to submit an STR. Patches are easy to apply with the PATCH utility (Chapter 4).

Gathering STR Information

Ultimately, if you decide that you want to submit an STR, set up for the STR by running AOS and other software that you normally use, with both SYSLOG and the SCP ERRLOG (if any) running. Then, if the error condition occurs, execute the following steps.

1. Collect as much information as you can about the problem, including OP console output and a summary of the problem that includes information on how often the problem has occurred, what you were doing before the problem occurred, what happened afterward, and whether you noticed anything that seemed unusual. In other words, you should provide as much information as you can about the circumstances surrounding the problem. The more information you can provide about the problem, the more quickly and easily DG can solve it for you.

2. If AOS has panicked *(displayed FATAL AOS ERROR),* record the panic code and values (or, with a hardcopy console, use the listing). Go to step 10.

3. Note all processes that are/were running, including both DG and user application. Describe types and priorities if you can.

4. If a user process (son of EXEC) is malfunctioning or is deadlocked, note its process ID (pid). Then create a break file and dump it to tape via the following steps.

   ```
   ) SUPERPROCESS   ON ↵
   +) TERMINATE/BREAK=pathname   pid ↵
   ```

   (Terminate process and create a break file named pathname. If the process won't terminate, you will need to force a shutdown. Go to step 8.)

   Mount tape on an available tape unit; e.g. MTB0.

   ```
   COPY   @MTB0   pathname ↵
   ```

   Go to step 17.

5. If an OP process (son of the master CLI) other than EXEC is malfunctioning or deadlocked, follow these steps. If the process is a CEO process, try to shut down CEO (type SUPERPROCESS ON ⤶, CEO.SYSTEM STOP ⤶). If the process is a XODIAC or INFOS II process, try to shut these processes down as you would for a normal shutdown. Now, follow the steps shown under step 4.

6. If the error condition is a process trap, follow these steps. When a process traps, it automatically creates a break file, with .BRK suffix, in its working directory. Note the trap type (e.g., VALIDITY), the program counter, accumulators (ACs), and carry (C). Use the CLI COPY command to copy the .BRK break file to tape; e.g., COPY @MTB0 pathname ⤶; and go to step 17.

7. If more than one user process (but not every user process) is hung or malfunctioning, or if EXEC displayed an *INTERNAL INCONSISTENCY* message, then EXEC, the peripheral manager, or the system itself may be having problems. You will need to force a shutdown. Go to step 8.

   If possible, warn users to log off if they can (use the SEND command or BROADCAST macro). If CEO, XODIAC, or INFOS II processes are running, try to shut them down normally (but *don't* use the DOWN macro, since it terminates EXEC).

   After you have as many processes as possible shut down, force ESD. Go to step 8.

Forcing ESD

8. If the entire system is hung, note whether the system console responds to commands. (This helps us determine whether the problem is in the peripheral manager or AOS itself.) Also, note whether disk READY lights are blinking (the READY light blinks when the disk is busy). If tape I/O is in process, note whether the tape(s) are moving.

   NOTE: In the following procedures, execute the I/O RESET command only once each time you attempt to force ESD.

   ESD may not run properly if you do not execute I/O RESET once, but it may lose information if you execute I/O RESET more than once. (For example, the S/140-type programmed console command R causes an I/O RESET, but the S/20-type programmed console command R does not execute an I/O RESET. Therefore, to cause an I/O RESET on an S/20-type programmed console machine, you must type the I command before you type the 14R command.) Consult your hardware manuals if you are unsure about which command executes an I/O RESET for your machine.

   With a programmed console, continue with step 8a. With data switches, skip to step 8b.

8a. Unlock the computer (if locked). Then, proceed as follows to force ESD:

- *If you have an MV/8000,* type the break sequence. The break sequence causes the *SCP-CLI>* prompt to display on your console. Then, type HALT⤶. To get the status information that you need, type

  SCP-CLI> . ⤶ (Type . ⤶ [period].)

AC0	*AC1*	*AC2*	*AC3*	*PC*	*CARRY*	*MAP*
nnnn	*nnnn*	*nnn*	*nnn*	*nnn*	*n*	*xxxx*

  SCP-CLI> CONTINUE ⤶

  Record the contents of the ACs, the PC, the carry bit, and the map status. Repeat this procedure four times. The fourth time, however, do *not* type CONTINUE⤶. Instead of CONTINUE⤶, type

  SCP-CLI> RESET ⤶
  SCP-CLI> START 14 ⤶

  and go to step 9.

- *If you have an S/140-type programmed console,* type the break sequence. The break sequence causes the *!* prompt to display on your console. Record the program counter, which prints before the *!* prompt. To get the status information that you need (the contents of the ACs, the map status, and the carry bit), type

```
! 0A  nnnnn
! 1A  nnnnn
! 2A  nnnnn
! 3A  nnnnn
! 10A nnnnn
! 12A nnnnn
! P
```

  Record the contents of the ACs, the map status, and the carry bit. Repeat this procedure four times. The fourth time, however, do *not* type P↵. Instead of P↵, type

```
! 14R
```

  and go to step 9.

- *If you have an S/20-type programmed console,* type the break sequence. The break sequence causes the ! prompt to display on your console. Record the map status, which prints before the ! prompt. To get the status information that you need (the contents of the ACs, the PC, and the carry bit), type

```
! 0A nnnnn
! 1A nnnnn
! 2A nnnnn
! 3A nnnnn
! 4A nnnnn
! 5A nnnnn
! P
```

  Record the contents of the ACs, the PC, and the carry bit. Repeat this procedure four times. The fourth time, however, do *not* type P↵. Instead of P↵, type

```
! I
! 14R
```

  and go to step 9.

- *If you have an S/120-type programmed console,* type the break sequence and record the contents of the ACs, the PC, the carry bit, and the map status, which display on your console. The map status displays on your console in the following way:

If the prompt is	then the map is
!	off
A!	user A map
B!	user B map
C!	user C map
D!	user D map

  Then, type

```
! P
```

  Repeat this sequence four times. On the fourth time, however, do *not* type P. Instead, type

```
! 14R
```

  and go to step 9.

8b. If you have data switches on your front panel, unlock the computer (if locked). Press STOP. Turn all of the rotary switches to the left (counter-clockwise). Note the PC that is in the lower of the two sets of lights. Then, note the status of the other lights (ION, CARRY, SUPER, USER, DCH, BMC, IOP, etc.) and press CONTINUE.

    Repeat this procedure three times.

    Now, press STOP and turn all of the rotary switches to the left (counter-clockwise). Note the PC that is in the lower of the two sets of lights. Note the status of the other lights (ION, CARRY, SUPER, USER, DCH, BMC, IOP, etc.), press RESET, set the data switches to 000014, and lift the START switch.

    Go to step 9.

9. The system console displays

    AOS PROCESSING ABORTED
    STRIKE "D" FOR AOS DUMP, "S" FOR SHUTDOWN, OR "R" FOR OPEN...
    ?

    Strike D for a dump.

10. For tape, get a tape at least 800 feet long, with the write-enable ring in. Mount it on unit 0 on the first controller. If the unit has a density switch, choose DENSITY HIGH.

    For diskette, get a diskette, make sure it is write-enabled, and insert it in unit 0. Several diskettes may be needed.

11. The system console says

AOS SYSTEM DUMP		*AOS SYSTEM DUMP*
LOAD TAPE ON MTA0	or	*LOAD DISKETTE FOR DUMPING*
STRIKE "D" WHEN READY:		*STRIKE ANY KEY TO CONTINUE*

12. Strike D. The dump proceeds. When done, the routine says

    STRIKE "S" FOR AOS SHUTDOWN, OTHER KEY TO HALT
    ?

13. Strike S to start ESD. If ESD succeeds, it displays the message *SYSTEM SHUTDOWN*; go to step 15.

    If ESD displays *ABNORMAL SHUTDOWN*, please restart ESD (type RESET then 14R ↵) and take another Memory Dump (steps 10 through 14). Then run FIXUP (step 14).

    If ESD panics, please note the panic values, dismount the tape, mount another tape, and take another Memory Dump (this may help identify the problem and *will* help us improve ESD). Restart ESD (RESET, I, 14R, etc.). If ESD panics again, ask for an open file report (R) and note which LDUs have open files.

14. Run FIXUP. Be sure to specify 3 for VERBOSITY and LPB as the error log file. If FIXUP produces pertinent error messages, submit the listing. FIXUP is described in Chapter 6.

15. Reboot AOS and use the CLI command COPY (not DUMP) to copy the symbol table file, sys.ST, to file 1 of the tape. For a tape example, if the Memory Dump tape is mounted on MTB0, type

```
) SUPERUSER   ON ↵
*) DIR   :SYSGEN ↵
*) COPY   @MTB0:1   sys.ST ↵
```
(sys is the name of your tailored system, without the .SY suffix.)

    For diskette, insert a write-enabled diskette in a unit and copy the symbol table to it. For example

```
) SUPERUSER ON ⤶
*) DUMP/V @DPI1 sys.ST ⤶
```

The symbol table is a critically important part of the STR. Without it, we won't be able to analyze your problem.

16. If there is any other software (like a user program) involved, please add it to the tape or a different diskette. Please add all patch history files (:UTIL:+.PH and :PATCH:+.PH). And if you suspect that the problem may be in EXEC, please add file :UTIL:EXEC.CONSOLES to the tape. For all material except the core dump (file 0) and the symbol table (file 1), please use the DUMP command. Please dump all these additional files to tape file 2. Also, please note what you have done on the tape reel.

17. If you can, submit the following information about the system environment at the time of the error condition.

    - devices that were attached to the system when the problem occurred. Do this via QPRINT :SYSGEN:sys.CSF ⤶, and note user devices and device codes (if any) on this listing;
    - hard, soft, or other pertinent error messages that appeared on the system console before and after the problem occurred;
    - A SYSLOG report, if it includes any pertinent hard, soft, or other error information.

Please report any event or circumstance needed to simulate the conditions at your site. Label the tape and all listing materials, and submit them to your DG support organization. Thank you.

What Next?

This chapter has explained something about errors and all about filing a Software Trouble Report (STR).

You may want to read about the Disk Formatter program, described in the following chapter, or system management issues, covered in Chapter 14.

End of Chapter

Chapter 11
The Disk Formatter

Read this chapter

- when you want some background on AOS logical disk units and how to use them;
- when you want to format one or more new (blank) disks, or reformat old ones;
- when you want to enter one or more bad blocks (noted as hard errors by AOS or unreadable by FIXUP) for a disk;
- when you want to rename a logical disk unit or change its access control list.

The Disk Formatter is an AOS utility program that formats one or more physical disks into a logical disk unit (LDU). If you brought up your own first system (Chapter 2 or 3), you already have some experience with the Disk Formatter. This chapter explains the rest, in the following sections:

- About the Disk Formatter
- About LDUs
- If You Make a Mistake
- Starting the Stand-Alone Disk Formatter
- Starting the Stand-Among Disk Formatter
- The Full Format
- The Partial Format
- Disk Formatter Error Messages
- What Next?

About the Disk Formatter

There are two versions of the Disk Formatter — a stand-alone version that runs only when AOS is *not* running; and a stand-among version, that runs under AOS. Each version offers two formats, Full and Partial.

The practical differences between the *versions* are that you must use the stand-alone version for disk(s) in the current LDU, or to format disks on controllers that are not supported by the current AOS system. You can use the stand-among version for any LDU that is *not* the master.

There's a big difference between the Full and Partial *formats*. The Full format ignores all AOS file structure and writes a new bitmap on the LDU, effectively destroying all AOS files on it; it can also write patterns to check the disk surface for flaws. The Partial format retains the old bitmap and uses read-only surface analysis. So — if there are AOS files that you want on the disk — you should use the Partial format of the pertinent Formatter.

Neither version of the Disk Formatter lays down a hardware format on a disk. Certified DG disks are shipped formatted; but if you ever need to reformat a disk, consult your DG support organization or engineer for details on how to use DTOS diagnostics for peripherals.

About LDUs

This section explains some things about LDUs, and how you use them.

Single- and Multiple-Disk LDUs

You can create a single- or multiple-disk LDU with a Disk Formatter Full format. A multiple-disk LDU can include up to eight disks. For any LDU, all disks involved must be ready before you can access the LDU.

The disk unit(s) in which you *format* an LDU are irrelevant to the LDU. For example, with removable packs, you can format an LDU in unit DPF0 and run it in unit DPF11; or you can format a two-disk LDU in DPF12 and DPF23 and run it in DPF1 and DPF2. This is why — during bootstrapping — you are asked to specify each disk in the LDU and its device code. It's also why, when you initialize an LDU from the CLI, you do it by disk unit name. Operations are simpler if each disk in an LDU has a "home" unit, but this is not required. Bootstrapping is easier if you have a system LDU in unit 0 — but even this is not mandatory for a tailored AOS system.

Single-disk LDUs are easier to use because they involve only one disk unit. Also, FIXUP and PCOPY disk-to-disk operations are simpler with single-disk LDUs.

Nearly always, your primary *system* LDU will be a single-disk LDU. With a single-disk system LDU, someone need only type one disk unit name when bringing up AOS. After AOS is up, other LDUs can be grafted onto the system LDU with CLI INITIALIZE commands. You can put these commands in the macro UP.CLI. These LDUs can be released with CLI RELEASE commands; and you can put the RELEASE command(s) in the DOWN.CLI macro.

There are two reasons to make your system LDU a multiple-disk LDU. The first is that you expect a file to span more than one disk when you own only two disk units. The second is that you have a fixed-head disk and want to run AOS from it for higher performance. In the latter case, you should specify the fixed-head disk as first unit in the LDU, and install a disk and system bootstrap, and, optionally, an AOS system on it. You can then boot AOS from the fixed-head disk device code.

The real advantage of a multiple-disk LDU is that it allows a contiguous file to span more than one physical disk. Some DG data management products — like INFOS II and DG/DBMS — may need such huge contiguous files. If your site will use such a file, you will need to build a multiple-disk LDU for it. Ideally, you'd run this LDU in *addition* to a single-disk system LDU.

An example of a system with a single-disk system LDU and a multiple-disk LDU named DATABASES is shown in Figure 11-1.

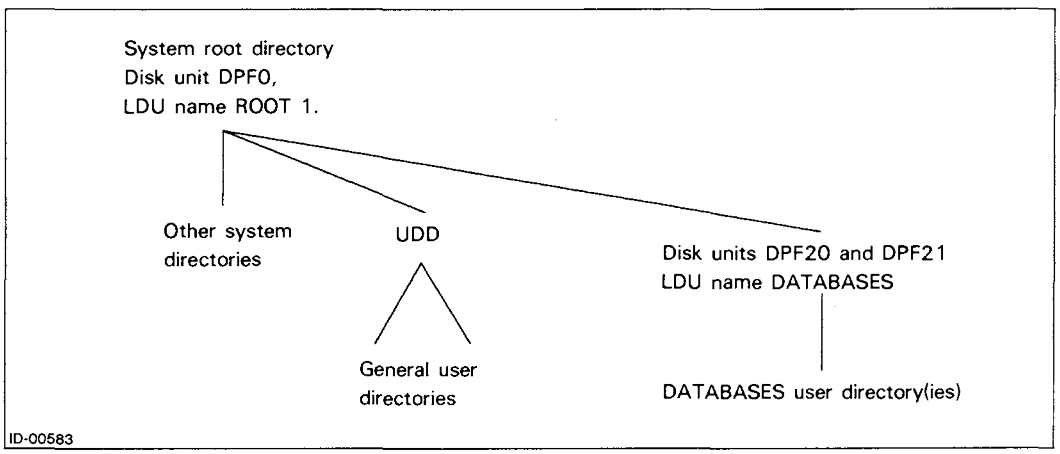


Figure 11-1. A Multiple-LDU AOS System

To set up the system in Figure 11-1, you would do the following things:

- Use the Disk Formatter to create an LDU named DATABASES.
- Bring up AOS. From the CLI, initialize the disk units that hold LDU DATABASES (INITIALIZE @DPF20 @DPF21↵). AOS then recognizes the LDU by the filename DATABASES.
- For each username you want on DATABASES, use the CLI MOVE command to move the user directory to DATABASES. For example, for user JACK:

  ```
  *) DIR :UDD ↵
  *) MOVE/V DATABASES JACK:# ↵
  ```
  ...(CLI verifies directories and files moved)...
- Delete the user directory in :UDD. For example

  ```
  *) DELETE/V JACK:# ↵
  ```
- Create a link named username in :UDD to :DATABASES:username. For example

  ```
  *) CREATE/LINK JACK DATABASES:JACK ↵
  ```

Thereafter, while DATABASES was initialized, any user whose directory had been moved to DATABASES would have the whole LDU available for database operations. The user directory pathname would be :DATABASES:username, but otherwise the user would be treated exactly as if he or she were in :UDD.

At startup, after AOS was brought up from DPF0, the UP.CLI macro could use the following commands to initialize the DATABASES LDU:

```
DIR :
INITIALIZE DPF20 DPF21
```

The DOWN.CLI macro would need only the following command to release DATABASES:

```
RELEASE :DATABASES
```

(This technique works for any nonmaster LDU. For example, assume that you have a lot of users and want to put some of them on their own LDU — perhaps named UDD1. You'd use the Disk Formatter to create UDD1, initialize it from the CLI, DIR into :UDD, move each directory you wanted over to UDD1, delete each of these directories from :UDD, and create links in :UDD to :UDD1:username. Then you'd put the appropriate INITIALIZE and RELEASE commands in the UP and DOWN macros.)

No CLI commands are needed to initialize a physical unit used for swapping. Such a unit is initialized automatically if specified to AOSGEN; or you can give the disk unit name during bootstrapping if you override default specs.

If — at any point — you want to change the physical disk configuration in an LDU, you must run a Formatter Full format on all physical disks involved. Dump all files from each LDU involved, run a Full format on the LDU, then load the files onto the new LDU.

LDU Access Control Lists (ACLs)

When you run the Formatter on an LDU, you can specify an access control list (ACL) or take the default, which creates a null ACL. For the master LDU, AOS ignores the ACL, so that it can execute programs (like the CLI) for users. (If — at AOSGEN — access control was not enabled for this system, then this system ignores *all* ACLs when it runs. But generating and running a system without access control is useful only when you want all users to have access to all files.)

For each nonmaster LDU, the ACL you specify (or default) with the Disk becomes effective when the LDU is initialized from the CLI. If the LDU has the default (null) ACL, some kind of access must be specified before anyone but a Superuser can access the LDU. You can do this with the CLI command ACL any time after the LDU is initialized. The easiest way to do it is in the UP macro; for example

```
DIR  :
INITIALIZE/S   @DPF10
WRITE   Initialized :DPF10 as [!STRING].   ACL is OP   WARE,,,,+ E
ACL   [!STRING]   OP,WARE   +,E
WRITE/L=LDU.NAME   [!STRING]
```

The ACL you specify in an ACL command is effective only while the LDU is initialized. The Formatter-specified ACL returns when the LDU is released. So, if you know what ACL you want for an LDU, you can assign it in a Formatter Full or Partial run. This will eliminate the need for runtime ACL commands. If you need to change the ACL at runtime, you can do it at will, as above.

The Bitmap/Overlay Areas and System Performance

The *bitmap* is an area on each LDU that describes which blocks are free and which are used for data storage. The overlay area contains the overlays of the currently running AOS system. There is a bitmap on every LDU; there is an overlay area only on an LDU that includes a system disk.

AOS must access the bitmap every time it creates or deletes a file. It must access the overlay area whenever it needs nonresident code to perform an operation. So the location of these two areas on the LDU can affect system performance.

The default bitmap and overlay area addresses are often chosen when an LDU is created. The default bitmap address is 3/8 the distance across the first disk in the LDU; and the default overlay area address follows the bitmap. These are good general-purpose addresses for a single-disk LDU: they are close to the center, which gives fast access if the LDU is nearly full of files. And they are closer to the beginning, which gives relatively fast access if the LDU has few files.

For a multiple-disk LDU that will hold (or does hold) very large contiguous files (like INFOS II or DG/DBMS database files), you *might* want to select nondefault addresses. In such a case, specify 0 (start of LDU) for the bitmap, and specify the bitmap size (given by the Formatter) as the overlay area address. Contiguous file space will then stretch from nearly the beginning of the LDU to the end.

Generally, though, you should take the Formatter defaults on bitmap and overlay area addresses.

If You Make a Mistake

If you type a response to the Disk Formatter and want to change it before pressing ↵, press the DEL key as needed, or press CTRL-U to erase the line. If you are beyond the line containing the mistake and you want to abort, do this in one of two ways.

- If the Disk Formatter is asking a question, you can restart it at the beginning by typing CTRL-C followed by CTRL-A (or abort the stand-among Formatter by typing CTRL-C CTRL-B). The Disk Formatter does not alter the disk until surface analysis begins (Full format) or until you change a value (Partial format).
- If the Disk Formatter is running surface analysis, CTRL-C CTRL-A won't work. If you *must* abort the stand-alone Formatter, type the break sequence (CMD and BREAK, or BRK, or BREAK, depending on your system console) and RESET ↵ (MV/8000), I ↵ (other programmed consoles), or press STOP and RESET (data switches). To abort the stand-among Formatter, type CTRL-C CTRL-B.

  If you abort the Disk Formatter, restart it from the beginning.

For any Formatter error message, see Table 11-4, near the end of the chapter.

Starting the Stand-Alone Disk Formatter

If you want to run the Disk Formatter on any disk in the master LDU, you must use the stand-alone version. The stand-alone Formatter is in disk file DFMTR in the root directory, on your master LDU; and you can boot the Disk Formatter by name. (If for some reason you *cannot* boot from the master LDU, you'll need to use an AOS system tape; the stand-alone Disk Formatter is in file 2 of this tape.)

You can run the stand-alone Disk Formatter only from the system console.

First, make sure each disk you want to format is in its unit, and that each unit is write-enabled and ready.

How you boot the Disk Formatter from the master LDU depends on what type of computer you have:

- If you have an S/20, S/120, or an S/280, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type nnH, where nn is the device code. For example,

  ! 27H   (S/20, S/120, S/280)

- If you have an S/140, the system console displays a ! prompt (if not, turn power off and on again). Next to the prompt, type 11A. The system will display a 6-digit number, after which you type 1000nn ⤶, where nn is the device code. Then, after the next *!* prompt, type 1000nnL. For example

  ! 11A   *xxxxxx*   100033 ⤶   (S/140)
  ! 100033L

- If you have an MV/8000, the system console displays an *SCP-CLI>* prompt. Next to the prompt, type RESET ⤶. Then, next to the second prompt, type BOOT nn ⤶, where nn is the device code. For example

  SCP-CLI> RESET ⤶   (MV/8000)
  SCP-CLI> BOOT 27 ⤶   (or BOOT 33 ⤶ or BOOT 24 ⤶)

- If your computer has hardware data switches (numbered 0 (or X4/0) through 15), make sure they are set to 1000nn, where nn is the device code. For device code 27, set switches 0, 11, 13, 14, and 15 up; for device code 24, set switches 0, 11, and 13 up; for device code 33, set switches 0, 11, 12, 14, and 15 up. Lift and release the RESET switch; lift and release the PR LOAD (PROG LOAD) switch.

Your program load steps bring the system bootstrap into memory. It displays

SPECIFY EACH DISK IN THE LDU
DISK UNIT NAME?   DPF0 ⤶   (or DPJ0 ⤶, or DPI0 ⤶, or other name)
DEVICE CODE?   ⤶

SYSTEM PATHNAME?   DFMTR ⤶

AOS DISK FORMATTER REV n
FULL FORMAT...

To boot the Disk Formatter from tape, get an AOS system tape and mount it on unit 0. How you boot from tape depends on what computer you have.

- If you have an S/20, S/120, or an S/280, the system console displays a ! prompt (if not, turn power off and on again). Next to the prompt, type 22H, where 22 is the device code. For example,

  ! 22H   (S/20, S/120, S/280)

- If you have an S/140, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type 11A. The system will display a 6-digit number, after which you type 100022 ↲, where 22 is the device code. Then, after the next *!* prompt, type 100022L. For example

  ! 11A   *xxxxxx*   100022 ↲   (S/140)
  ! 100022L

- If you have an MV/8000, the system console displays an *SCP-CLI>* prompt. Next to the prompt, type RESET ↲. Then, next to the second prompt, type BOOT 22 ↲, where 22 is the device code. For example

  SCP-CLI> RESET ↲   (MV/8000)
  SCP-CLI> BOOT 22 ↲

- If your computer has hardware data switches (numbered 0 (or X4/0) through 15), make sure they are set to 100022, where 22 is the device code. For device code 22, set switches 0, 11, and 14 up. Lift and release the RESET switch; lift and release the PR LOAD (PROG LOAD) switch.

The step above program loads the tape bootstrap, TBOOT. TBOOT says

FROM MT-0: 2 ↲   (File number 2)

AOS DISK FORMATTER REV n
FULL FORMAT DESTROYS...

You must run a Full format

- if the Disk Formatter has never run on the physical disk(s); or
- if you want to change an LDU's physical disk configuration; for example, make two existing LDUs into one, or vice versa.

In nearly all other cases, you will want a Partial format. Skip to the pertinent section: "Full Format" or "Partial Format."

Starting the Stand-Among Disk Formatter

You can run the stand-among Disk Formatter on any disk that is not part of the master LDU. It's not as fast as the stand-alone Formatter, but it does allow AOS to remain up while you run it. You can run it from the system console or any user console enabled by EXEC.

The stand-among Disk Formatter is in directory :UTIL (loaded there during the starter system initial load). Its filename is DFMTR.PR.

First, make sure each disk you want to format is in its unit, and that each unit is write-enabled and ready.

With directory :UTIL in your searchlist, type

) XEQ DFMTR ↲

AOS DISK FORMATTER REV n
FULL FORMAT DESTROYS...

You must run a Full format

- if the Disk Formatter has never run on the physical disk(s); or
- if you want to change an LDU's physical disk configuration; for example, make two existing LDUs into one, or vice versa.

In nearly all other cases, you will want a Partial format. Go to the pertinent section: "Full Format" or "Partial Format."

The Full Format

The Disk Formatter announces itself. Then it asks whether you want a Full or Partial format.

AOS DISK FORMATTER REV n

FULL FORMAT DESTROYS ANY AOS DISK STRUCTURE, PARTIAL FORMAT RETAINS IT.

FULL (F) OR PARTIAL (P OR <NL>)?

A Full format writes a new bitmap to the LDU, effectively destroying all files by overwriting pointers to them. It also can run surface analysis patterns to check disk surfaces for bad blocks — this overwrites the files themselves. So, before you proceed with a Full command, if there is material on the LDU that you want to try and dump, abort the Formatter. For the stand-alone Formatter, use the break sequence; for the stand-among Formatter, press CTRL-C CTRL-B. From the AOS CLI, dump the files from the LDU. Then restart the Formatter as shown above.

To specify a Full format, type

F ↵

FULL FORMAT

ENTER UNIT NAMES FOR EACH UNIT IN THE LDU (<NL> WHEN DONE):

DISK UNIT NAME?

Type the name of the disk unit that holds the physical disk to be the first disk in the LDU. Table 11-1 shows the disk unit names and device codes of all disks supported by AOS. For example,

DPF1 ↵

The stand-among Formatter knows — via AOS — the device code of the disk unit. So it skips the next question and asks for another *DISK UNIT NAME?*. (If the disk controller was not generated into the current AOS system, you *must* run the stand-alone Disk Formatter.)

DEVICE CODE?

Type the device code of the disk unit you just specified. Default device codes are 27 and 67 for DPF-type disk controllers, 33 and 73 for DPI-type and other moving-head models, 26 and 66 for fixed-head controllers, and 26 for DESKTOP GENERATION systems. The device codes are shown in Table 11-1. If the disk unit you specified is on the default device code for its controller, you can press ↵ in response to the *DEVICE CODE?* query.

The Formatter repeats the question(s) *DISK UNIT NAME?* (and *DEVICE CODE?* for stand-alone) so that you can specify more than one disk. A maximum of eight physical disks are allowed in an LDU. After you describe the last physical disk, press ↵ in response to *DISK UNIT NAME?*

DO YOU WANT TO ALLOCATE A DIAGNOSTIC AREA? [N]

This question lets you reserve an area on disk for later installation of DG's Advanced Diagnostic Executive System (ADES). ADES can run from a medium other than disk, but it runs much faster from disk; also, diagnostics are easier to run remotely if ADES is on disk. To use ADES, you must purchase it and have it installed on the disk by a DG field engineer. Also, ADES requires a minimum of 8000 disk blocks (3.6 Mbytes). Consequently, your disk must be larger than 25 Mbytes. Space reserved for ADES is lost for AOS file storage.

Unless you really want to install ADES and run it from disk, answer No by pressing ↵. ADES runs only from the system disk — which means you should *always* answer No if the disk isn't the system disk. If you answer no, skip to the *LOGICAL DISK UNIQUE ID* prompt.

If you really want to reserve an area for ADES, type Y ↵. Then, the Disk Formatter asks

ENTER THE NUMBER OF BLOCKS (11610 TO 35230) THAT ARE REQUIRED ? [23420]

The displayed figures are octal. ADES needs at least 8 000 blocks (17500 octal). The default, 23420, is 10,000 blocks. Decide on the number of disk blocks needed for the diagnostics you want installed, and then type this number (octal!). The Formatter will now assume that this disk is a system disk.

Table 11-1. AOS Disk Unit Names and Device Codes

Disk Model Number and Description	Default Device Code of Controller	Disk Number on Controller	Disk Unit Name(s)
4231A. A moving-head disk unit with a top-loading 92-Mbyte disk.	33	first (0) second (1) third (2) fourth (3)	DPE0 DPE1 DPE2 DPE3
	73	first (0) second (1) third (2) fourth (3)	DPE10 DPE11 DPE12 DPE13
6030. A moving-head diskette unit. The controller can run up to four 0.3-Mbyte diskette units. (Note that the same controller that runs a 6045 disk unit can run 6030 units.)	33	first (0) second (1) third (2) fourth (3)	DPD0 DPD1 DPD2 DPD4
	73	first (0) second (1) third (2) fourth (3)	DPD10 DPD11 DPD12 DPD13
6045. A dual moving-head disk unit that includes one 5-Mbyte fixed disk and one 5-Mbyte removable disk cartridge. (Note that the same controller that runs 6030 diskette units can run a 6045 disk unit.	33	first pair second pair third pair fourth pair	(Removable / Fixed) DPD0 / DPD4 DPD1 / DPD5 DPD2 / DPD6 DPD3 / DPD7
	73	first pair second pair third pair fourth pair	(Removable / Fixed) DPD10 / DPD14 DPD11 / DPD15 DPD12 / DPD16 DPD13 / DPD17
6060, 6061, 6067, and 6122; 6160 and 6161. All are moving-head disk units.	27	first (0) second (1) third (2) fourth (3)	DPF0 DPF1 DPF2 DPF3
The 6060, 6061, 6067, and 6122 are free-standing units with removable disks; a controller can run four units. The 6160 and 6161 are sealed units with bay-mounted fixed disks; a controller can run two units.	67	first (0) second (1) third (2) fourth (3)	DPF10 DPF11 DPF12 DPF13
A 6060 holds a 96-Mbyte disk; a 6061 holds a 190-Mbyte disk; a 6067 holds a 50-Mbyte disk; a 6122 holds a 277-Mbyte disk; a 6160 holds a 73-Mbyte disk; a 6161 holds a 147-Mbyte disk.	none; chosen at installation	first (0) second (1) third (2) fourth (3)	DPF20 DPF21 DPF22 DPF23
	none; chosen at installation	first (0) second (1) third (2) fourth (3)	DPF30 DPF31 DPF32 DPF33

(continues)

Table 11-1. AOS Disk Unit Names and Device Codes

Disk Model Number and Description	Default Device Code of Controller	Disk Number on Controller	Disk Unit Name(s)
6063, 6064, 6066. A bay- mounted, fixed-head disk. The 6063 holds 1 Mbyte disk; a 6064 holds a 2-Mbyte disk; a 6066 has two 6064 units and one controller.	26	first (0) second (1) third (2) fourth (3)	DKB0 DKB1 DKB2 DKB3
	66	first (0) second (1) third (2) fourth (3)	DKB10 DKB11 DKB12 DKB13
6070. A dual moving-head disk unit that includes one 10-Mbyte fixed disk cartridge and one 10-Mbyte removable disk cartridge.	33	first pair second pair third pair fourth pair	(Removable / Fixed) DPG0 / DPG4 DPG1 / DPG5 DPG2 / DPG6 DPG3 / DPG7
	73	first pair second pair third pair fourth pair	(Removable / Fixed) DPG10 / DPG14 DPG11 / DPG15 DPG12 / DPG16 DPG13 / DPG17
6098, 6099, 6100, 6103. A bay-mounted moving-head unit. May have a 1.26 Mbyte diskette on the same controller. Disk holds 12.5 or 25 Mbytes.	33	first(0)	DPI0 (diskette is DPI1)
	73	first (0)	DPI10 (diskette is DPI11)
6101, 6104, 6220-D, 6222-D. A bay-mounted, sealed unit, with a 1.26-Mbyte diskette. Disk capacity is 12.5, 25, 5, or 15 Mbytes respectively. They are designed for S/20s and other microECLIPSE systems.	26	first (0)	DPK0 (diskette is DPK1)
	66	first (0)	DPK10 (diskette is DPK11)
6102, 6105, 6220, 6222. A bay-mounted, sealed unit. Disk capacity is 12.5, 25, 5, or 15 Mbytes respectively.	26	first (0)	DPK0
	66	first (0)	DPK10
6224. A 15-Mbyte unit that runs on the burst multiplexor channel (BMC) instead of the data channel. It's designed for S/20 and other microECLIPSE systems.	25	first (0)	DPL0
	65	first (0)	DPL1
6225, 6227, 6234. A fixed, bay-mounted moving-head disk unit that holds 5, 15, or 50 Mbytes respectively.	33	first(0)	DPI0
	73	first (0)	DPI10

(continued)

Table 11-1. AOS Disk Unit Names and Device Codes

Disk Model Number and Description	Default Device Code of Controller	Disk Number on Controller	Disk Unit Name(s)
6225-D, 6227-D, 6234-D. A fixed, bay-mounted moving head disk unit that holds 5, 15, or 50 Mbytes respectively, with up to three 1.26-Mbyte diskettes on the same controller.	33	first(0)	DPI0
	73	first (0)	DPI10
6236 and 6237; 6239 and 6240 Each is a rack-mounted, sealed, moving-head unit with the power switch on the upper right. It has a LED display that shows the unit number and can show the current cylinder or disk fault code. A controller can run four units. Up to three units fit in a cabinet. A model 6236 unit holds 354 Mbytes; a model 6237 is three 6236 units in one cabinet, on one controller. A model 6239 unit holds 592 Mbytes; a model 6240 is three 6239 units in one cabinet, on one controller.	24	first (0) second (1) third (2) fourth (3)	DPJ0 DPJ1 DPJ2 DPJ3
	64	first (0) second (1) third (2) fourth (3)	DPJ10 DPJ11 DPJ12 DPJ13
	none; chosen at installation	first (0) second (1) third (2) fourth (3)	DPJ20 DPJ21 DPJ22 DPJ23
	none; chosen at installation	first (0) second (1) third (2) fourth (3)	DPJ30 DPJ31 DPJ32 DPJ33
6267 and 6268. A 5-1/4 inch diskette unit, capacity 368 Kbytes per diskette, used by DESKTOP GENERATION and other systems.	20	first (0) second (1)	DPM0 DPM1
	60	first (0) second (1)	DPM10 DPM11
6271. A compact, 15-Mbyte unit designed for DESKTOP GENERATION systems.	26	first (0) second (1)	DPN0 DPN1
6280. Same as model 6224, but holds 50 Mbytes instead of 15 Mbytes.	25	first (0)	DPL0
	65	first (0)	DPL10
6290. Two 6239 units in one cabinet on one controller.	see 6236		
6301. Same as model 6271, but holds 38.6 Mbytes instead of 15 Mbytes.	26	first (0) second (1)	DPN0 DPN1
6336. Same as model 6271, but holds 71.2 Mbytes instead of 15 Mbytes.	26	first (0) second (1)	DPN0 DPN1

(concluded)

The Disk Formatter now displays each disk number with octal start and end logical block addresses. The Formatter deals only with octal numbers (except that it accepts 8 as the number of the last physical disk in the LDU). For example, it might display

DISK NUMBER 1: 00000000000 THRU 00001325657

DISK NUMBER 2: 00001325660 THRU 00002653535

And it asks for the logical disk's unique ID.

LOGICAL DISK UNIQUE ID (1 TO 6 CHARS)?

AOS uses the logical disk unique ID to determine which physical disks belong to this LDU.

For simplicity, you might want to use an ID that is as close as possible to the LDU name, which you'll specify next. The LDU name will be the filename of the LDU. For example, if you plan to name the LDU UDD1, you would type UDD1↵. If you plan to name it DATABASE, you'd use a six-character abbreviation; e.g., DATABA↵.

Create an ID from one to six characters long, and type it in. Any filename character is legal: A through Z (uppercase and lowercase are treated the same), 0 through 9, period (.), dollar sign ($), question mark (?), and underscore (_). Each ID should be unique among LDUs.

(If you are creating a destination LDU for the PCOPY program, you may choose to give it the same ID and name as the source LDU, or not. LDU names and IDs are irrelevant to PCOPY. But PCOPY requires that a destination LDU's bitmap, overlay, and remap area addresses be the same as the source LDU's.)

Whatever ID you decide on, you will want to keep a record of it for future use.

After you type the ID, the Disk Formatter asks

LOGICAL DISK NAME (1 TO 31 CHARS)?

For any nonmaster LDU, the name you type here will be the filename of the LDU. This name will be displayed when you initialize the LDU from the CLI, or boot it (if a system disk).

People and AOS can use the LDU name just as any other directory pathname. An LDU's pathname is the directory it was initialized from. For example, an LDU named DATABASES, initialized from the root directory, would have the full pathname :DATABASES. If you need more background on this, read "Single- and Multiple-Disk LDUs," earlier in this chapter.

After deciding on the LDU name (1 to 31 filename characters), type it. The Disk Formatter asks

ACCESS CONTROL LIST
USER NAME OR TEMPLATE (1 TO 15 CHARS)?

AOS maintains an access control list (ACL) for each LDU directory. The ACL lists names or name templates for users who can access this directory and the type(s) of access each user or user group has. To create the ACL, you specify usernames and the access privileges you want these usernames to have.

Whatever ACL you specify is ignored if this LDU runs as the master LDU, because all users must have execute (E) access to the root directory.

If you don't know exactly which usernames and access privileges you want for this LDU, a good general-purpose ACL is OP,OWARE and +,E. This gives username OP all privileges and all users execute privileges. If you don't want to specify an ACL, press ↵ for the default, which is a null ACL.

Later, if needed, you can override the LDU ACL with the CLI command ACL after initializing the LDU; or you can change it with a Partial format.

Templates and LDU ACLs work the same way as templates and filename ACLs, described in Chapter 9, "Other Runtime Tools". But, for your convenience, we'll review templates and the ACL issue. If you don't want a review, skip the next section.

Username Templates and ACLs

You can specify a username literally (e.g., ADAM for a user who will log on as ADAM), or you can specify one or more username templates. A username template allows a group of users with a variety of usernames to access the LDU. A username template uses one or more template characters to symbolize other filename characters. The template characters are hyphen (−), plus (+), and asterisk (*).

− symbolizes any number of filename characters, except periods;

+ symbolizes any number of filename characters, including periods;

* symbolizes any single character, except a period.

For example, the template

*FORTRAN

matches username AFORTRAN, but not AMFORTRAN, nor .FORTRAN.

The template −FORTRAN

matches username BOBFORTRAN, but not BOBFORT, nor BOB.

The template FORTRAN+

matches username FORTRANA and FORTRAN.USER, but not .FORTRAN, nor AFORTRAN.

The template −FORTRAN+

uses two template characters and matches usernames BOBFORTRAN.XX and FORTRAN, but not FORT.RAN, nor .FORTRAN.

Each time you supply a username or template, the Disk Formatter asks about the user privileges. A user can have five types of access to a directory (which an LDU is); or a user can have no access privilege. The access privileges are O, W, A, R, or E. Table 11-2 explains the meanings of the different access control privileges.

Table 11-2. Access Control Privileges

Letter	A User Who Has	Can
O	Owner access	Initialize the LDU and change its ACL.
W	Write access	Insert and delete filenames in the LDU's primary directory or change its ACL.
A	Append access	Add new files to the LDU's primary directory, or change its access control list.
R	Read access	List the files in the LDU's primary directory.
E	Execute access	Use the LDU's name in a pathname.

You will often assign combinations of these privileges. For example, RE) allows the specified username(s) to use the LDU's name in a pathname (E) and list the files in its primary directory (R).

Specifying Usernames and Privileges

The Disk Formatter is asking for an

ACCESS CONTROL LIST
USER NAME OR TEMPLATE (1 TO 15 CHARS)?

Having decided on a username, type it in; for example, OP↵. The Disk Formatter then asks for the privileges.

PRIVILEGES (O, W, A, R, E, NEW-LINE)?

Decide on the privileges you want for the pertinent username(s) and type the answer; e.g., OWARE ↵. An answer of NEW-LINE (↵) explicitly gives the user no privileges; this is useful when you want to give everyone *except* a specific username/template access to an LDU. For example, assume that you want to give all usernames — except those beginning with $ — write access to an LDU. You'd give a username of + ↵ and privilege of W ↵; then you'd give a username of $+ ↵ and privilege of NEW-LINE (↵).

The Formatter repeats the *USER NAME...?* and *PRIVILEGES...?* questions, allowing you to set up quite a specific *ACL.*

USER NAME OR TEMPLATE (1 TO 15 CHARS)?

If you have another username to specify, type it; for example, +↵.

PRIVILEGES (O, W, A, R, E, NEW-LINE)?

Give the privileges for the username(s); e.g., E ↵.

When you press ↵ in response to *USER NAME...?*, the Formatter asks

SURFACE ANALYSIS?

This begins a series of questions about surface analysis. During analysis, the Disk Formatter writes a test pattern to and reads it from each disk block, then it compares the results. It records bad disk blocks in a bad block table that AOS keeps partially in memory. When AOS wants to access a block whose address is noted in this table, the table directs it to a good block in an area called *the remap area.* You can choose up to five test patterns.

A bad block can bring down AOS, so it's critically important that all bad blocks be recorded. You should run all five test patterns on each new disk, and whenever you suspect that a disk may have developed one or more new bad blocks.

If you don't want any patterns run on this disk, type N↵; the Formatter then skips to the *FOR DISK #n* bad block sequence, below.

If you want one or more of the checkout patterns run, type Y↵. Then, for a multiple-disk LDU, the Disk Formatter asks

DISK NUMBER?

To have patterns run on all disks in the LDU, press ↵. (You *can* specify numbers individually, if you want to *omit* patterns on a disk; e.g. 1↵, and 3↵; but omitting patterns on a disk isn't recommended.)

Now, the Formatter wants to know

YOU MAY RUN UP TO FIVE (5) PATTERNS...HOW MANY WOULD YOU LIKE TO RUN?

To ensure the validity of the disk structure, the Formatter writes to the disk a variety of bit patterns, reads from the disk, and compares the results. The more patterns you run, the higher the degree of validity. You can specify up to five bit patterns, which we recommend doing.

Type the number of test patterns that you want run on the disk. Table 11-3 shows the approximate time required for surface analysis on each type of disk supported by AOS.

Table 11-3. Surface Analysis Times for Disks

Disk Model	Capacity (Mbytes)	Approximate Time Per Test Pattern
6060	96	13 minutes
6061	190	26 minutes
6122	277	26 minutes
6160	73	11 minutes
6161	147	22 minutes
6227, 6271	15	10 minutes
6234	50	8 minutes
6236	354	50 minutes (approximately)
6063	1	3 minutes
6064, 6066	2	6 minutes per disk unit
6097 (diskette)	1.26	2 minutes
6070	10 per disk	14 minutes per disk
6045	5 per disk	8 minutes per disk
6096, 6099	12.5 (disk) 1.25 (diskette)	14 minutes for hard disk 2 minutes per diskette
6100, 6103	25 (disk) 1.25 (diskette)	28 minutes for hard disk 2 minutes per diskette
6030 (diskette)	0.3	2 minutes

The Disk Formatter requires no interaction as it runs the patterns, so you can leave the system console as they run.

The Disk Formatter now runs the pattern(s) on the current disk. As each pattern runs, the Formatter displays

--RUNNING PATTERN n

where *n* is the octal test pattern.

As the Formatter finishes the pattern(s) on each disk, it updates the disk's bad block table. Each disk that isn't analyzed retains its existing bad block table. If an existing table is invalid, the Formatter prints a message and zeros the bad block table. The Formatter issues an error message and terminates if it finds more than 126 bad blocks on one physical disk. All error messages are explained in Table 11-4.

When the Disk Formatter completes the surface analysis on this disk, it says

n BAD DISK BLOCKS

If there are bad blocks, it asks

PRINT BAD BLOCK STATISTICS [N]

You may want to see the bad block statistics. If so, type Y ↵; and the Formatter describes the address, cylinder, head, and sector of each bad block. You might want to note these for future reference. To skip the statistics, press ↵ or type N ↵.

Next, the Disk Formatter asks for any additional bad block numbers.

ADDITIONAL BAD BLOCK NUMBER (<NL> WHEN DONE):

Generally, on a Full format, you will have no additional bad blocks to enter; if this is true, press ↵ and skip to the *BITMAP...* questions. If you want to enter one or more bad blocks, type each block's address. You can do it by typing the logical block number (in octal); or you can type the disk-number, cylinder, head, and sector numbers of each block in this format:

disk-number:cylinder-number,head-number,sector-number

(Omit the head number for a bad block in a fixed-head disk). Press ↵ to signal the end of your bad block entries. Now the Disk Formatter asks if it should print the updated bad block statistics.

PRINT BAD BLOCK STATISTICS? [N]

Press ↵ or answer Y ↵. If you answer Y ↵, the Formatter displays the updated bad block statistics. Then, if there is another disk in the LDU, the Formatter returns to run patterns (if specified) on it; then, it runs through the bad block statistics again.

When the Formatter has asked about bad block tables for all disks in the LDU, it displays the bitmap size and asks about the bitmap address:

BITMAP SIZE: n

BITMAP ADDRESS? [default]

The default address (displayed in brackets) is 3/8 of the distance across the first disk in the LDU. This is a good general-purpose choice, and we recommend it. To select the default, press ↵.

There may be a reason to start the bitmap at the beginning of the LDU (described near the beginning of this chapter). If you know that you want the bitmap at the beginning of the LDU, type 0 ↵.

If the bit map area you specify or default contains bad blocks, the Disk Formatter issues a warning and asks you to confirm the address. Do *not* confirm it. Instead, type N↵; then add 100 octal to the original bitmap address and try again. Keep doing this until the Formatter accepts your answer.

The Disk Formatter then writes the bitmap to the LDU. Then it asks

SYSTEM DISK? [Y]

Your answer to this determines whether or not the Formatter will reserve space on this LDU for an AOS system bootstrap and system overlay area.

Your first LDU already holds an AOS system. Other LDUs need not include space for a system. But, since space for a system represents much less than 1% of the total disk space (on the smallest of the DPF disks), you can generally default the answer to yes. Of course, if you're formatting a diskette or if you know that this disk will never hold a system, answer no (N↵); the Formatter then skips the next two questions.

You may want to format a diskette as a system disk in order to boot such stand-alone utilities as DFMTR, PCOPY, and FIXUP. If so, you should specify zero overlay area and a very small remap area. Doing this will give you as much space on the diskette as possible.

If you answer ↵ (or Y↵), the Formatter asks you to specify the

OVERLAY AREA SIZE? [default]

The overlay area is reserved for AOS overlays — parts of the system that AOS needs to perform some system operations. The default size (for AOS revision 5.00) is 620 (octal) blocks, 400 decimal.

Generally, you should default this question by pressing ↵. The Formatter then asks for the

OVERLAY AREA ADDRESS? [default]

The default address is as close as possible to 3/8 of the distance across the first disk in the LDU. (It immediately follows the bitmap, if you defaulted the bitmap address.) If you defaulted the bitmap address, you should default the overlay area address. To do so, press ↵.

If you put the bitmap area at the beginning of the LDU, then the overlay area should follow the bitmap. To do this, type the size that the Formatter gave for the bitmap (e.g., 133 ↵ for a 190-megabyte LDU).

If the overlay area that you default or specify contains a bad block, the Disk Formatter will issue a warning and ask you to confirm your response. Don't confirm; performance will suffer if AOS has to go all the way to the remap area to pick up an overlay. Instead, type N↵. Then add 100 octal to the original address and type this value. Repeat until the Formatter accepts your answer without a warning.

Finally, the Disk Formatter asks you to specify the size and starting address of remap areas for each physical disk in the LDU.

DISK NUMBER n REMAP AREA SIZE? [default]

Generally, you should default this question by pressing ↵. The Disk Formatter then asks for the

DISK NUMBER n REMAP AREA ADDRESS? [default]

To take the default, press ↵. If you take the default, the Disk Formatter uses the first free, 126-block area on this physical disk as the disk's remap area. Generally, you should take the default unless there are a lot of bad blocks.

For a large, multiple-disk LDU that will hold big contiguous files, you should put the first disk's remap area at the beginning of the first physical disk (default), and put the the last disk's area at the end of the last physical disk.

For every bad block AOS needs to access, it must go to the remap area to find a substitute. If this disk has a lot of bad blocks, you may want to put the remap area in the middle of the disk. To do it on a single-disk LDU, specify the address right after the bitmap and overlay area (if this LDU is formatted as a system disk).

The Disk Formatter does not allow bad blocks in the remap area. If you specify or default an area containing bad blocks, the Formatter issues an error message and repeats the question. As above, add 100 octal to the default or your specified value and try again. Repeat until the Formatter accepts your answer.

This is the last question; the Disk Formatter now says

LOGICAL DISK CREATED

DONE!

From the stand-alone Formatter, you return to the *SCP-CLI>*, the *!* prompt, or nothing (if switches). If you want to format another LDU, type CONTINUE↵ on MV/8000, P↵ on other programmed consoles, or press the CONTINUE switch if you have a machine with data switches and return to the beginning of this section. Or, you might want to install an AOS system on it, as described in the next chapter.

From the stand-among Formatter, you return to the AOS CLI. To format another LDU, type XEQ DFMTR↵ again and return to the beginning of this section. Or, you might want to install an AOS system on it, as described in the next chapter.

In any case, if the disks in the LDU involve one or more new disk controllers, you will want to run AOSGEN and create a tailored system to support the new controller(s). If your current tailored system supports the new LDU's controller(s), you might try out the new LDU from AOS (using the CLI command INITIALIZE).

If you ever need to change the name, ACL, or bitmap, overlay, or remap areas on this LDU, or to identify new bad blocks in it, you can run a Disk Formatter Partial on it, as described in the next section.

Note that whenever you build an LDU, be sure to write the LDU ID and name on a paper label, and stick it on the disk cover or unit. If the disk is part of a multiple-disk LDU, this label should also describe the other disks in order, and sequence number of this disk. For example, the label might say

```
LDU ID is:  UDD1     LDU name is:   UDD1
This LDU includes three model 6061 disks.
This is disk number:  2
LDU created:  14 March 1985
```

Without the label, it's easy to lose track of disk IDs/names and sequence numbers — especially on removable disks.

Figure 11-2 shows a sample Disk Formatter Full dialog, for a two-disk LDU, running on an MV/8000.

```
SCP-CLI>   RESET ↵     (For stand-among Disk Formatter, this is XEQ DFMTR ↵; and skip to AOS
                        DISK FORMATTER banner.)

SCP-CLI>   BOOT 27 ↵

SPECIFY EACH DISK IN THE LDU
DISK UNIT NAME?    DPF0 ↵        (disk unit name)
DEVICE CODE?    ↵

SYSTEM PATHNAME?    DFMTR ↵

AOS DISK FORMATTER REV n
FULL FORMAT DESTROYS ANY AOS DISK STRUCTURE, PARTIAL RETAINS IT.

FULL (F) OR PARTIAL (P OR <NL>)?     F ↵

FULL FORMAT

ENTER UNIT NAMES FOR EACH UNIT IN THE LDU ( <NL> WHEN DONE):
DISK UNIT NAME?    DPF10 ↵
DEVICE CODE?    ↵
DISK UNIT NAME?    DPF11 ↵
DEVICE CODE?    ↵
DISK UNIT NAME?    ↵

DO YOU WANT TO ALLOCATE A DIAGNOSTIC AREA? [N] ↵
DISK NUMBER 1: 00000000000 THRU 00001325657
DISK NUMBER 2: 00001325660 THRU 00002653535

LOGICAL DISK UNIQUE ID (1 TO 6 CHARS)?    UDD1 ↵
LOGICAL DISK NAME (1 TO 31 CHARS)?    UDD1 ↵

ACCESS CONTROL LIST
USER NAME OR TEMPLATE (1 TO 15 CHARS)?    OP ↵
PRIVILEGES (O, W, A, R, E, NEW-LINE) ?    OWARE ↵
USER NAME OR TEMPLATE (1 TO 15 CHARS)?    + ↵
PRIVILEGES (O, W, A, R, E, NEW-LINE)?    E ↵
USER NAME OR TEMPLATE (1 TO 15 CHARS)?    ↵
SURFACE ANALYSIS? [N]    Y ↵
DISK NUMBER?    ↵

YOU MAY RUN UP TO...    3 ↵
ANALYZING DISK #1
—RUNNING PATTERN 155555
—RUNNING PATTERN 133333
—RUNNING PATTERN 066666
0 BAD BLOCKS
ADDITIONAL BAD BLOCK NUMBER: (<NL> WHEN DONE):    ↵
0 BAD BLOCKS
ANALYZING DISK #2
—RUNNING PATTERN 155555
—RUNNING PATTERN 133333
—RUNNING PATTERN 066666
0 BAD BLOCKS
ADDITIONAL BAD BLOCK NUMBER (<NL> WHEN DONE ):    ↵
0 BAD BLOCKS
```

DG-26914T

Figure 11-2. Full Disk Formatter Dialog for a Two-Disk LDU (MV/8000) (continues)

```
BITMAP SIZE: 266
BITMAP ADDRESS [00000417753]  ↵
SYSTEM DISK: [Y]  ↵
OVERLAY AREA SIZE? [000620]  ↵
OVERLAY AREA ADDRESS? [00000420141]  ↵
DISK NUMBER 1 REMAP AREA SIZE? [000176]  ↵
DISK NUMBER 1 REMAP AREA ADDRESS? [000174]  ↵
DISK NUMBER 2 REMAP AREA SIZE? [000176]  ↵
DISK NUMBER 2 REMAP AREA ADDRESS? [1325660]  ↵

LOGICAL DISK CREATED
DONE!
.
.
.
SCP-CLI>    (for stand-alone Disk Formatter)

or

)           (for stand-among Disk Formatter)
```

DG-26914T

Figure 11-2. Full Disk Formatter Dialog for a Two-Disk LDU (MV/8000) (concluded)

The Partial Format

When you start up the Disk Formatter (by specifying pathname DFMTR↵ for the stand-alone version, or by typing XEQ DFMTR↵ for the stand-among version), it says

AOS DISK FORMATTER REV n

FULL FORMAT DESTROYS ANY AOS DISK STRUCTURE, PARTIAL RETAINS IT.

FULL (F) OR PARTIAL (P OR <NL>)?

For a Partial format, type

P ↵

PARTIAL FORMAT

ENTER UNIT NAMES FOR EACH UNIT IN THE LDU (<NL> WHEN DONE):

DISK UNIT NAME?

Enter the name of the disk unit that holds the first disk in the LDU. Table 11-1 (earlier) shows the disk unit names and device codes of all disks supported by AOS. For example,

DPF1 ↵

The stand-among Formatter knows — via AOS — the device code of the disk unit. So it skips the next question and asks for another *DISK UNIT NAME?*.

But the stand-alone Formatter asks

DEVICE CODE?

Type the device code of the disk unit you just specified. Default device codes are 27 and 67 for the first and second DPF-type disk controllers, 33 and 73 for DPI-type and all other moving-head models, and 26 and 66 for fixed-head controllers. The device codes are shown in Table 11-1, earlier. If the disk unit you specified is on the default device code for its controller, you can press ⤶ in response to the *DEVICE CODE?* query.

The Formatter repeats the question(s) *DISK UNIT NAME?* (and *DEVICE CODE* if stand-alone) so that you can specify all physical disks that were fully formatted into the LDU. When you have identified them all, press ⤶ in response to *DISK UNIT NAME?*

The Disk Formatter now displays each disk number with octal start and end logical block addresses. The Formatter deals only with octal numbers (except that it accepts 8 as the number of the last physical disk in the LDU). For example, it might display

DISK NUMBER 1: 00000000000 THRU 00001325657

DISK NUMBER 2: 00001325660 THRU 00002653535

And it asks if you want to change the logical disk's unique identification (ID).

NEW LOGICAL DISK UNIQUE ID (1 TO 6 CHARS)? [default]

The *[default]* is the value assigned when this LDU was created via a Full format run. If you are satisfied with the default, press ⤶; the Formatter then goes on to the next question.

The ID can be from 1 to 6 characters long; any filename character is legal. Each ID should be unique among all your LDUs. (If you want this LDU to serve as a destination LDU for the PCOPY program, it can — but need not — have the same ID and name as the source LDU. LDU names and IDs are irrelevant to PCOPY. But PCOPY requires that a destination LDU's bitmap, overlay, and remap area addresses be the same as the source LDU's.)

For simplicity, you may want to use an ID that is as close as possible to the LDU name (if you plan to change the current name). After choosing an ID, type it and ⤶.

After you type or default the ID, the Disk Formatter asks whether you want a

NEW LOGICAL DISK NAME (1 TO 31 CHARS)? [default]

As with the ID, if you are satisfied with the default, press ⤶ and skip to the next question.

If you want to change the LDU name, recall that for any nonmaster LDU, the LDU name is its filename. This name will be displayed when you initialize the LDU from the CLI, or boot it (if a system LDU). People and AOS can use the LDU name just as any other directory pathname. An LDU's pathname incorporates the directory it was initialized from. For example, an LDU named DATABASES, initialized from the root directory, would have the full pathname :DATABASES. If you need more background on this, see "Single- and Multiple-Disk LDUs," earlier in this chapter.

After deciding on the LDU name (1 to 31 filename characters), type it.

Next, the Disk Formatter asks

NEW ACCESS CONTROL LIST? [N]

If you don't want to change the current LDU access control list (ACL), press ⤶; the Formatter then skips to the *READ ONLY SURFACE ANALYSIS?* question.

To change the ACL, type Y⌋. The Formatter will then ask for the username (or template), then the privileges for that username. It allows up to seven username and privileges specifications. It will repeat the two questions until you press ⌋ in response to *USERNAME...?*, or until you specify seven usernames. For example,

Y ⌋
USER NAME OR TEMPLATE (1 TO 15 CHARS)? OP ⌋
PRIVILEGES (O,W,A,R,E)? OWARE ⌋
USER NAME OR TEMPLATE (1 TO 15 CHARS)? SARAH ⌋
PRIVILEGES (O,W,A,R,E)? ARE ⌋
USER NAME OR TEMPLATE (1 TO 15 CHARS)? + ⌋
PRIVILEGES (O,W,A,R,E)? E ⌋
USER NAME OR TEMPLATE (1 TO 15 CHARS)? ⌋

This gives OP all access to the LDU primary directory, gives all users Execute access, and gives user SARAH Append, Read, and Execute access. The username/template and privileges are explained earlier in this chapter.

If needed, you can override the LDU ACL given here with the CLI command ACL after you initialize the LDU.

After you specify or default the access control list, the Disk Formatter asks

READ ONLY SURFACE ANALYSIS? [N]

Read only surface analysis involves reading each disk block. This is not as thorough a test as the write/read sequence used in a Full format. To do it, the Formatter requires about half as much time as to run a single test pattern on all disks in the LDU (times are shown in Table 11-3).

If you press ⌋, the Formatter goes to the *PRINT BAD BLOCKS...?* question. If you answer Y⌋, the Formatter asks for the

DISK NUMBER?

You can press ⌋ to have analysis done on all disks; the Formatter then proceeds. Or, you can specify the number of each disk on which you want the analysis done; for example 1⌋, 2⌋, etc. If you specify numbers, press ⌋ in response to this question when you have specified all the disk numbers you want.

The Formatter now does the read analysis. If it cannot read a block, it notes the block as bad.

The Formatter now describes the bad blocks for this disk:

FOR DISK #n
n BAD DISK BLOCKS

If there are any bad blocks, it asks

PRINT BAD BLOCK STATISTICS? [N]

To see the details on old and new bad blocks, type Y ⌋. To skip the details, press ⌋; the Formatter then summarizes old and new bad blocks and skips to *UPDATE BAD BLOCK TABLE.* A sample bad block display looks like this.

  ADDR: 00000574537 CYL: 000653 HEAD: 000010 SECT: 000017
  ADDR: 00000747566 CYL: 001043 HEAD: 000014 SECT: 000006
* *ADDR: 00000747570 CYL: 001043 HEAD: 000014 SECT: 000006*

* *NEW BAD BLOCK WHICH WAS NOT ALLOCATED.*
** *NEW BAD BLOCK WHICH WAS ALLOCATED (MUST RUN FIXUP).*

If n NEW ALLOCATED BAD BLOCKS is not 0, one or more new bad blocks have developed in AOS file(s); someone must run FIXUP on the LDU to restore its integrity.

But if the Formatter finds *many* new bad blocks (allocated or unallocated) during read-only analysis, this may mean disk hardware problems. In such cases, you might want to abort the Formatter with CTRL-C CTRL-B (stand-among) or the break sequence (stand-alone) followed by RESET↓ (MV/8000), or an I↓ (other programmed consoles), or press the RESET switch (data switches); then consider running diagnostics on the disk unit with a different pack (if possible). After all, you don't want a whole slew of bad blocks noted on the disk if the disk is fine and the unit *heads* are out of alignment.

Next, the Disk Formatter asks if you want to

UPDATE BAD BLOCK TABLE? [N]

Press ↓ if you don't have any new bad blocks to enter, or if you don't want the Disk Formatter to enter any bad blocks it found during the read-only analysis. If you press ↓, the Formatter skips to the *BITMAP...* questions.

Generally, if the Formatter found only a few new bad blocks, you should answer Y↓. The Formatter then enters the new bad blocks it found; and it asks you to enter other bad blocks.

ADDITIONAL BAD BLOCK NUMBER (<NL> WHEN DONE):

Specify each bad block by typing its logical block number (in octal); or type the disk number, cylinder, head, and sector numbers of each block in this format:

disk-number,cylinder-number,head-number,sector-number

(Omit a head number for a bad block on a fixed-head disk.) Press ↓ to signal the end of your bad block entries.

Now the Disk Formatter asks if it should print the updated bad block statistics:

PRINT BAD BLOCK STATISTICS? [N]

Press ↓ or type Y ↓. If you answer Y ↓, the Formatter displays the updated bad block statistics. Then, if there is another disk in the LDU, it does the read-only analysis (if specified) and runs through the bad block sequence again.

When the Formatter has asked about bad block information for all disks in the LDU, it displays the LDU's bitmap size and asks about the bitmap address:

BITMAP SIZE: n

NEW BITMAP ADDRESS? [default]

There is one bitmap for all physical disks in the LDU.

You can always take the default bitmap address, given when this LDU was fully formatted, by pressing ↓.

With AOS revision 4.20, the Formatter defaults for the bitmap and overlay area addresses changed from the beginning of the LDU to 3/8 the distance across the first disk in the LDU. The new defaults generally yield better performance than the old ones. So, if the LDU you are formatting was created with a Formatter of revision 4.20 or later, and the bitmap and overlay area addresses were defaulted, you should usually not change these addresses; press ↓ to retain the bitmap address.

If the LDU was created by an earlier Formatter, with the bitmap and overlay areas at the beginning of the disk, you might want to move these areas to a better place. To do this, divide the number of blocks in the *first* disk by 2 (octal arithmetic), and specify this address as the bitmap starting address. To divide, use the last logical address given by the Formatter earlier (n in *DISK NUMBER 1, s THRU n).*

If — for reasons described near the beginning of this chapter — you want to have the bitmap at the beginning of the LDU, type 0 ↓.

If the bitmap area you specify or default contains bad blocks, the Disk Formatter issues a warning and asks you to confirm the address. Do not confirm; type N ↵. If the area has AOS files in it, the Formatter will say *SPECIFIED AREA ALREADY ALLOCATED.* For either error, add 100 octal to the address typed and try again. Keep doing this until the Formatter accepts your answer.

Depending on your answer, the Formatter leaves the bitmap in its original spot, or moves it to a new one.

If the LDU was formatted as a system disk, the Formatter skips the next question. Otherwise, it asks

ALLOCATE A SYSTEM BOOTSTRAP AREA ? [N]

This question allows you to set up the LDU as a system disk, from which you can boot and run an AOS system.

If you answer Y ↵, the Disk Formatter will reserve a 126-block area for SYSBOOT. It will also ask about overlay area size and address. To be able to boot and run AOS from this LDU, you must answer Y ↵ to this question, and specify the correct overlay area size to the next question. (The size is 620 octal for AOS revision 5.00 and later.) Then you must use the Installer to install a disk bootstrap, system bootstrap, and system on the LDU. Next, from AOS you must copy the CLI, GHOST, and peripheral manager files to the LDU (CLI MOVE or COPY commands). Thereafter, if this LDU includes a disk on unit 0, you will be able to boot and run AOS on it as if it were your original LDU.

If you don't want to be able to boot and run AOS from this LDU, press ↵ and skip to the *REMAP AREA* questions.

NEW OVERLAY AREA SIZE [default]

Generally, you should take the default in this by pressing ↵. But if the default is 0, and you want to set up the LDU as a system disk, type the appropriate value, described three paragraphs back, and ↵.

NEW OVERLAY AREA ADDRESS [default]

To take the default, given when the LDU was fully formatted, press ↵. (But if you moved the bitmap in the previous step, move the overlay area to start immediately after the bitmap. To do this, add the bitmap size given by the Formatter to the bitmap address you typed — octal arithmetic — and type this number.)

As with the bitmap, if the overlay area you specify or default contains bad blocks, the Disk Formatter issues a warning and asks you to confirm the address; type N↵. If the area has AOS files in it, the Formatter will say *SPECIFIED AREA ALREADY ALLOCATED.* For either error, add 100 octal to the original bitmap address and try again. Keep doing this until the Formatter accepts your answer.

Depending on your answer, the Formatter leaves the overlay area in its original spot, or moves it to the new one.

Finally, the Disk Formatter asks you about the remap areas on each disk in the LDU.

NEW DISK NUMBER n REMAP AREA SIZE? [default]

Generally, you should default this question by pressing ↵. But note the size if you would like to specify a nondefault address for the remap area. The Formatter then asks for the

NEW DISK NUMBER n REMAP AREA ADDRESS? [default]

To take the default, given when the LDU was fully formatted, press ↵. Generally, you should take the default unless there are a lot of bad blocks. If you press ↵; the Formatter then announces *DONE* and stops.

If this disk has a lot of bad blocks, you might want to move the remap area to the middle of the disk. To do it, divide the total number of blocks on the disk by 2 (octal arithmetic), and type this number.

The Disk Formatter does not allow bad blocks in the remap area. If you specify or default an area containing bad blocks, the Formatter issues an error message and repeats the question. As above, add 100 octal to the default or your specified value and try again. Repeat until the Formatter accepts your answer.

* This is the last question; the Disk Formatter now says

LOGICAL DISK CREATED

DONE!

From the stand-alone Formatter, you return to the *SCP-CLI>* or *!* prompt, or nothing (if switches). If you want to format another LDU, type CONTINUE) on MV/8000, or P) on other programmed consoles, or press the CONTINUE switch on machines with data switches and return to the beginning of this section. Otherwise, you might want to bring up AOS and try out your new LDU. Or, you might want to install an AOS system on it, as described in the next chapter.

From the stand-among Formatter, you return to the AOS CLI. To format another LDU, type XEQ DFMTR) again and return to the beginning of this section. Otherwise, you might want to try your new LDU with the INITIALIZE command. Or, you might want to install an AOS system on it, as described in the next chapter.

Note that whenever you change an LDU's name or unique ID, be sure to write the new ID and/or name on the paper label attached to the disk cover or unit. If the disk is part of a multiple-disk LDU, this label should also describe the other disks, and the LDU sequence number. Without the label, it's easy to lose track of disk IDs/names and sequence numbers.

Figure 11-3 shows a sample Disk Formatter Partial dialog, running on an MV/8000, which changes the ID, name, and ACL, and bitmap address on the LDU in unit DPF1.

```
SCP-CLI>   RESET ↵        (For stand-alone Disk Formatter, this is XEQ  DFMTR ↵; and skip to AOS
                           DISK FORMATTER banner.)

SCP-CLI>   BOOT 27 ↵      (Or BOOT 33 ↵)

SPECIFY EACH DISK IN THE LDU
DISK UNIT NAME?    DPF0 ↵
DEVICE CODE?    ↵

SYSTEM PATHNAME?    DFMTR ↵

AOS DISK FORMATTER REV n
FULL FORMAT DESTROYS ANY AOS DISK STRUCTURE, PARTIAL RETAINS IT.

FULL (F) OR PARTIAL (P OR <NL>)?    ↵

PARTIAL FORMAT

ENTER UNIT NAMES FOR EACH UNIT IN THE LDU ( <NL> WHEN DONE):
DISK UNIT NAME?    DPF1 ↵
DEVICE CODE?    ↵
DISK UNIT NAME?    ↵

DISK NUMBER 1: 00000000000 THRU 00002042416

NEW LOGICAL DISK UNIQUE I.D. (1 TO 6 CHARS)? [VULCAN]    UDD2 ↵
NEW LOGICAL DISK NAME (1 TO 31 CHARS)? [VULCAN]    UDD2 ↵

NEW ACCESS CONTROL LIST? [N]    Y ↵
USER NAME OR TEMPLATE (1 TO 15 CHARS)?    OP ↵
PRIVILEGES (O, W, A, R, E, NEW-LINE)?    OWARE ↵
USER NAME OR TEMPLATE (1 TO 15 CHARS)?    + ↵
PRIVILEGES (O, W, A, R, E, NEW-LINE)?    RE ↵
USER NAME OR TEMPLATE (1 TO 15 CHARS)?    ↵
READ ONLY SURFACE ANALYSIS? [N]    Y ↵

DISK NUMBER?    ↵
ANALYZING DISK #1

2 BAD BLOCKS
PRINT BAD BLOCK STATISTICS? [N]    Y ↵

** ADDR: 00000574537 CYL: 000653 HEAD: 000010 SECT: 000017
* ADDR: 00000747566 CYL: 001043 HEAD: 000014 SECT: 000006

* NEW BAD BLOCK WHICH WAS NOT ALLOCATED.
** NEW BAD BLOCK WHICH WAS ALLOCATED (MUST RUN FIXUP).

1 OLD BAD BLOCKS
0 NEW UNALLOCATED BAD BLOCKS
1 NEW ALLOCATED BAD BLOCKS

UPDATE BAD BLOCK TABLE? [N]    ↵
ADDITIONAL BAD BLOCK NUMBER (<NL> WHEN DONE ):    ↵
2 BAD BLOCKS
PRINT BAD BLOCK STATISTICS? [N]    ↵
```

DG-26915T

Figure 11-3. Partial Disk Formatter Dialog for a Single-disk LDU (MV/8000) (continues)

```
BITMAP SIZE: 205
NEW BITMAP ADDRESS [00000000000]        1021200 ↵

NEW OVERLAY AREA SIZE? [000620]      ↵
NEW OVERLAY AREA ADDRESS? [00000000174]      1021600 ↵

SPECIFIED AREA ALREADY ALLOCATED
NEW OVERLAY AREA ADDRESS? [00000000174]      1022100 ↵

NEW DISK NUMBER 1 REMAP AREA SIZE? [000176]      ↵
NEW DISK NUMBER 1 REMAP AREA ADDRESS? [000174]      ↵

LOGICAL DISK CREATED
A BAD BLOCK WAS ALLOCATED, MUST RUN FIXUP
DONE!
```

Run FIXUP on the LDU.

DG-26915T

Figure 11-3. Partial Disk Formatter Dialog for a Single-disk LDU (MV/8000) (concluded)

Disk Formatter Error Messages

The error messages that you might receive from the Disk Formatter appear alphabetically in Table 11-4.

Table 11-4. Disk Formatter Error Messages

Message	Meaning and Action
(nothing)	You may have specified a nonexistent disk unit. Wait a few seconds. If nothing happens, enter the break sequence (CMD and BREAK, or BRK, or BREAK keys); RESET↲, BOOT n↲, and try again.
A BAD BLOCK WAS ALLOCATED, MUST RUN FIXUP	One or more of the disk blocks that belongs to an AOS file is bad. The block is "part" of the file. The Disk Formatter has assigned the block to the bad block table, but you must run FIXUP on the disk to correct the file structure. This message occurs on a Partial format only.
BAD BLOCK IN INVISIBLE SPACE, CAN'T FORMAT DISK	One of the first eight blocks on the disk is bad; the disk is unusable in its current state. The problem may be head alignment or a flawed disk. DTOS disk diagnostics or other recovery action may be needed. Call your DG support organization.
BAD BLOCK IN REMAP AREA	The remap area you specified (or defaulted) contains a bad block. Add 100 octal to the REMAP AREA ADDRESS specified or defaulted; then type this number as the remap address. If this doesn't work, try it again.
CAN'T EXPAND TO SPECIFIED AREA	AOS files occupy some (or all) of the area specified. See the message *SPECIFIED AREA ALREADY ALLOCATED* in this table for action.
CAN'T OPEN DISK, ERROR CODE n	The Disk Formatter can't open the disk for I/O. Code *n* is an AOS error code, which the CLI will interpret for you if you type MESSAGE n↲. If the code is 000063, it means device already in use; this means that you tried to run the Formatter on a disk that is part of the master LDU, or on a disk that was grafted onto the current LDU via the CLI command INITIALIZE. If the former, you must run the *stand-alone* Disk Formatter. If the latter, RELEASE the disk using its LDU name from the CLI, then restart the Disk Formatter.
DISK ERROR DEVICE d u STATUS= s	The Disk Formatter encountered a disk block that it cannot read or write. It returns to the *DISK UNIT NAME?* query for this disk. Make sure the disk is write-enabled; then restart the Formatter run. If the problem recurs, DTOS disk diagnostics may be needed; contact your DG support organization. The *d* is the device code; the *u* is the unit number; and *s* is the hardware status code. You can check *s* in the Peripherals manual (Preface). In any case, record *s* for your DG support organization when you report this problem.
FATAL DISK ERROR DEVICE d u STATUS s	The Disk Formatter encountered an unrecoverable error while trying to read or write a disk block. The problem may be hardware failure. DTOS diagnostics may be needed. Follow the recovery action suggested for *DISK ERROR*, above.

(continues)

Table 11-4. Disk Formatter Error Messages

Message	Meaning and Action
INCONSISTENT DIB INFO, DISK FORMAT INCORRECT	During a Full format, the Disk Formatter writes a Disk Information Block (DIB) to each physical disk in the LDU. The DIB contains the LDU unique ID, LDU name, sequence number, and name and sequence number of all disks in the LDU. This message (which appears on a Partial format only) can mean one of the following things. First, it may mean that the disk you just specified doesn't belong to the LDU at all. Check via the paper disk labels; then, mount the proper disk(s) or specify the proper unit numbers and retry. Second, it may mean that the *sequence* in which you specified the disk is wrong. For example, you typed the unit name of the disk formatted as the *second* disk first. Try changing the order in which you specify the disks. Third, it may mean that the revision and/or sequence numbers are invalid. This can mean that the DIB was damaged; if so, you will need to run a Full format on the disk(s).
INVALID BAD BLOCK TABLE	The bad block table is not valid. The table may have been overwritten. The Formatter zeros the current table. If the disk contains valuable files, see if AOS can access it; if so, DUMP all its files. Then run a Formatter Full command on the disk, specifying at least one pattern; and reload the files.
INVALID DISK NUMBER	The LDU you're working with does not contain the specified disk number. Retry.
NO SPACE FOR NAME OR ACL BLOCK	The LDU lacks a free block for name or ACL information. This may be a hardware problem; the disk may need DTOS diagnostics.
NO SPACE FOR SYSTEM BOOTSTRAP	The system bootstrap (SYSBOOT) requires 124 blocks in the first disk in the LDU. The Formatter can't find this space. If the disk is really full, dump and reformat it; then reload the files and try again.
NOT ENOUGH CONTIGUOUS SPACE ON DISK FOR SPECIFIED AREA	The contiguous disk space needed is not available. Add 100 octal to the area start address and retry. If this fails a few times, reboot AOS, and type F/PACKET↲ from its master directory. This shows logical disk addresses in entries 22 and 23 (octal). Files with lower logical addresses are earlier on the disk. Dump and delete some of them, run the Formatter again, then reload the files.
SPECIFIED ADDRESS IS OUT OF RANGE	The LDU does not contain the specified address. This error message may also indicate that a disk in the LDU lacks a remap area. Try another address; check all disks in the LDU with a Formatter Partial run.
SPECIFIED AREA ALREADY ALLOCATED	AOS files, or one of the special areas (bitmap, etc.) occupy part or all of this area. Add 100 octal to the disk address shown for the item (bitmap address, etc.) and type this figure. Repeat this until the Formatter accepts your answer.

(concluded)

Table 11-4. Disk Formatter Error Messages

Message	Meaning and Action
SPECIFIED AREA HAS A BAD BLOCK IN IT, CONFIRM	The area you specified (or defaulted) has a bad block in it. Do not confirm; type N↵. Add 100 octal to the specified (or default) area and try again. Repeat this until the Formatter accepts your answer.
TOO MANY BAD DISK BLOCKS	The Disk Formatter found more than 126 bad blocks on this disk while performing surface analysis. Retry from the beginning. If the message recurs during the second surface analysis, DTOS diagnostics may be needed.
TOO MANY PHYSICAL UNITS IN LDU	There cannot be more than eight physical disks in an LDU. Retry. Do not enter more than eight physical disks per LDU.

(concluded)

What Next?

This chapter gave some background on creating and using LDUs, then it described the Full and Partial formats of the stand-alone and stand-among Disk Formatter.

If the LDU(s) that you built involve one or more new disk controllers, you will need to run AOSGEN and generate an AOS system that supports the controller(s). If not, you might want to install a system on the new LDUs, described in the next chapter. Or, if you are ready, try running the new LDU(s) from AOS — and perhaps put the CLI INITIALIZE commands for them in the UP.CLI macro.

End of Chapter

Chapter 12
The Installer

Read this chapter

- when you want to understand what the Installer does;
- when you want to install an AOS disk bootstrap, system bootstrap, or operating system on a logical disk unit (LDU).

The Installer is a utility program that installs a disk bootstrap, system bootstrap (SYSBOOT), and/or an AOS operating system on an LDU. If you brought up your own first system (Chapter 2 or 3), you already have some experience with the Installer. This chapter tells the rest, in the following major sections:

- Some Background
- About the Installer
- If You Make a Mistake
- The Stand-Alone Installer
- The Stand-Among Installer
- Installer Error Messages
- What Next?

Some Background

Three AOS-based programs are needed to bring up an AOS operating system from disk. The first, called the disk bootstrap, is on the first few blocks of an LDU. The second, the system bootstrap (SYSBOOT), is in the next hundred or so blocks. Finally, the AOS system itself is somewhere on the disk, often in directory :SYSGEN.

When you boot AOS from an LDU, the following things happen.

- the hardware reads the disk bootstrap from the LDU into main processor memory;
- the disk bootstrap executes and reads SYSBOOT into memory;
- SYSBOOT executes, then asks for the system pathname;
- you specify or default an existing AOS system name;
- SYSBOOT copies the disk-based part of the AOS system to the overlay area reserved for it by the Disk Formatter;
- SYSBOOT loads AOS into memory, and
- the AOS system executes.

The Installer is the program that writes the disk and system bootstraps — and, optionally, an AOS system — to the LDU. An AOS system *need not* be installed because SYSBOOT can find it, using the pathname you specify.

When you install an AOS system, the Installer copies it to the LDU as a standard disk file. Then, it creates a pointer to the system name in "invisible" space at the beginning of the LDU. This pointer allows you to specify the AOS system by simply pressing ⤶ when SYSBOOT asks for the *SYSTEM PATHNAME?* SYSBOOT reads the pointer, copies the system as above, and executes it. The pointer is invisible to AOS, so the installed AOS system cannot be deleted or accessed in any way from AOS.

AOS Installation Tradeoffs

Installation is mandatory for the first AOS system. But you can install a tailored AOS system or leave it uninstalled. The tradeoffs are as follows.

- Benefits — installation allows you to bring up the system by pressing ⤶ at the *SYSTEM PATHNAME?* query, instead of typing a pathname like SYSGEN:NEWSYS.SY⤶. Because the system is invisible, it cannot be deleted.
- Disadvantages — You can't tell which system is running, or which is installed, if you have multiple systems in :SYSGEN.

Only one AOS system can be installed on an LDU, although there can be many system.SY files on it. So if you install a system, it will overwrite the installed system (if any) on the LDU. The original system file will remain as a disk file in directory :SYSGEN (or wherever it was generated).

Whether or not your primary AOS system is installed, its name should be posted on the system console (perhaps on a paper tape label). If you install a system, you should write "Installed" and the date on the label.

About the Installer

The Installer can install a disk bootstrap, SYSBOOT, and/or AOS system on any LDU formatted as a system disk.

There are two versions of the Installer: a stand-alone version that runs only when AOS is not running; and a stand-among version that runs under AOS.

The stand-alone Installer requires an AOS system tape or diskette and works with any system LDU.

The stand-among Installer accepts disk file pathnames and doesn't use a system tape; it works only with a *nonmaster* LDU that has not been initialized (CLI command INITIALIZE).

If You Make a Mistake

If you type a response to the Installer and want to change it before pressing ⤶, press the DEL key as needed or press CTRL-U to erase the line.

If you are beyond the line containing the mistake, you should proceed to the end, then rerun the Installer. It asks only a few questions and runs fast.

For any Installer error message, see Table 12-1, at the end of the chapter.

The Stand-Alone Installer

The stand-alone Installer is in file INSTL in the master LDU's root directory. You can boot it directly from this LDU.

To use the Installer, perform the following steps:

1. Make sure all disks in the LDU are mounted, and that their units are ready and write-enabled.

2. Mount an AOS system tape or diskette on unit 0, on the first controller, or insert a system diskette in any unit. The tape/diskette can be either your own tailored system or the DG-supplied AOS starter system tape or diskette. (Making a tailored AOS system tape or diskette is described near the end of Chapter 4.)

3. If AOS is not shut down, shut it down. On an MV/8000, make sure that the SCP-CLI is active on the system console. (You can run the stand-alone Installer only from the system console.)

4. Now you are ready to program load. How you do this depends on what type of computer you have:

4a. If you have an S/20, S/120, S/280, or a DESKTOP GENERATION system, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type nnH, where nn is the device code of your master LDU. For example

    ! 26H   (S/20, S/120, S/280, or a DESKTOP GENERATION SYSTEM)

    Go to step 5.

4b. If you have an S/140, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type 11A. The system will display a 6-digit number, after which you type 1000nn ↵, where nn is the device code of the master LDU. Then, after the next *!* prompt, type 1000nnL. For example

    ! 11A   *xxxxxx*   100033 ↵   (S/140)
    ! 100033L

    Go to step 5.

4c. If you have an MV/8000, the system console displays an *SCP-CLI>* prompt. Next to the prompt, type RESET↵. Then, next to the second prompt, type BOOT 27↵. For example

    SCP-CLI> RESET ↵   (MV/8000)
    SCP-CLI> BOOT 27 ↵

    Go to step 5.

4d. If your computer has hardware data switches (numbered 0 or X4/0 through 15), make sure they are set to 1000nn, where nn is the device code. Lift and release the RESET switch; lift and release the PR LOAD (PROG LOAD) switch. Go to step 5.

5. Step 4 loads the disk bootstrap, which loads the system bootstrap, SYSBOOT. Then SYSBOOT says

    SPECIFY EACH DISK IN THE LDU:
    DISK UNIT NAME?

    Type the name of your master LDU. For example

    DPI0 ↵
    DEVICE CODE?

6. Unless you know the disk unit is on a nonstandard code, press ↵ to select the default code.

   ↵

   SYSBOOT now asks

   SYSTEM PATHNAME?

7. Type the filename of the Installer program (INSTL):

   INSTL ↵

   SYSBOOT now loads the Installer into memory.

8. The Installer says

   AOS INSTALLER REV n
   ENTER ALL UNITS IN LDU:
   DISK UNIT NAME?

9. Type the unit name that holds the first physical disk in the LDU; e.g., DPF10 ↵ for the first disk unit on the second controller. The unit names and device codes of all disks supported by AOS in Table 11-1.

   The Installer now asks for this unit's device code:

   DEVICE CODE?

10. If this unit's controller is on the default device code, you can press ↵ to choose the default answer.

    The default device code is 27 for the first DPF-type controller, and 67 for the second DPF-type controller; for the first DPJ-type controller, it is 24. For other disks, see Table 11-1.

    Next, if the LDU was formatted to contain more than one physical disk, the Installer asks

    DISK UNIT NAME?

11. The Installer repeats the *DISK UNIT NAME?/DEVICE CODE?* questions until you have identified the units that hold all disks in the LDU, and their device codes.

    When you have identified them all, the Installer writes a disk bootstrap to the LDU and says

    —DISK BOOTSTRAP INSTALLED
    INSTALL A SYSTEM BOOTSTRAP?

12. Type Y↵ if you want a copy of the system bootstrap (SYSBOOT) written to the LDU. Or type N↵ if you do not. Installing the system bootstrap overwrites the existing bootstrap (if any) but does not affect the installed AOS system (if any).

    From tape, you should generally type Y↵ — especially if there is a new revision of AOS on this tape. For diskette, generally type N↵. If you don't want a new SYSBOOT installed, answer N↵ and go to step 13.

12a. If you type Y↵ the Installer asks

FROM MAG TAPE (M) OR DISKETTE (D)?

If you answered M↵ above, the Installer asks

FROM MT-0, FILE #:

If you answered D↵ above, the Installer asks

DISKETTE UNIT NAME ?

12b. To install from mag tape, type 4↵ and skip to the next paragraph. To install from diskette, make sure that the diskette is mounted in a unit (replace the system diskette, if needed). Then type the name of the unit; for example, DPI1↵.

The Installer now reads the tape or diskette and tries to copy SYSBOOT to the LDU. If successful, it displays

—SYSTEM BOOTSTRAP INSTALLED
INSTALL A SYSTEM?

If you choose to install an AOS system, it will overwrite the system (if any) that is installed on the LDU. This all depends on the system tape or diskette you're using. If it's your own tape or diskette, then it has a tailored AOS system on it. If it's a DG-supplied AOS tape or diskette, then it has *the starter system* on it.

13. Type Y↵ or N↵, depending on whether or not you want the system that is on the tape or diskette installed on the LDU. If you type N↵, you're done; the Installer says *DONE!,* and terminates.

If you type Y↵, the Installer asks

FROM MAG TAPE (M) OR DISKETTE (D) ?

And you should answer as you did in step 12. Then, as before, the Installer asks either

FROM MT-0, FILE #:

or

DISKETTE UNIT NAME ?

13b. To install from tape (either a DG-supplied system tape or an AOS tape you created using the SYSTAPE macro), type 5↵. To install from diskette, make sure that a diskette with a system is inserted in a diskette unit. Then type the unit name; e.g., DPI1↵.

The Installer now reads the tape file or diskette and tries to install the contents on the LDU, placing a pointer in invisible space as described earlier. Then it displays the following messages and terminates:

—SYSTEM INSTALLED

DONE!

To run the Installer again, type CONTINUE↵ (MV/8000), P↵ (other programmed consoles) or press the CONTINUE switch and retrace the steps above. Otherwise, you might want to bootstrap the AOS system on this LDU to see how it does. After you specify each disk and device code in the LDU, press ↵ to bring up the installed system.

Figure 12-1 shows a sample stand-alone Installer dialog.

```
Have all disks ready and write-enabled.

SCP-CLI>   RESET ↵
SCP-CLI>   BOOT   22 ↵

FROM MT-0, FILE #:     3 ↵

AOS INSTALLER REV n
ENTER ALL UNITS IN LDU:
DISK UNIT NAME?     DPF10 ↵
DEVICE CODE?     ↵

DISK UNIT NAME?     DPF11 ↵
DEVICE CODE?     ↵

--DISK BOOTSTRAP INSTALLED
INSTALL A SYSTEM BOOTSTRAP?     Y ↵

FROM MAG TAPE (M) OR DISKETTE (D)?     M ↵
FROM MT-0, FILE #?     4 ↵

--SYSTEM BOOTSTRAP INSTALLED
INSTALL A SYSTEM?     Y ↵

FROM MAG TAPE (M) OR DISKETTE (D)?     M ↵

FROM MT-0, FILE #?     5 ↵

--SYSTEM INSTALLED
DONE!
```

DG-26908T

Figure 12-1. Installing Bootstraps and an AOS System on an LDU, via the Stand-Alone Installer (MV/8000)

The Stand-Among Installer

The stand-among Installer is in directory :UTIL, filename INSTL.PR. It works only with nonmaster LDUs that were formatted to include a system disk. It won't work on the master LDU or any LDU that has been initialized from the master.

Before running it, you should know the full pathname of the AOS system you want to install. To run it, follow these steps.

Make sure all disks in the LDU are mounted, and that their units are ready and write-enabled.

Beside the AOS CLI prompt, type

```
) DIR    :UTIL ↵
) SEARCH   [!SEARCH],   :PER ↵        (PER needs to be in the searchlist.)
) XEQ   INSTL.PR ↵

AOS INSTALLER REV n
ENTER ALL UNITS IN LDU:
DISK UNIT NAME?
```

Type the unit name that holds the physical disk in the LDU; for example, DPF10↵. The unit names of all disks supported by AOS appear in Table 11-1, in the previous chapter.

If the LDU was formatted to contain more than one physical disk, the Installer asks

DISK UNIT NAME?

The Installer repeats the *DISK UNIT NAME?* questions until you have identified the units that hold all disks in the LDU.

When you have identified them all, the Installer writes a disk bootstrap to the LDU and says

–DISK BOOTSTRAP INSTALLED

INSTALL A SYSTEM BOOTSTRAP?

Type Y↵ if you want the system bootstrap (SYSBOOT) written to the LDU. Type N↵ if you do not. Installing a system bootstrap overwrites the existing bootstrap (if any) but does not affect the installed AOS system (if any).

Generally, it can't hurt to answer Y↵. But if you don't want a new SYSBOOT written to the LDU, answer N↵ and go to the *INSTALL A SYSTEM?* question.

If you say Y↵, the Installer asks for the bootstrap's pathname:

PATHNAME?

The SYSBOOT pathname, in root directory, is :SYSBOOT, so type :SYSBOOT↵. The Installer now copies SYSBOOT to the LDU and says

-- SYSTEM BOOTSTRAP INSTALLED

INSTALL A SYSTEM?

If you choose to install an AOS system, it will overwrite the system (if any) that is installed on the LDU. You can install any existing AOS system that you want.

Type Y↵ if you want to have a system installed; type N↵ if you do not. If you answer N↵, you're done; the Installer says *DONE!* and terminates.

If you answer Y↵, the Installer wants to know the

PATHNAME?

Type the full pathname, with .SY suffix, of the AOS system you want to install. For example, :SYSGEN:NEWSYS.SY ↵.

The Installer then copies the system to the LDU, placing a pointer in invisible space as described earlier. Then it terminates and control returns to the CLI.

-- SYSTEM INSTALLED
DONE!
)

To run the Installer again, type XEQ INSTL↵ and retrace the steps above. Otherwise, you might want to shut down the current AOS system and bootstrap the installed AOS system on this LDU to see how it does. After you specify each disk and device code in the LDU, press ↵ to bring up the installed system.

Figure 12-2 shows a sample stand-among Installer dialog.

```
Ready all disks in the LDU.

) DIR   :UTIL ⤸
) SEARCH   [!SEARCH],:PER ⤸
) XEQ INSTL ⤸

AOS INSTALLER REV n
ENTER ALL UNITS IN LDU:

DISK UNIT NAME?      DPF10 ⤸

DISK UNIT NAME?      DPF11 ⤸

--DISK BOOTSTRAP INSTALLED
INSTALL A SYSTEM BOOTSTRAP?      Y ⤸

PATHNAME?      :SYSBOOT ⤸

--SYSTEM BOOTSTRAP INSTALLED
INSTALL A SYSTEM?      Y ⤸

PATHNAME?      :SYSGEN:NEWSYS.SY ⤸

--SYSTEM INSTALLED

DONE!

)
```

DG-26909T

Figure 12-2. Installing Bootstraps and an AOS System on an LDU, via the Stand-Among Installer

Installer Error Messages

While you are running the Installer, you may receive one of the error messages shown alphabetically in Table 12-1.

Table 12-1. Installer Error Messages

Message	Meaning
ABORT message	The Installer hit a fatal error and aborted. If the message allows you to correct the problem, do so and retry. Otherwise, try to find the message text in this table.
BAD SYSTEM FILE	The tape file whose number you specified is too big for its reserved area on the LDU. This probably means that you specified the wrong file. If you get a *MUST RUN FIXUP* message, run FIXUP on the LDU. Rerurn the Installer and type the correct file number for the system tape. The number is 4 for a system bootstrap, 5 for an AOS system. Make sure the tape is a system tape, created by the SYSTAPE.CLI macro.
DEVICE ALREADY IN USE	The Installer can't open the disk(s) involved. This occurs if you try to run the stand-among Installer on an LDU that a program has already opened (for example, the master LDU). Try the stand-alone Installer.
DISK ERROR, DEVICE d u STATUS= s	The Installer has encountered a disk block from which it cannot read or write; it aborts. The *d* is the device code, *u* is the unit number, and *s* is the hardware status code, described in the Peripherals manual (see Preface). Make sure the unit(s) are write-enabled and retry. If problem persists, try a Disk Formatter Partial format (Chapter 11). The disk may need hardware (DTOS) diagnostics; consult your DG support organization.
DISK FORMAT REV # MISMATCH	The disk has an early revision number; or perhaps the Disk Information Block (DIB) has been damaged. Try a (current revision) Disk Formatter Partial format on the LDU, changing nothing. If this runs, bring up AOS, try to dump all files from the LDU, run a Disk Formatter Full on the LDU, and run the Installer again.
DISK SPACE EXHAUSTED	The Installer cannot find the disk space it needs to install a disk or system bootstrap. The easiest way to handle this is to dump the LDU, run a Full format on it, try the Installer again, and reload the files.
FILE DOES NOT EXIST	The entry you typed does not identify an existing tape or disk file. Try retyping the devicename, or respecify the tape file number, or type an existing file pathname.
HARD TAPE ERROR	The Installer has hit a hard error on the tape unit. Retry. If the same message recurs at the same point, the tape is probably bad; try another system tape. If it recurs, you may need to run hardware diagnostics on the tape unit or contact your DG support organization.
INCONSISTENT LDU	One or more of the disks involved do not belong to the LDU. Mount the correct disk(s) and retry; or, specify the disks in a different order.

(continues)

Table 12-1. Installer Error Messages

Message	Meaning
INVALID LOGICAL DISK ADDRESS	The installer could not convert the logical address to a valid physical disk address. If this error occurs while installing a file onto a diskette, it usually means that the file is too large to fit on the diskette.
INVALID DISK FORMAT	The disk has not been formatted via a Disk Formatter Full format. Run a Formatter Full format on the disk(s) involved.
INVALID DISK SEQUENCE NUMBER	The disk sequence number is not between 1 and 8. For action, see *DISK FORMAT REV* message, above.
MUST RUN FIXUP ON THIS LDU	The LDU is unstable because a utility or AOS has aborted. Run FIXUP on the LDU.
SOFT TAPE ERROR	A soft (correctable) tape error occurred; proceed as usual.
SYSTEM BOOTSTRAP AREA NOT ALLOCATED	The LDU was not formatted as a system disk. Run a Disk Formatter Partial format, say Y to *ALLOCATE A SYSTEM BOOTSTRAP AREA*, and change nothing else (described in Chapter 11).
SYSTEM BOOTSTRAP TOO LARGE	The file you specified is too large for the SYSBOOT area reserved by the Disk Formatter. See message *BAD SYSTEM FILE* in this table for recovery action.

(concluded)

What Next?

This chapter told you how AOS bootstrapping software works, and how to use the Installer utility to place this software on an LDU.

Next, you might want to read about some cautions and hints to the operator (in the next chapter), or system management issues (in Chapter 14).

End of Chapter

Chapter 13
Cautions and Hints to the System Operator

Read this chapter

- when you want to learn some simple things to avoid;
- when you have a problem with the system and don't know what to do.

This chapter attempts to tell the operator, as quickly and tersely as possible, what not to do, and what to do in confusing or serious error situations. Each section ends with a series of blank lines — designed for the operator, system manager, or anyone in authority to add his or her own cautions and hints. Neither section is exhaustive; it simply attempts to cite common situations.

The major sections are

- Cautions
- Hints
- What Next?

Cautions

By observing the following list of cautions you will help preserve the integrity of your multiuser system.

- Don't cut power to CPU or disk units without shutting the AOS system down; if the CPU has a LOCK switch, keep it in the LOCK position.
- On an MV/8000, don't type main processor control commands like HALT or RESET to the SCP-CLI when AOS is running.
- Always shut down the system normally unless an emergency forces an abnormal shutdown.
- On an MV/8000, S/280, S/120, S/140, and all DESKTOP GENERATION systems, don't panic if you accidentally hit the break sequence and see the SCP-CLI running on the system console. Type TTY↲ on an MV/8000, and P ↲ on the others, to return control to AOS.
- Don't allow hardware diagnostics (ADES or DTOS) to run on a disk that has valuable data, unless you know the program will use only a reserved diagnostic track on the disk. Hardware diagnostics destroy the file structure and status information on some disk models. Generally, consult your DG support organization before running hardware disk diagnostics. When running tape diagnostics, always use a scratch tape.
- Don't leave the CPU, disk units, or system console unattended for long periods, unless you know that all people in the area are reliable and experienced users. Cutting power to the CPU or disks, or resetting the main processor, will usually bring AOS down abnormally.
- Don't leave a privileged CLI running on the system console. The master CLI and its son CLIs (except for LOCK_CLI) have all process control and super privileges. Do not endanger system integrity by allowing users access to these privileges.
- If your system is connected to another site — via modem or network — consider remote user privileges carefully. A privileged profile, with network or modem access, gives a network or modem user access to all your files from outside your installation.

Use the following blank lines for your own notes or comments.

Hints

This section describes topics alphabetically by subject.

batch streams — EXEC must be running, streams continued, printer started, and XLPT process running. If streams aren't running, check with CX STATUS↵ and ?↵ (for the XLPT process).

backup — Follow routine, using labeled tapes, diskettes, or PCOPY disk-to-disk. Macros that ease this are shown in Chapter 9. All databases (CEO, INFOS II, etc.) and user files must be closed to be effectively backed up.

bootstrapping — Type RESET↵, BOOT 27↵ or BOOT 33↵, and disk unit name (DPF0↵, DPI0↵), pathname, date and time, and default everything else; UP↵.

break sequence — The BRK, CMD and BREAK, or BREAK keys give the SCP- CLI control of the system console (MV/8000), or display the *!* prompt (computers with programmed consoles), suspending any program that is running (such as AOS). To restore it to AOS, type TTY ↵ on an MV/8000, and P ↵ on the others.

CLI (AOS) — A key to system control; use Help as needed. Read the CLI manual.

clock, system — This must be correct; SYSLOG and all date/time operations depend on it. Be sure to enter the correct date and time when you bring up the system. If the date or time is wrong, correct it with the CLI command DATE or TIME — when EXEC is *not* running.

console, system — For security, on an "open" system, run LOCK_CLI (Chapter 9). When the SCP CLI or master AOS CLI (PID 2) is running, anyone at the system console has access to the entire AOS system. If the system console seems dead, type CTRL-Q to clear a possible CTRL-S; make sure it is on and on line; check fault lights (if any is on, turn console off and on again). If there is no response, see Chapter 6, "Deadlocks" or Appendix A, message "(nothing)."

CPU — Maintain regular preventative maintenance (vacuuming, filters, and so on). If there are problems, like recurrent fatal errors or failure to bootstrap, check Appendix A. When AOS is up, avoid the front panel switches, unless you must unlock to break an AOS deadlock (break, then run ESD), or unless you want to cut CPU power. If you can, lock the computer before turning power on; this will simplify bootstrapping.

definitions — See the Glossary.

diagnostics — Diagnostics are usually run by a DG field engineer; you can consult your support organization for advice. When running any kind of diagnostic, keep all the devices except those being tested off line or write-disabled (to prevent inadvertant erasure of data).

disk packs — Keep each in its unit or protective cover, in a dust-free place. Keep a paper tape label on each disk cover or unit, with unit name, device code, LDU name and ID, and name of other disks in the LDU (if this applies). Do not leave a disk pack in a precarious position, where vibration or accident could cause it to fall.

disk units — Never turn off unless AOS has been shut down. Make sure they are write-enabled before bootstrapping AOS. If you are running a non-AOS program and want to make sure a disk won't be written to, then press the write-enable switch to disable or off. In an "open" shop, you may want to keep users away from the units. Maintain regular preventative maintenance (vacuuming, filters, and so on). After a FATAL ERROR, or HARD ERROR, if you suspect surface damage on a disk unit that uses removable packs, don't insert another pack in the unit to check. Also, don't put the suspect pack in another unit. (A pack with surface damage may damage the read/write heads of any unit that runs it.)

diskette — Do not open the diskette door or press the CPU RESET switch while the diskette light is on. Wait a few seconds for the light to go off; then open the door. Handle diskettes carefully; do not write on a diskette label with a ball-point pen; and store diskettes vertically in a temperate, dust-free place, away from strong magnetic fields.

errors, fatal, hard, or soft — Messages are in Appendix A. On a fatal AOS error, note the panic values in the logbook, do a memory dump as a matter of course, then run ESD with open file report. If ESD runs, reboot AOS. If ESD fails, rerun it. If ESD fails again, get an open file report. Run FIXUP on all LDUs with open files; then reboot AOS. On hard disk errors, make sure the unit is on-line and write-enabled; if so, and the status code indicates a bad sector (bad block), run a Disk Formatter Partial with read-only analysis (Chapter 11) and tell the Formatter to update the bad block table; then run FIXUP and try again. If hard errors persist, consult your DG support organization and run without the disk until it is repaired. Soft errors on tape are a potential danger sign only; but if one occurs on a disk, shut it down and run a Formatter Partial as described earlier in this paragraph.

EXEC — Use CLI QDISPLAY, XHELP, CX STATUS and CX MOUNTSTATUS commands as needed; CX PAUSE to pause streams and printers; CX HOLD on a dubious request, CX FLUSH to kill an active request. For START syntax, read :UP.CLI via the CLI TYPE command. Use XEQ PED and ? macro as needed.

fault light — On the CPU front panel, this usually means a problem with power supply; see appropriate part of Appendix A. If a fault light glows on a disk unit at startup, turn disk power off and on again. If a disk shows a fault while on line, release it from AOS (if possible) and cut power. On a line printer, the TAPE FAULT light may mean EXEC's XLPT process terminated; check with ?↵; if there is no XLPT process, start it with EXEC's START and CONTINUE commands (syntax is in UP.CLI macro). If a printer FORMAT light won't go off, turn the printer off and on, then type CX ALIGN @devicename ↵ to it, align the paper, and type CX ALIGN/CONTINUE @devicename ↵. If the FORMAT light remains lit, turn printer power off, terminate the XLPT process with CX TERM @devicename ↵, turn power on, bring XLPT up again (as above), then type ALIGN and ALIGN/CONTINUE commands as above. On a hardcopy console, a fault light may mean that there is a jammed or damaged ribbon; check. If the ribbon is okay, turn console power off and on again to clear the fault.

help — Use help or XHELP; or ? or CX STATUS or CX MOUNTSTATUS. Call on a resource person: system expert or DG support organization.

labeling diskettes — Use the CLI's labeling mechanism.

labeling tape — Use XEQ LABEL @unitname labelname ↵, up to six characters; or, with PCOPY, use PCOPY.

line printer — If it won't stay on line, EXEC's XLPT co-operative process may have terminated; check with ? macro; restart if needed with EXEC START and CONTINUE commands contained in :UP.CLI. (Also see the subject fault light.)

macros — CLI macros are a key to easy operation. Use the ones available (some are shown in Chapter 5); and write your own as needed.

magnetic tapes — These are often the mainstay of data backup. You can label them with the LABEL utility at any time, even when EXEC is prompting for a UNIT MOUNT. Archive tapes should have paper labels on them and be stored routinely in a library.

magnetic tape units — Keep unloaded and off line unless they are in use. Soft errors can sometimes be reduced if you clean the read/write heads with an alcohol-soaked Q-tip.

panics — See errors.

preventative maintenance — Make sure it's done routinely; it can save a lot of time and money.

processes — Use ? or XEQ PED as needed.

shutdown, normal — Pause streams and printer(s) with CX PAUSE commands; get all users out of text editors and other interactive programs (SEND or BROADCAST); get back to master CLI (BYE or UNLOCK, password, BYE for LOCK_CLI); then BYE and YES.

slow, sluggish system — A high-priority process (like a dump) may be consuming most of the system's resources. Or, many batch streams may be running. Generally, run dumps and multiple batch streams after normal working hours. Possibly, if your system console is a hard-copy console, it may be having hardware faults (like a worn-out ribbon); a system console fault that slows console printing can slow the whole system.

super- privileges — Don't leave them on; a super- prompt showing on an unattended system console means that the system is wide open. Try to set up access control to minimize your need to turn them on. Use EXEC commands where possible.

terms — See the Glossary.

text editing — You can do this with either the SED or SPEED text editor; SED is sketched in Chapter 5.

users —To check, use ? or XEQ PED. For a new user, run PREDITOR. Within reason, do what you can to boost productivity; use LOGON.MESSAGE for news; encourage use of help messages; create your own help messages (Chapter 5). For a user's view of AOS, read *Learning to Use Your AOS System.*

Use the following blank lines for your own notes or comments.

What Next?

This chapter suggested some things to avoid, and some things to consider, for the person who is acting as the system operator.

You may want to look at the next chapter or an appendix of interest, or review earlier material, or read another pertinent book, or maybe simply run the system.

End of Chapter

Chapter 14
System Management Considerations

Read this chapter

- when you want to understand process and disk space concepts to make your system more efficient;
- when you want to make your system as secure as possible;
- for a summary of the things that should be done regularly (logs, dumps, preventative maintenance, etc.)
- when you receive an AOS update or new revision;
- when you want to know how to get help from DG;
- for an example of a real UP.CLI macro.

Thus far, this book has simply given information, without caring about the role of the reader. This chapter touches on some management issues — not *how* to manage, but information and suggestions that can help the person who manages the system make some system-oriented decisions.

In some organizations, the person who physically operates the system has limited powers. LOCK_CLI runs on the system console, only a few people (who may not be present) know the password. The person who operates the system must use EXEC and other CONTROL commands to control the system. If this applies to your organization, you can use this chapter to learn how to make *this book* suitable for the people who operate the system. The pertinent section is "System Security".

The major sections within this chapter are

- AOS Process Types and How to Use Them
- Disk Space and Performance
- System Security
- Routine Procedures (things to do regularly)
- How to Handle Updates and New Revisions from DG
- Getting Help from DG
- A Real System's UP Macro
- What Next?

AOS Process Types and How to Use Them

This section explains how AOS manages processes, then offers some suggestions for your own system.

In the main, AOS manages its resources quite efficiently, apportioning memory and CPU time to interactive and batch processes according to their types and priorities. But there are some steps you can take to optimize this for your own needs.

The details on AOS memory management appear in the *Advanced Operating System (AOS) Programmer's Manual*, Chapter 2.

Processes in Memory

AOS runs each program as a *process*, with its own process ID (PID). There can be many processes — up to 64 — all running simultaneously.

Processes compete for main memory according to *type*; priority is secondary. The process types are

- resident
- pre-emptible
- swappable

The AOS system scheduler allots each type main memory as follows.

Process Type	How It Gets Main Memory
Resident	Gets main memory on demand and keeps it.
Pre-emptible	Gets main memory if the memory is not needed by a resident process. The scheduler swaps a pre-emptible process if: A resident or higher-priority pre-emptible process requires memory; or The process becomes blocked and *any* other process requires memory.
Swappable	Gets main memory if the memory is not required by a resident or active pre-emptible process. The scheduler can swap the process when a resident or pre-emptible process requires main memory.

By default, all user processes, including batch processes created for users, are swappable.

Processes and CPU Time

When a process gets control of the CPU, it retains control for a period called a *timeslice*. AOS determines the timeslice from the behavior and priority of the process. After the timeslice period expires, AOS gives control to another process. AOS can change the timeslice, according to the behavior of the process.

But before a process gets its timeslice, it must compete with other processes and get control of the CPU. Processes compete for CPU time on the basis of priority, as follows.

Process Types and Priorities	How They Get CPU Control
Resident and Pre-emptible 1 = highest 255 = lowest	After either type wins memory, the scheduler treats the two types the same way. It gives CPU control based on priority. For example, if a resident process with priority 2 and a pre-emptible with priority 1 simultaneously demand the CPU, the pre-emptible gets it first.
Swappable 1 = highest 3 = lowest	If no resident or pre-emptible process is ready and due to run, the scheduler gives the process CPU control based on the process' behavior and (less significantly) its priority.

The magnitude of difference between priority numbers has no effect — only the relative difference. This means that running four processes at priorities 4, 20, 200, and 201 is exactly the same as running them at priorities 1, 2, 3, and 4, if they are the only processes on the system.

CPU Competition among Swappable Processes

Swappable processes compete for CPU time only when no resident or pre-emptible process wishes to run. The scheduler chooses a swappable process for CPU control based on its prior behavior and priority (1 to 3 for swappable processes). The scheduler favors interactive processes (processes that request CPU time often but use small amounts at a time).

The priority of a swappable process has *some* impact here — less, however, than its prior behavior.

Choosing Process Types and Priorities

If you take no action to control process types or priorities, nearly all processes will be swappable, priority 2. This is the PREDITOR default type and priority.

The master CLI and its sons, including EXEC and all its sons, are all created as swappable processes. The qualities and tradeoffs on the types are as follows:

- Resident Processes. A resident process cannot swap; the process always remains in main memory. Its pages remain in main memory until the process either issues a ?MEMI system call or terminates.

  AOS always runs its peripheral manager as a resident process. Some other DG products are designed to run resident, as recommended in the product documentation; generally you will want to follow the recommendations. But otherwise, avoid creating a resident process unless it must be very responsive. You might consider residency for something like a small alarm process in a real-time environment.

- Pre-emptible Processes. A pre-emptible process commands main memory as long as it remains unblocked, unless a resident or higher priority pre-emptible process needs the memory. This is a good general-purpose type for a process you want to favor over swappable processes.

  Processes that you *might* consider making pre-emptible are EXEC's XLPT co-operative (if you want to do a lot of printing), or batch jobs (the backup macros shown in Chapter 9 run dumps as pre-emptible processes). Other candidates include important short-term processes. For example, assume a short-term process needs to run uninterrupted every 10 minutes, and will then issue a delay call. You might make it pre-emptible, with the highest priority (1). Then, the scheduler will not permit any lower-priority process to displace it while it runs. When it issues the delay call, it becomes blocked, and AOS swaps it out. When the delay period has expired and it's ready to run again, it can displace any other lower-priority nonresident process in main memory.

- Swappable Processes. Swappable, priority 2, is the default for all processes. All swappable processes of the same priority compete for system resources on an equal footing. Unless you want to favor a process, let it be created swappable, with priority 2.

  You might consider a priority of 1 if a process needs an edge over other swappables. Or you might consider a priority of 3 if you want a process to have the least possible effect on other processes. For example, in a class, the instructor's process can run at priority 1 while the students' processes run at the default, priority 2.

Generally, you can't really decide on the type and priority issue until you've had some experience with your applications software. So we suggest that you run with the defaults for awhile before setting up your own process parameters.

Creating Different Kinds of Processes

The primary process-creating command is PROCESS. It creates a son process with the type, priority, and other parameters specified with switches.

A process can change another process' (or its own) type as long as that process is in the caller's process hierarchy, with the command PRTYPE. Similarly, it can change priority with the command PRIORITY.

But no process can create a son with, or give itself, privileges that are not granted in its user's user profile. The PREDITOR defaults don't allow a process to change priority or type.

The master CLI and any of its CLI sons (except LOCK_CLI) can change its type or priority; and it can create sons of different types and priorities. So the system operator, via a privileged CLI, can always start any kind of process; for example:

```
) PROCESS/RESIDENT/PRIORITY=1/SUPERU/DEFAULT   MYPROG ↲
```

But probably, you will not want to do this directly. Instead, after deciding on your process parameters, you will want to put them in the UP.CLI macro.

If you will want *users* to be able to create processes of different types and priorities, you must edit their profiles to allow this. Do it with caution, because misuse of these privileges can allow undesirable processes to dominate the system.

Table 14-1 names the PREDITOR profile privileges in the order asked, and describes the effect on the user's processes. Each PREDITOR question gets more detail in Chapter 7.

Table 14-1. Profile Privileges That Relate to Process Control

Privilege	What It Allows
CREATE WITHOUT BLOCK	The user process can create sons (that in turn can create sons) without blocking. This allows the user process to proliferate. All its sons (and grandsons) can remain unblocked, increasing system overhead. This privilege relates to the next two. A CEO user needs this privilege.
UNLIMITED SONS SONS	These two parameters control the number of sons the user process can create. Each son can have all the privileges of its father. Generally, a user should not have the unlimited sons privilege. Each son requires some overhead, especially if its father isn't blocked (first question).
CHANGE PRIORITY	The user process (and sons) can change its own priority if it has this privilege.
CHANGE TYPE	The user process (and sons) can change its own type if it has this privilege.
CHANGE USERNAME	The user process (and sons) can change its username, perhaps to a privileged name like OP, and acquire access to EXEC.
SUPERUSER	Of itself, this privilege doesn't affect the multiprocess environment. But superusers can run PREDITOR, and give themselves any privilege.
SUPERPROCESS	The user process (and sons) can change its own type and priority; and it can block or terminate any other process.
PRIORITY	The user process (and sons) can change its priority if it has this privilege.

EXEC Process Control Options

EXEC has a number of commands you can use to tailor the multiprocess environment. They are

LIMIT — can limit CPU time for batch streams or page usage for line printers.

PRIORITY — can change the process type and priority of batch and spooler (printer) processes (combines CLI commands PRTYPE and PRIORITY for these).

QPRIORITY — can direct batch streams or devices to accept only requests that fall within a given queue priority range.

XBIAS — can instruct batch streams or printer queues to favor small requests or large requests.

Each of these EXEC commands is described in Chapter 8.

Disk Space and Performance

AOS must make at least one disk access whenever any process wants to read, write, create, or delete a file. In an active system, it may make hundreds of accesses each minute.

When your system is new and its LDU(s) hold relatively little material, file access time is very short. (Creating LDUs is described in Chapter 11, "The Disk Formatter.") But as files accumulate, AOS and the disk hardware may require more time to access files.

There are several things you can do to streamline disk access and keep performance near its optimum. There are four factors involved.

- Overall free space
- File fragmentation
- Bitmap and overlay area addresses
- Directory hash frame size

Overall Free Space

The amount of free disk space usually affects performance more than any other factor. When more than 70% of an LDU's blocks are occupied, the read/write heads must spend significantly more time moving over the disk to find free space whenever a process creates a file. Above 70% of capacity, the access time needed rises steeply. Much above 70%, everyone on the system may note increased response time. If more than 95% of an LDU is used, you should take immediate action to free some space.

The most convenient way to handle the space factor is by limiting each user's space in his or her PREDITOR profile. Chapter 7, the PREDITOR chapter, explains several ways to approach this. If disk space is tight, and you don't want to acquire more disk units just now, you can ask users to delete all their old .ST, .PR, and .OB files. Temporary files and break files (suffixes .TM, .TMP, .ED, and .BRK) can also be deleted; so can backup files (.BU suffix) after the file system is archived (backed up). Having users delete files they don't need can open up a considerable amount of space.

There may also be obsolete files in :UTIL, and obsolete user directories in :UDD. You can simply delete the former via the CLI. You should dump the latter for the record, then use the PREDITOR D command to delete both the obsolete profile and user directory.

Use the CLI command SPACE : (or other control-point directory name) to check number of disk blocks used and remaining. For example, on a 190-megabyte (370889-block) LDU:

) SPACE :↵
MAX 370889, CURR 259662, REM 111277 (About 70% of the blocks are used.)

.

.

.

(Time passes.)

) SPACE :↵
MAX 370889, CURR 315260, REM 55629 (About 85% of the blocks are used.)

Get users to clean up.

) SPACE :↵
MAX 370889, CURR 240430, REM 130459 (Better; under 70% of the blocks are used.)

File Fragmentation

Each AOS file has one or more index blocks and multiple data elements. (An element is [by default] one disk block, but anyone can specify many more blocks — creating a file with many contiguous blocks — with the CLI CREATE command and /ELEMENTSIZE switch.)

After an LDU begins to fill up, files become fragmented: AOS must search farther on disk for space for new or modified file elements. For example, one element could be near an outer cylinder, and the next available space near an inner cylinder. To write the next element, the system must move the read/write head all the way across the disk.

File fragmentation often occurs when an LDU is nearly full. Simply deleting files (as above) may or may not eliminate it. If cleaning up the LDU doesn't help, and the LDU's unit(s) appear to be doing a lot of seeks, you can suspect fragmentation.

To minimize or eliminate fragmentation, have the LDU cleaned up as above. Then dump all files from the LDU and run a Disk Formatter Full format on it, as described in Chapter 11. Make sure the bitmap and overlay area are near the "middle" of the first disk. Then reload all the files and see if performance picks up. It usually will.

You can always eliminate fragmentation of a specific file by creating it with a large element size, but this is impractical and inefficient for most files. It is useful — and recommended — for database files that will be used by data management systems like INFOS II.

Bitmap and Overlay Areas

AOS must write to the bitmap every time it allots a new disk block to a file. It must access the overlay area every time it needs a system page that isn't already in memory. If an LDU is almost full, these areas should be near the "middle" of the disk to minimize disk seek time. They may already be near the middle; you can check with a Disk Formatter Partial format run.

Directory Hash Frame Size

The hash frame size is part of the algorithm that AOS uses to decide where to record any filename that it creates in a directory.

You don't need to understand how this algorithm works, but you may want to understand how to use hash frame sizes to speed up file access.

Each directory has a hash frame size, assigned when the directory was created. The default hash frame — used when the creating process omits a size — is 7. This is suitable for directories that contain up to about 140 files. But there are a few directories that contain many more than 140 files. UTIL, for example, may be home to hundreds of files. And UTIL is an often-used directory. Similarly, for a small directory with about 10 files, a good hash frame size would be 1.

For the optimum hash frame size, if a directory will contain many more than 140 files, divide the number of files by 20 and take the nearest prime number. Then, create the directory, specifying the optimum size with CREATE's /HASHFRAMESIZE= switch. The hash frame size is transparent to users; only AOS sees it, so people can access the directory just like any other directory. To check a directory's hash frame size, use FILESTATUS' /HASHFRAMESIZE switch.

If a directory already exists (as UTIL does), and you want to change its hash frame size, you must dump all its files, delete it, create it with the desired hash frame size, and reload the dumped files.

For example, let's assume you've been using your system for a while. UTIL now contains some 300 files, and you want to change its hash frame size. Dividing 300 by 20, and allowing for growth, you decide on a new hash frame size of 17. To implement it, you'd follow these steps:

- Dump UTIL to magnetic tape (DIR : ↵, DUMP/V tape UTIL:# ↵).
- Check UTIL's hash frame size, then delete and recreate it with the following CLI commands.

```
*) FILES/HASH   :UTIL ↵
UTIL 7

*) DIR   :UTIL ↵
*) DELETE/V   # ↵

...(files deleted)...

*) DIR   : ↵
*) DELETE/V   UTIL ↵
DELETED UTIL

*) CREATE/DIR/HASH=17   UTIL ↵
```

- Mount the dump tape(s) and load the entire tape file.

The hash frame size can make a significant difference in access time to files in *any* directory that contains either a lot of files or very few files.

System Security

This section outlines some security issues, then tells you how to change the password of LOCK_CLI and disable the console break sequence.

From the standpoint of security, computer installations fall into two categories: *closed shop*, and *open shop*.

In a closed shop, few people have access to the hardware or system console. Users (including applications programmers) work on consoles in a separate area. The CPU, disk and tape units, and even line printers are out of bounds to all but a few carefully screened and trained people. In such shops, system operators are on duty most of the time; they start applications programs, do dumps, bring the system up and down, etc.

In an open shop, the hardware and system console are less restricted. They may be open to programmers or even to nonusers. There may or may not be a system operator. Physical security is not nearly as tight as in a closed shop.

In any installation, anyone who can touch the CPU or disk units can flip a few switches and bring the system down. The only way to protect these is to lock them up and/or keep a reliable system operator on duty at all times. So, in an open shop, you must trust the people who have physical access to the hardware.

The closed/open shop issue is a management issue; we can't tell you which way to do it. But the files and information on the system may be even more important than the hardware — and we *can* offer some advice on this issue.

In terms of general access, AOS is a secure system. Only people who have user profiles can log onto the system; and only a privileged person can create user profiles. The default user profiles are nonprivileged; so the standard user *cannot* become privileged. AOS prevents any nonprivileged user from accessing any file unless the file's ACL allows this. The default ACLs do not allow it. So, if you go with the defaults, your system will be safe from casual trespass or file violation.

To maintain and enhance the security provided by AOS — in either shop — you need to observe a few precautions. The major ones involve

- Super privileges. Any Superuser has access to all files on the system. Superusers can change their own profile or — less obviously — learn the operator's or other privileged user's password.
- Other privileges, like change username, may allow a user to *become* a privileged user. Users with the access devices privilege can bypass AOS, access disk data directly, and perhaps destroy disk file structure. Other potentially dangerous privileges are shown in Table 13-1.
- Remote access. Users on remote systems, or modems, can do anything local users can do if they have the privileges in their local profiles.
- The system console. If the master CLI or any unlocked son CLI is running, it has all privileges, and is open to anyone who has access to the system console. You can eliminate this danger by running LOCK_CLI as a matter of course. The system still remains vulnerable to the break sequence, unless you disable this with the CPU LOCK switch or, on an MV/8000, the SCP command LOCK.

  Possibly, in an open shop or even in a closed one, you may want to curtail the system operator's powers at this console.

You can deal with the first three issues — super privileges, other potentially dangerous privileges, and remote access — by giving all local users standard profiles. Take the PREDITOR default value *[NO]* on all these privileges. Keep privileged passwords private.

Retain the original ACLs on all AOS files, as shipped by DG. Give read and/or execute access only to those files that you want the user(s) to read; any user with execute access can see what's in a program by running it in a debugger.

To safeguard the system console, there are two issues involved: LOCK_CLI, and the break sequence (for machines without data switches).

LOCK_CLI and Its Password

LOCK_CLI is a special, lockable CLI. Once locked, it does not recognize privileged commands, or commands that might displace it. No one can terminate it without unlocking it; and no one can unlock it unless he or she knows the password. The operation of LOCK_CLI is described in Chapter 9. But how to set the password is described only here.

When to Use LOCK_CLI

In an open shop, the system console may be open to users or the general public. If an operator is not always on duty, you may want to run LOCK_CLI to protect the system. Some organizations would run it because there *was* no official operator, or if they wanted to limit the operator's powers.

Changing the Password

LOCK_CLI, like the standard CLI, consists of two files, LOCK_CLI.PR and LOCK_CLI.OL. It is in directory :UTIL.

The password supplied with LOCK_CLI is PASSWORD, but you should change this if you want to run LOCK_CLI. To do it, you must edit file LOCK_CLI.PR with the DEDIT disk file editor, putting the new password in location PASSW, and ending the password with a null (ASCII 0). Up to 31 filename characters, including spaces, are allowed, but they must all be uppercase.

The following example shows you how to change the original password (PASSWORD) to MAGIC.

Choose a display console. If this is a user console, log on with a privileged profile.

If this is the system console and it is running a locked LOCK_CLI, unlock it by typing UNLOCK↵, then PASSWORD↵. If you are in a normal CLI, proceed.

Press the ALPHA LOCK key on the console. Then type the following:

) SUPERUSER ON ↵

*) DIR :UTIL ↵

*) XEQ DEDIT/S=LOCK_CLI.ST LOCK_CLI.PR ↵

AOS FILE EDITOR

\+ PASSW: *050101 +* ↵

Type PASSW: to see an octal display of the first two characters of the current password. After the DEDIT prompt (+ sign), press ↵.

\+ MODE A ↵

Type MODE A ↵ for the ASCII display.

\+ PASSW: *PA +* "M*400+"A <CR>

Open location PASSW again to see what the values are in ASCII. The first two characters are PA. Type a quotation mark ("); type the first character of the new password (in this case, M); type *400 (this puts the M in the left byte). Next, type + ; type a second quotation mark; and then type the second character of the new password (A). Finally, press the CR key to close this location and to open the next location.

PASSW+1 :SS + "G*400+"I <CR>

Type a quotation mark; type the third character of the new password (G); type *400. Next, type + ; type a quotation mark; and then type the fourth character (I). Finally, press CR to close this location and to open the next one.

PASSW+2 :WO + "C*400 ↵

C is the last letter in the example password. To put the C in the left byte, you must multiply by 400 as shown. The syntax is

"char*400

(For a password with an even number of characters, after the last character you must type 0 (null) to end the password, and press ↵ to close the location.)

\+ PASSW: *MA +* <CR>

When done with the password, go back and verify it. The first two characters are okay. Press the CR key.

PASSW+1 *:GI +* <CR>	The second two characters are okay. Press the CR key.
PASSW+2 *:C<000> +*	And the last two characters (C and the final null) are okay.
BYE ↵	Type BYE and press ↵ to leave DEDIT.
*) XEQ LOCK_CLI ↵	Back to the CLI. Execute LOCK_CLI...
AOS CLI REV n ...	
) SUPERUSER ON ↵	Try Superuser...
)	LOCK_CLI ignores the command.
) UNLOCK ↵	Type UNLOCK...
MAGIC ↵	and the new password (doesn't echo).
) SUPERUSER ON ↵	See if it's unlocked...
*) LOCK ↵	Yes, success. Lock it again.
)	(Lock command turns off Superuser.)

If you cannot make the new password work, log onto a user console with a privileged profile, turn Superuser on, and check :UTIL:LOCK_CLI.PR, locations PASSW, again; then try the UNLOCK again.

Note that file LOCK_CLI.PR should have a null ACL, but file LOCK_CLI.OL should have an ACL of OP,R so this CLI can read its own overlay file. The files are shipped with these ACLs, but you can assign them as follows:

```
*) ACL/K   LOCK_CLI.PR ↵
*) ACL/V   LOCK_CLI.PR ↵
LOCK_CLI.PR
```

(The CLI displays nothing, which means that the ACL is null.)

```
*) ACL   LOCK_CLI.OL   OP,R ↵
*) ACL/V   LOCK_CLI.OL ↵
LOCK_CLI.OL   OP,R
*)
```

(The CLI displays an ACL of OP,R.)

Now, with the new password and correct ACLs, LOCK_CLI can safeguard the system console, user processes, and the system itself. To run it automatically, edit the UP.CLI macro as described under LOCK_CLI in Chapter 9.

The people who operate the system don't absolutely *need* to know LOCK_CLI's password. If they don't know it, they won't be able to shut the system down normally from the system console — which might be what you want. (Anyone who has the Superuser privilege can get the Superprocess privilege, and terminate the LOCK_CLI or master CLI from a *user* console — so if the operator has a privileged profile, he can do this.)

In any case, the fewer people who know the password the better — because anyone who knows it can UNLOCK, read or modify files, or turn on Superuser and take control.

LOCK_CLI retains the username OP, so it can issue commands to EXEC while locked. Thus, the operator can issue EXEC commands even while restricted to a locked CLI.

Disabling the Break Sequence on an MV/8000

Even with LOCK_CLI running, someone can enter the break sequence, halt the CPU, and cause delays while ESD or FIXUP is run. Even worse, the person could run ESD, reboot AOS, and come up in the master CLI, with all its powers.

You can prevent this by disabling the break sequence on an MV/8000. This is useful only when the system console is separate from the CPU, because anyone can create the effect of a break sequence with CPU switches.

How to Adapt This Book for Restricted System Operators

If your system operators have few privileges, there are some parts of this book you may not want them to read. If so, you can remove certain parts of the book and give the operator(s) the rest. Generally, the parts to remove are

Chapters 2 and 3	If you don't want the operator to bring up the first AOS system.
Chapter 4	If you don't want the operator to run AOSGEN.
Chapter 5	If you don't want the operator to run PREDITOR or create the multiuser environment.
Chapter 7	If you don't want the operator to run PREDITOR.
Chapter 14	This chapter, if you don't want the operator to know about the LOCK_CLI password. Also, if you plan to remove chapters, it would be tactful to remove this one.

Routine Procedures

This section touches on some things you might want to consider doing regularly to keep things running well.

- Backup procedures. These are very important and ideally they will be done each day. You can use the DUMP command, the PCOPY program, or both. Chapter 10 shows some CLI macros that can make these procedures relatively painless.
- Log files. SYSLOG can be very helpful in terms of user accounting information, and as a soft/hard error log. You might start it as a matter of course from the UP.CLI macro. The log files will grow rapidly, so someone should delete them when they're no longer needed.
- Preventative maintenance. This can help safeguard your hardware, and keep your whole system running smoothly.

How to Handle Updates and New Revisions from DG

DG continually improves its software products and updates its technical manuals. It sends the improved products and manuals to all customers who are on the *Software Subscription Service*.

So, if you subscribe to this service, you will periodically get tapes and new manuals. Generally, you should make an effort to install the updated software. In nearly all cases, it will run your existing applications even better than your current revision.

Each AOS revision is one integer number greater than the last; the revisions go 6.00, 7.00, 8.00, and so on. DG may issue multiple AOS *updates* for each revision. The updates are numbered on the right of the decimal point; for example, for revision 7.00, the updates go 7.01, 7.02, 7.03, 7.04, and so on.

AOS Updates

AOS updates consist of patches (needed changes to one or more program disk files), shipped on magnetic tape or diskettes. When you receive an AOS update, load it from file 0 of the tape, or from the diskette; then print the update notice, as follows.

) SUPERUSER ON ↵	
*) DIR : ↵	
*) DELETE/2=IGNORE PATCH_FILES ↵	(List file contains old patch file names.)
*) LOAD/V/DELETE/L=PATCH_FILES @MTxn:0 ↵	(x is B, C, or A; n is the unit number.)
.	
.	(verifies old files deleted and new files loaded)
.	
*) QPRINT AOS_UPDATE_NOTICE ↵	(Print — or use TYPE, without a line printer — the update notice file.)

Now, read the notice and apply the needed patches using the PATCH utility. This is described in Chapter 4, near the end.

Loading a New AOS Revision

Each AOS revision includes new versions of *all AOS program and support files,* on an AOS system tape. A revision also includes a printed *Release Notice* — also supplied as a disk file in :UTIL, form RELEASEnn.nn. A revision does *not* include updates. If you receive a revision and update at the same time, load the revision first, then the update.

If, after receiving a new AOS revision, you want to load *all* the new AOS files, follow the steps described next. The steps advise you to install the new starter system — overwriting the current installed system, if any — but you can always install your new tailored system over this with little effort.

To start, read the Release Notice, "Notes and Warnings" section, to check for any interrevision incompatibilities. The Release Notice also names all directories and files on the new AOS system tape. Generally, the only files shipped on an AOS system tape are those in the root (tape file 6), and directories :UTIL, :SYSGEN, and :HELP (tape file 7).

By default, each file on the tape will overwrite its older version on your LDU (if there is an older version). So you will want to protect certain files during the update. The files you will want to protect include UP.CLI, NEWERMES.CLI, and any other DG-supplied files that you have tailored for your system.

So, From AOS, turn the PERMANENCE characteristic on in the root and :UTIL for the tailored files that you don't want overwritten with new ones. For example:

```
DIR :UTIL ↵
PERMANENCE NEWERMES.CLI ON ↵
PERMANENCE UP.CLI+ ON ↵
```

Shut down AOS.

You are now ready to load from the update media. If you are loading from tape, mount the tape on unit 0 (device code 22) and continue to the next section. If you are loading from diskettes, skip to the section, "Loading from Diskettes."

Loading from Tape

Boot from tape and install the new bootstraps and starter system. How you do this depends on what type of computer you have.

- If you have an S/20, S/120 or an S/280, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type 22H. For example

  ! 22H (S/20, S/120, S/280)

- If you have an S/140, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type 11A. The system will display a 6-digit number, after which you type 100022 ↵. After the next *!* prompt, type 100022L. For example

  ! 11A *xxxxxx* 100022 ↵ (S/140)
  ! 100022L

- If you have an MV/8000, the system console displays an *SCP-CLI* prompt. Next to the prompt, type RESET ↵. Then, after the next prompt, type BOOT 22 ↵. For example

  SCP-CLI> RESET ↵ (MV/8000)
  SCP-CLI> BOOT 22 ↵

- If your computer has hardware data switches (numbered 0 (or X4/0) through 15), make sure they are set to 100022 (switches 0, 11, and 14 up, the others down). Lift and release the RESET switch; lift and release the PR LOAD (PROG LOAD) switch.

The step above program loads the tape bootstrap, TBOOT. TBOOT says

FROM MT-0: 3 ↵

AOS INSTALLER REV n
SPECIFY ALL UNITS IN THE LDU
DISK UNIT NAME? DPF0 ↵ (type master LDU unit name.)
DEVICE CODE? ↵
(Enter all the units in the LDU.)

--DISK BOOTSTRAP INSTALLED
INSTALL A SYSTEM BOOTSTRAP? Y ↵

FROM MAGTAPE (M) OR DISKETTE (D)? M ↵
FROM MT0, FILE #: 4 ↵

--SYSTEM BOOTSTRAP INSTALLED
INSTALL A SYSTEM? Y ↵

FROM MAGTAPE (M) OR DISKETTE (D)? M ↵
FROM MT0, FILE #? 5 ↵

--SYSTEM INSTALLED

DONE!

Now, bring up the new starter system and specify an initial load. Again, how you do this depends on what type of computer you have.

- If you have an S/20, S/120, or and S/280, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type nnH where nn is the device code. For example

  ! 27H (S/20, S/120, S/280)

- If you have an S/140, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type 11A. The system will display a 6-digit number, after which you type 1000nn ↵, where nn is the device code. After the next *!* prompt, type 1000nnL. For example

  ! 11A *xxxxxx* *100033* ↵ (S/140)
  ! 100033L

- If you have an MV/8000, the system console displays an *SCP-CLI>* prompt. Next to the prompt, type RESET ↵. Then, after the next prompt, type BOOT 27 ↵. For example

  SCP-CLI> RESET ↵
  SCP-CLI> BOOT 27 ↵ (Device code of master LDU unit.)

- If your computer has hardware data switches (numbered 0 (or X4/0) through 15), make sure they are set to 1000nn, where nn is the device code. For code 24, set switches 0, 11, and 13 up, the others down. For code 27, set switches 0, 11, 13, 14, and 15 up, the others down. For code 33, set switches 0, 11, 12, 14, and 15 up, the others down. Lift and release the RESET switch; left and release the PR LOAD (PROG LOAD) switch.

The step above brings the *disk* bootstrap into memory. The system bootstrap says

SPECIFY EACH DISK IN THE LDU
DISK UNIT NAME? DPF0 ↵ (Master LDU unit name.)
DEVICE CODE? ↵
SYSTEM PATHNAME? ↵
DATE (MM/DD/YY)? (type correct date)
TIME (HH:MM:SS)? (type correct time)

OVERRIDE DEFAULT SPECS? Y ↵
MASTER LDU: xxxxxx
NUMBER OF BUFFERS... ↵
SWAP FILE DEFINITION? ↵
INITIAL LOAD [N] Y ↵
FROM MTA0, FILE #: [6] ↵

The system will try to load the new files from tape file 6 into the root directory. For each file that already exists in the root, it will ask whether or not you want to replace the old copy. Since you are updating, answer Y ↵. For example

ERROR 26
FILE GHOST.PR
DELETE OLD COPY? Y ↵

When you've finished, the AOS CLI prompt will come up on the system console. Since you are updating, use the LOAD command with the /DELETE switch, which automatically replaces each file with the revised version, where this applies. For example

) SUPERU ON ↵
*) LOAD/V/DELETE @MTA0:7 ↵ (Use @MTA0:7 with tape on an MTB or MTC unit)

DELETED...
DELETED...
DELETED...

*) REWIND @MTA0 ↵
*)

Now dismount the tape and skip to the section, "Generate a New AOS System."

Loading from Diskettes

If you are loading from diskettes, follow these steps:

1. With any system but a Model 10/SP, skip to step 2. With a Model 10/SP, the diskette device code is 20 and the device name is DPM0. Insert the diskette marked "BOOTABLE D200 EMULATOR" in unit 0, and type 20H to boot from that unit. When the *!* prompt reappears, remove the diskette from the unit.

2. For DPI and DPK disks, set the switch so that the diskette unit is unit 0. Mount the first AOS revision diskette in unit 0. (To install an update, skip to step 34.)

3. Boot from disk. The system asks

   SPECIFY EACH DISK IN THE LDU
   DISK UNIT NAME?

4. Type the diskette unit name; for example, DPI0 ↵.

5. *DEVICE CODE?* ↵ (Or, if nonstandard, type the diskette device code and ↵.)

6. *SYSTEM PATHNAME?* INSTL ↵ (Filename of Installer.)

   AOS INSTALLER REV n

   .
   .
   .

   ENTER ALL UNITS IN LDU
   DISK UNIT NAME?

7. Remove AOS diskette number 1 from the unit and insert AOS diskette number 2. (Diskette 2 has an installable SYSBOOT on it.)

8. Type the unit name of your disk (not diskette). For example

   DPI1 ↵

9. *DEVICE CODE?* ↵ (Or, if nonstandard, type the diskette device code and ↵)

   --DISK BOOTSTRAP INSTALLED

10. *INSTALL A SYSTEM BOOTSTRAP?* Y ↵

11. *FROM MAG TAPE (M) OR DISKETTE (D)?* D ↵

    DISKETTE UNIT NAME?

12. Type the diskette unit name; for example DPI0 ↵

    (Installer copies SYSBOOT from the diskette to disk.)

    --SYSTEM BOOTSTRAP INSTALLED
    INSTALL A SYSTEM?

13. Remove AOS diskette number 2 and insert AOS diskette number 3 (this contains the starter system).

14. Type Y ↵

15. *FROM MAG TAPE (M) OR DISKETTE(D)?* D ↵

    DISKETTE UNIT NAME?

16. Type the diskette unit name; for example DPI0 ⮐

    (Installer copies AOS system from diskette 3 to disk.)

    -- SYSTEM INSTALLED
    DONE!

17. For DPI and DPK disks set the switch so the nonremovable disk is unit 0. Bootstrap from the hard disk as usual (e.g., by typing 26H or 100033L). It asks

    SPECIFY EACH DISK IN THE LDU
    DISK UNIT NAME?

18. Type the correct disk unit name; e.g., DPI0 ⮐

19. *DEVICE CODE?* ⮐

20. *SYSTEM PATHNAME?* ⮐ (Specify installed system.)

    AOS REV n

21. *DATE (MM/DD/YY) ?* (Type correct date.)

22. *TIME (HH:MM:SS) ?* (Type correct time.)

23. *OVERRIDE DEFAULT SPECS?* Y ⮐ (Must type Y ⮐ to update.)

24. *NUMBER OF BUFFERS IN CACHE [default]* ⮐

25. *SWAP FILE DEFINITION [default]* ⮐

26. *INITIAL LOAD [N]* Y ⮐ (Must type Y ⮐ to update.)

    FILE # [MTA0:6]?

27. Remove AOS diskette number 3 and insert AOS diskette number 4.

28. Next to the *FILE...* question, type the diskette unit name; for example, DPI1 ⮐

29. The system now copies files from the diskette to the hard disk. For each file, the system asks if you want to delete the old copy. Confirm by pressing ⮐. For example

    ERROR 26 (Means "File already exists".)

    FILE: GHOST.PR
    DELETE OLD COPY [Y] ? ⮐

    If any of the files to be deleted have the PERMANENCE characteristic on, the system asks if you want to delete the permanent file. Confirm by pressing ⮐. For example

    ERROR 324 (Cannot delete a permanent file.)

    FILE: ERMES
    DELETE PERMANENT FILE [Y] ? ⮐

30. When all files have been loaded from diskette, the system displays

    AOS CLI REV n

31.) SUPERUSER ON ⮐ (Turn Superuser on.)

    *)

32. Remove the AOS diskette from its unit and replace it with the next AOS diskette.

33. Type

    *) LOAD/V/DELETE   diskette-name ⤶   (For example, LOAD/V/DELETE @DPI1 ⤶)

    .
    . (The system loads files.)
    .
    *)

    Repeat these steps (31 and 32) until you've loaded the last AOS diskette.

34. Next, if you received one or more update diskettes, load them the same way you loaded the AOS system diskettes in the previous step.

35. For a computer other than a Model 10/SP, skip to step 36. For model 10/SP, you must copy a D200 emulator file to the hard disk. The default emulators are on a diskette labeled "D200 EMULATOR." For background information and instructions, see the section, "Installing the Terminal Emulator", in Chapter 4.

36. Now, remove the last diskette and store it with the other diskettes in a safe place.

Proceed to generate a tailored AOS system as described in the next section.

Generate a New AOS System

You are now ready to generate a new AOS system *from your old system specifications file* file. This is easy; for example:

*) SEARCH :UTIL ⤶
*) DIR :SYSGEN ⤶
*) XEQ AOSGEN/DEFAULT=oldsysname ⤶

AOSGEN now comes up. You need only name the new system (e.g. SYS_5.00) and build it (N and B commands). All specs except the name will be the same as the old system's.

When the CLI prompt returns, patch the system as described near the end of Chapter 4.

Then shut down the starter system and boot the tailored system by name.

*) BYE ⤶
DO YOU REALLY WANT TO ...   Y ⤶

What you do next depends on what type of computer you have:

- If you have an S/20, S/120, S/280, or a DESKTOP GENERATION system, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type nnH, where nn is the device code. For example,

  ! 27H   (S/20, S/120, S/280, or a DESKTOP GENERATION system)

- If you have an S/140, the system console displays a *!* prompt (if not, turn power off and on again). Next to the prompt, type 11A. The system will display a 6-digit number, after which you type 1000nn ⤶, where nn is the device code. Next, after the *!* prompt, type 1000nnL. For example,

  ! 11A   *xxxxxx*   100033 ⤶   (S/140)
  ! 100033L

- If you have an MV/8000, the system console displays an *SCP-CLI* prompt. Next to the prompt, type RESET ⤶. Then, after the next prompt, type
  BOOT 27 ⤶. For example,

  SCP-CLI> RESET ⤶   (MV/8000)
  SCP-CLI> BOOT 27 ⤶

- If your computer has hardware data switches (numbered 0 (or X4/0) through 15), make sure they are set to 1000nn, where nn is the device code. For code 24, set switches 0, 11, and 13 up, the others down. For code 27, set switches 0, 11, 13, 14, and 15 up, the others down. For code 33, set switches 0, 11, 12, 14, and 15 up, the others down. Lift and release the RESET switch; left and release the PR LOAD (PROG LOAD) switch.

The step above brings the *disk* bootstrap into memory. The system bootstrap says

SYSTEM PATHNAME? SYSGEN:newsysname.SY ↓
date
time

OVERRIDE DEFAULT SPECS [N] ↓

After the new system comes up, build a new ERMES by typing

```
) SUPERU   ON ↓
*) DIR   :UTIL ↓
*) NEWERMES ↓
.
.
.
*)
```

then bring up EXEC and the multiuser environment:

```
*) :UP ↓
.
.
.
```

Turn PERMANENCE off for the files you made permanent (if you leave permanence on, you won't be able to text edit these files).

```
*) DIR   :UTIL ↓
*) PERM   NEWERMES.CLI   OFF ↓
*) PERM   UP.CLI+   OFF ↓
*)
```

If there were no errors patch the new system and make a system tape (as described in Chapter 4) and you're done. LDUs that you've built with the older Disk Formatter will work perfectly with the new AOS software (such programs are designed to be revision-independent).

If, for any reason, you wish to reload the *old revision*, get out the old AOS system tape or diskette and follow the procedure above.

As you can see, the update procedure isn't difficult. Restrictions and possible problems with it are as follows.

- It assumes that you haven't reconfigured the directory structure shipped by DG. For example, if the old revision of SED is not in directory :UTIL, the new revision will not overwrite it.
- Your master LDU cannot include disks other than those on the first DPF or DPI controller, because the starter system supports only one of these controllers.
- The new LOCK_CLI, with the original password, PASSWORD, will overwrite the old one; so, if you care about LOCK_CLI, you must edit its program file and insert the desired password (covered earlier in this chapter).

Note that you *can* load new files selectively; for example, LOAD/V/DEL @MTA0:7 :UTIL:SED.PR↓. But this is not recommended because the error message (ERMES) file may differ between the revisions. Better to do the whole thing.

Updating Your Manuals

DG ships the Release Notice both in printed form and as a disk file. Updates to your *manuals* are provided only as disk files. The pathnames are :UTIL:sss_pppppp_rr, where sss is the series, pppppp is the part number, and rr is the revision. For example, for manual 093-000122-05, the pathname is

:UTIL:093_000122_05

To keep your manuals up to date, we suggest that you examine these files, print the ones you want, and make corrections in the pertinent manuals. Then — if you need the disk space — delete the files. You can always reload any or all of them, if needed, from the AOS system tape or diskette.

Getting Help from DG

Generally, when a customer buys a DG system, the price includes a certain period of engineer/support time, time on the Software Subscription Service, and perhaps training courses.

After the initial period has expired, customers must renew service contracts to assure continued support.

In the United States, support people work from the Atlanta Service Center; and, depending on your contract, you can call this center for help.

Outside the US, the support people are often DG system engineers; your system engineer can give you more information on support.

A Real System's UP Macro

An UP macro can give you an overview of an entire AOS operating system: how it configures its devices; what queues are available to users; how many user consoles and virtual consoles it permits; and what products it runs.

Figure 14-1 shows a real system's UP macro. This macro:

- defines its own switches;
- creates a system banner (Lion, followed by the system and system revision);
- starts SYSLOG;
- initializes LDUs and sets their ACLs;
- runs the QCMP queue compacter;
- starts up EXEC and continues batch streams;
- starts an INFOS II process;
- enables local and virtual user consoles;
- brings up the network;
- sets type and priority for PID 2; and
- runs LOCK_CLI.

Five macros called by this UP.CLI are not shown. They are

- UP.LPB.CLI
- SYSLOG_UP.CLI (which resembles the one shown in Chapter 9)
- UP_NETWORK.CLI (supplied with the XODIAC networking system and tailored for this installation)
- CEO.SYSTEM.CLI
- CEO.FSA.CLI (supplied with DG's Comprehensive Electronic Office (CEO) system

```
[!EQUAL,([!NEQUAL,002,[!PID]]ERROR[!END]%\NC\NOQCMP\NOINFDS&
\NOCONTINUE\IVERIFY\NOCEO\NONET%),()]
   SEARCHLIST  :UTIL
   PUSH
   WRITE [!READ [!ASCII 207] Press NEW LINE when printers are aligned. ]
   SUPERUSER  ON
   DELETE/2=IGNORE   :REV
   REVISION/L=REV  :GHOST.PR
   STRING  Lion  [!SYSTEM]  [REV]
   SYSID  [!STRING]
   DIRECTORY  :
   ACL  @NULL   +,WARE
   ACL  @MTB-   +,WARE
   WRITE  Initializing  DPF30 as :UDD
   INITIALIZE/1=WARNING/2=WARNING    @DPF30
   PUSH
    DIRECTORY   :MAGIC
    WRITE Initializing DPF31 as  :UPDATE.40
    INITIALIZE/1=WARNING/2=WARNING    @DPF31
    ACL UPDATE.40   $+,,  ATLANTA,,  OP,OWARE  +,RE
    WRITE  Initializing  DPF20 as  :UPDATE.50
    INITIALIZE/1=WARNING/2=WARNING    @DPF20
    ACL UPDATE.50 $+,,  ATLANTA,,  OP,OWARE +,RE
   POP
   WRITE Initializing  DPF1 as :PENDING
   INITIALIZE/1=WARNING/2=WARNING     @DPF1
   WRITE Initializing  DPF11 as  :INFOS
   INITIALIZE/1=WARNING/2=WARNING     @DPF11
   ACL  UDD  OP,DWARE  +,E
   :SYSLOG_UP
   [!NEQUAL,(/NOQCMP),(%/NOQCMP%)]
     WRITE Compacting the queues...
     XEQ  QCMP/YES
     [!END]
   SEARCHLIST [!SEARCH], :UTIL:CEO_DIR
   PROCESS/DEFAULT/DIRECTORY=@/NAME=EXEC  EXEC
   PAUSE  2
   CONTROL  @EXEC  PROMPTS  OFF
   CONTROL  @EXEC  QPRIORITY  (2,3)   0   250
   CONTROL  @EXEC  QPRIORITY    1     1   254
   CONTROL  @EXEC  QPRIORITY    4     255 255
   [!NEQUAL,(/NOCONTINUE),(%/NOCONTINUE%)]
     CONTROL  @EXEC  CONTINUE
     CONTROL  @EXEC  SILENCE
     DIRECTORY  :UTIL
     UP.LPB.CLI
     [!END]
  CONTROL  @EXEC  SILENCE  (@LPB,@LPB1)
  PAUSE  3
  POP
  PROMPT  TIME  CHECKTERMS
```

Figure 14-1. A Tailored UP.CLI Macro (continues)

```
  [!NEQUAL,(/NOINFOS),(%/NOINFOS%)]
    WRITE  Bringing up INFOS II now...
    DIRECTORY  :INFOS
    [!EQUAL,(/IVERIFY),(%/IVERIFY%)]
       PUSH
       SEARCHLIST  :INFOS:UTIL
       DIRECTORY  :UDD:PENDING
       PAUSE  3
       XEQ   :INFOS:UTIL:IVERIFY     PENDINGINFO.DB
       POP
       [!END]
     PROCESS/DEFAULT/DIR=@/NAME=INFOS   :INFOS_II
     PAUSE  3
     DIRECTORY  :
     [!END]
  [!NEQUAL,(/NOCEO),(%/NOCEO%)]
    DIR  :UTIL:CEO_DIR
    CEO.SYSTEM  START  LOG
    CEO.FSA CHECKPOINTON 30
    DIR  :
    [!END]
  [!NEQUAL,(/NONET,%/NONET%)]
    :UP.NETWORK
    [!END]
  [!NEQUAL,(/NC),(%/NC%)]
    CONTROL  @EXEC  ENABLE/ALL
    PAUSE  10
    [!END]
  CONTROL @EXEC PRIORITY (@LPB, @LPB1)  1
  CONTROL  @EXEC  PROMPTS  ON
  SUPERUSER  ON
  PRTYPE  2  PREEMPTIBLE
  PRIORITY  2  1
  XEQ  :LOCK_CLI
  SUPERUSER  OFF
[!ELSE]
POP
[!ELSE]
WRITE *ERROR*
WRITE %0\% works only when invoked from the master CLI.
WRITE Required arguments: None
WRITE Optional switches:,,/NC - Do not enable consoles.
WRITE,,,,,,,,,,,,,,,,,,,,,/NOQCMP - Do not run QCMP.
WRITE,,,,,,,,,,,,,,,,,,,,,/NOINFOS - Do not PROC up INFOS II.
WRITE,,,,,,,,,,,,,,,,,,,,,/IVERIFY - Run IVERIFY before starting INFOS II.
WRITE,,,,,,,,,,,,,,,,,,,,,/NOCONTINUE - Do not continue batch or printers.
WRITE,,,,,,,,,,,,,,,,,,,,,/NOCEO - Do not start up CEO.
WRITE,,,,,,,,,,,,,,,,,,,,,/NONET - Do not bring up the network.

[!END]
```

Figure 14-1. A Tailored UP.CLI Macro (concluded)

What Next?

Chapters 1 through 5 of this book describe *bringing up* an AOS system from scratch. Chapters 6 through 14 (this one) gave the details of *running* (and some issues involved in managing) an AOS system.

At this point, you might want to check out the appendixes — or review earlier material.

End of Chapter

Appendix A
Important Errors and Error Messages

While bringing up or running your computer system, you may encounter certain serious errors. This appendix summarizes the error messages output from such errors. These messages always appear on the system console or computer panel lights.

Table A-1 describes the error messages that are printed on the system console (listed alphabetically). It explains the probable source (cause) of the error and one or more possible solutions.

Some error messages may consist only of a number, without a text explanation. This appendix also describes the most common of these — keyed by number from the system parameter file, PARU.SR — in Figure A-1. When AOS is up and running, you can use the CLI command MESSAGE to interpret these numbers. For example

) MESS 123 ↵
123 PHYSICAL UNIT OFF LINE
) MESS 63 ↵
63 DEVICE ALREADY IN USE
)

Power up fault (error) codes appear on the system console. Power *supply* codes may appear in the panel lights. The power up and power supply error codes appear in tables that follow Figure A-1.

Common errors that *users* may receive from EXEC or the system are covered in Chapter 8, Table 8-5.

The sections that follow Figure A-1 describe

- MV/8000 power-up fault codes (displayed on the system console); and
- S/280 power supply error codes (shown in the CPU panel lights).

Table A-1. Important Errors and What To Do About Them

Message	Source and Meaning	Action
(nothing)	At startup, there is no power to the CPU, console, disk, or tape units.	Power up the CPU, system console, disk units, and tape units (if needed).
	At startup, the system console is not on line (ON LINE light off), or too dim.	Put it on line (press the CMD and ON LINE keys), or check the brightness.
	At startup, you typed the wrong device code; there is no such device.	Wait 10 to 20 seconds; then, type the break sequence (press the CMD and BREAK/ESC keys). Retype the boot command.
	At startup, the bootstrap disk or tape unit is not ready or on line.	Make the unit(s) ready and/or put on line.
	At startup from tape, the tape unit is set to the wrong density, or it is not on line, or the the unit door is open.	The tape unit is off line. Put it back on line. Close the door (if any), and try (if any), and try again.
	At startup from diskette, The diskette is misinserted.	Remove the diskette. For DESKTOP GENERATION systems, reinsert the diskette with the write-enable notch up and the paper label facing you. For other diskette drives, reinsert the diskette with the paper label up.
	Input medium (disk or tape) is unreadable. (The first blocks of the medium are often required to load the program.) This could mean a blank or bad tape or disk.	Make sure the device is ready and/or on line. If so, perhaps the wrong disk or tape is mounted. If you are booting from tape (device code 22), file 0 of the tape must contain the program TBOOT. If you are booting from disk (code 27, 33, etc.) the first blocks of the disk must contain a disk bootstrap.
	At startup, bootstrap jumpers are not installed. On any locked CPU (lock switch), when you turn the power on, the CPU tries to program load from the device code selected with hardware jumpers. Usually these jumpers are inserted to the primary disk device (e.g., 27) by the DG engineer who installs the hardware. If these jumpers are not inserted, or specify a device code of a device that is not on line, the CPU cannot program load.	Make sure the system console and bootstrap device are on line as above. If they are on line, unlock the CPU, and turn the power ON and OFF again. The system console should display *!* or *BOOT DEVICE?* If so, the jumpers probably are not installed. You must program load manually (e.g., type 26H), as described in Chapter 6. To make cold starts easier later on, you might want to have a DG engineer install jumpers to the primary disk device code.

(continues)

Table A-1. Important Errors and What To Do About Them

Message	Source and Meaning	Action
	During normal AOS operations (power still on to the CPU and system console).	At the system console, type CTRL-Q to undo any CTRL-S that might have suspended display. If this works, fine. If it doesn't work, type CTRL-Q again and continue to the next recommended action. Check for a high-priority process (like a dump) that's using system resources. If the system console is a hard-copy console, check for a fault. Turn it off and on again, and/or replace the ribbon it it's worn out. Try typing CTRL-C CTRL-C on the system console. If the system console echoes ^C^C, AOS is probably okay and the problem is with one or more of the user processes. Warn all users of an impending shutdown; then bring the multiuser environment down (DOWN ↵) and up (UP ↵) again. If the system console doesn't react to CTRL-C CTRL-C, the system is deadlocked. Type the break sequence, then proceed as described in Chapter 6, "Abnormal Shutdown."
! (exclamation)	From programmed console.	If AOS was running, it is now frozen. To have it continue, type P; then ↵. If the CLI prompt doesn't return, run emergency shutdown: type I, and then 14R. If AOS was not running and now you want to restart it, do a warm start as shown in Chapter 6.
ABNORMAL SYSTEM SHUTDOWN	From AOS, after you have tried to shut it down or run ESD. This means that — for some reason — AOS could not close all files and release the nonmaster LDUs (if any were initialized). This problem can occur if you initialize a removable LDU (like a diskette) and remove it from its unit without releasing it, and then shut down.	Run Open File Report to see which LDUs have open files. Then run FIXUP on each LDU that has open files. Make sure that you release an LDU (RELEASE name ↵) properly before you shut down.

(continued)

Table A-1. Important Errors and What To Do About Them

Message	Source and Meaning	Action
ABORT -- message	From Disk Formatter, this may mean hard errors on the LDU.	Make sure the disk is write enabled; retry the Formatter. If it aborts again, contact your DG support organization.
	From the Installer, this means a fatal error.	Perhaps you specified the wrong file to the Installer or the disk is write-protected. Reboot. If you saw a *MUST RUN FIXUP* message, run FIXUP and reboot. Rerun the Installer. For more detail, see the error messages in Chapter 12, "The Installer".
	From FIXUP, the DIB block may be bad.	See *CANNOT* message in this table.
	From AOS during initialization. There may be a message or number code that explains it. If there is only a code, find it in Table A-2.	If the message contains the number 243, you must run FIXUP on the LDU before you can bring up AOS on it.
	From CLI or EXEC. A utility program hit a fatal error and couldn't continue. AOS stays up.	If the message allows you to correct the problem, do so. Otherwise, try to find the message text in this table.
AIR FLOW ALARM - CHECK FILTERS	From SCP on the system console. Air flow in the MV/8000 cabinet is inadequate. The main CPU keeps running.	Shut down AOS (if up) immediately; the temperature may soon reach a critical point. Correct the problem.
AN UNLABELED DISKETTE HAS BEEN INSERTED. DO YOU WANT TO LABEL THIS DISKETTE ? [N]	From the CLI, on a write command to a labeled diskette. The diskette you inserted is not labeled.	If you want to label the diskette, type Y ⤶. The CLI will then ask for the volume ID you want, and will display a default based on the label you specified for the first diskette. If you thought the diskette *was* labeled, or think you may have made a mistake and inserted a diskette with valuable data, press ⤶. Then remove the diskette and try to find and insert the diskette with the label you expected.
ATTEMPT TO ACCESS PROCESS NOT IN HIERARCHY	From AOS. Either the process doesn't exist or your command can't be executed because your process is not the father of the target process.	See if the process exists by typing ? ⤶. If it does exist, and you still want to execute the command, turn SUPERPROCESS on and try again.
BOOT TIMEOUT	From SCP EPROM. SCP EPROM couldn't program load from the device specified or from the code in jumpers with the CPU locked on power up.	If the disk or tape is not on line and ready, put it on line. If the message recurs, get the MV/8000 system tape and reload the microcode as described at the beginning of Chapter 3. Also see the error mesage under "nothing" at the beginning of this table.
C---	During MV/8000 power up, partial *CONSOLE READY* message. There are problems with one or more SCP or main processor components.	Make sure the diskette is properly inserted; or perhaps insert another copy of the diskette. Turn the CPU power off and on again. For interpretation, see Table A-2.
CALLER NOT PRIVILEGED FOR THIS ACTION	From the system. Your command requires a privilege or PID that your process lacks. For example, you need to turn SUPERUSER on to start or change SYSLOG or set the system date or time.	If you really want to have the command obeyed, either log on with a privileged profile and turn SUPERUSER on, or go to the system console and become PID 2, which has all privileges.

(continued)

Table A-1. Important Errors and What To Do About Them

Message	Source and Meaning	Action
CANNOT DELETE UNEXPIRED FILE ON LMT	From the system, on a write command to a labeled tape or diskette. The retention date for the fileset has not yet expired. The system cannot write to the tape/diskette set. The default retention date is 90 days and can be selected with the DUMP command and the /RETAIN= switch.	You can either select another tape or diskette set for the dump, or use this set. To use this set, you'll need to relabel each tape or diskette. To relabel tapes, use the LABEL utility. Then retry the write. To relabel diskettes, the easiest course is to turn auto-labeling on, using the commands:) OPERATOR OFF ↲) OPERATOR/LABEL ON ↲ This allows the CLI to relabel the diskettes, overwriting the old label. Then restart the dump. For a dump to either tape or diskette, you can shorten the retention period via the /RETAIN= switch, so this doesn't happen again.
CANNOT READ IN THE DIB FOR THE LDU	From FIXUP. The DIB (Disk Information Block) is unreadable.	There may be surface or hardware format damage. Rerun FIXUP from the beginning.
CAUTION - USTORE OR SPAD NOT LOADED CORRECTLY	From SCP on power up. The microcode or scratchpad memory hasn't been loaded correctly.	First, turn CPU power off, lock the CPU (if possible), and turn the power on again. This reloads the SCP-OS and microcode. If there are no errors, proceed with normal operations. If the error recurs, try another SCP-OS diskette. If this doesn't help, contact your DG support organization.
CHECKSUM BAD	From SCP on power up. The SCP operating system may have failed a checksum test.	Follow the steps described under the *CAUTION-USTORE*... message in this table.
CHECKSUM ERROR	From AOS or support program. The tape unit hardware couldn't read your tape.	Retry; if the error recurs, try another tape or a different unit (if possible). Sometimes, cleaning the tape or tape unit read/write heads with an alcohol-soaked cotton pad or swab will help.
CONTROL POINT DIRECTORY MAX SIZE EXCEEDED, FILE file	From the system. Your command cannot be completed because the directory *file* (or a Control Point Directory above *file*) has used all blocks allowed for it.	To recover, you will have to: expand the space available to this directory (via the SPACE command; e.g., SPACE XDIR 5000 ↲); or delete some files. If you expand a user directory (:UDD:username) this way, the expansion lasts until the person logs off. If the full directory is the system root (:) or any other directory of type LDU, you must delete some files, since expanding an LDU requires a full disk format.

(continued)

Table A-1. Important Errors and What To Do About Them

Message	Source and Meaning	Action
COULDN'T ACCESS CODE FOR MESSAGE (EXEC)	From EXEC. The :ERMES message file, from which most programs get the text for messages, has a bad ACL, is missing, or is invalid.	Make sure the ACL of file :ERMES includes +,RE. If this error recurs, rebuild ERMES as described in Chapter 5.
CPUID INDICATES LESS THAN n KB OF MEMORY	From AOS during system initialization. The main processor CPUID indicates too little memory for AOS.	Turn CPU power on and off again to reload microcode and correct the CPUID.
DIRECTORY ACCESS DENIED	From CLI. You do not have the required access privilege(s) to this file.	Turn Superuser on and retry. You may want to change the file ACL to give your username the needed privilege(s) so you can gain access to this file later without turning Superuser on.
DIRECTORY DELETE ERROR	From the system. The directory you tried to delete has subordinate directories.	You may not really want to delete this directory. Check all subdirectories in it by typing) F/TY=DIR/TY=CPD dir:# ↵ Then delete selectively. If you really want to delete the whole directory, repeat the deletion with the # template.
DIRECTORY IN USE --CANNOT DELETE	From the system. You cannot delete this directory because it's being used. Perhaps it's the working directory, initial directory, or it's in someone's search list.	If you really want to delete this directory, make sure it isn't your working directory or initial directory; have users remove it from their search lists; try the delete again.
DISK ERROR, status = n	From FIXUP, PCOPY, the Disk Formatter, or Installer.	Make sure the disk is write enabled. If this is not the problem, you might want to try a Disk Formatter Partial format on the LDU. One or more disks may need hardware formatting via DTOS, or there may be surface damage. See the *Peripherals* or other disk manual for status.
DISK IS IN USE FIXUP MUST BE RUN ON LDU	From SYSBOOT during bootstrapping.	Proceed, but when you see *SYSTEM PATHNAME*, type FIXUP ↵. Then run FIXUP on the LDU.
ERROR n	During power up.	See Table A-2 or A-3 (as pertinent) for the cause of the error. Run DTOS tests.
ERROR: n FILE:path	During AOS initialization; this is nonfatal.	From CLI, type MES n ↵. This describes the error.
*ERROR: ** message*	From SYSBOOT during AOS bootstrapping.	The *message* often indicates an inconsistent LDU. If so, respecify, or mount the correct disk(s) for the LDU.
EXEC NOT AVAILABLE	You tried a Q-series command and EXEC isn't running.	Bring EXEC up (UP ↵).
EXECUTE ACCESS DENIED	From CLI. You lack the required E privilege.	See *DIRECTORY ACCESS DENIED* message.
FATAL AOS ERROR: n	A software/hardware error *n*, has prevented AOS from continuing.	See Chapter 6, "Abnormal Termination". (Do a memory dump and run ESD.) In any case, note the error.

(continued)

Table A-1. Important Errors and What To Do About Them

Message	Source and Meaning	Action
FATAL DISK ERROR	From FIXUP. FIXUP probably hit a new bad block.	See the *FATAL DISK ERROR* message in the FIXUP table at the end of Chapter 6.
FATAL ERROR: message	From AOS, during system initialization.	If the message is a number, see Figure A-1 for an explanation of the number. If the *message* is text, try to find it in this table. Then try ESD via RESET and START 50 ⤶; if ESD fails, run FIXUP.
FILE ACCESS DENIED	From CLI. You lack the required Read or Execute privilege.	See *DIRECTORY ACCESS DENIED* message.
FILE ALREADY EXISTS: file	From the system, during load or move command. The file already exists in the specified directory.	If you want to move the file into this directory, repeat the command with the /RECENT switch (to keep the most recent version of the file) or the /DELETE switch (to delete the file in the destination directory and replace it regardless of the creation date).
FILE ALREADY EXISTS: file REPLACE OLD COPY?	From the system, while installing a new revision of AOS.	Type Y ⤶ to replace the older version of the file.
FILE DELETED...	From FIXUP. FIXUP had to delete the file.	Restore it from backup media. FIXUP messages at the end of Chapter 6 give details.
FILE DOES NOT EXIST	From FIXUP or Installer.	Try adding the @ prefix to the devicename or respecify tape file number.
	From AOS starter system.	Specify unit @MTA0 for tape or the correct diskette unit name. The starter system recognizes only MTA0 as a tape unit.
	From CLI or EXEC. The program can't find the file.	Correct your searchlist or enter the full file pathname; use FILES and PATH commands to check.
FILE INACCESSIBLE, RUN FIXUP ON THE LDU	From the system. It found a file open that should have been closed.	FIXUP should be run on this LDU. As soon as it is practical, release the LDU (or shut down, if the file is on the master LDU). Then run FIXUP on it.
FILE IS EXCLUSIVELY OPENED, CAN'T OPEN file	From the CLI, during a command that reads a file (often the DUMP command). A program has the file exclusively opened, which means the system can't read it.	If you really need to have the file read (for example, you need to have it backed up), abort the command, shut down the program (like text editor or data management program) normally, then retry the command. Normally, the multiuser environment should be down *before* you do system backups.

(continued)

Table A-1. Important Errors and What To Do About Them

Message	Source and Meaning	Action
FILE RENAMED...	From FIXUP. FIXUP had to rename the file.	See this message in the FIXUP error table (end of Chapter 6) for action.
FIXUP CHECKSUM ERROR	From FIXUP. This means that the FIXUP disk file is unstable.	Load FIXUP from your AOS system tape (device code 22, file 1) or from AOS diskette number 1 (pathname FIXUP). Later, from AOS, you must replace the bad FIXUP file. From tape, type) LOAD/V/D @MTx0 FIXUP ↵ From diskette, start with the second dump file diskette (diskette number 4). Type) LOAD/V/D diskette FIXUP ↵ Continue to load diskettes. From either tape or diskettes, wait for confirmation that the original file has been loaded (deleting the bad version).
FROM DISK BOOT: message	The *message* may give the cause; e.g., SYSBOOT not installed.	Install a system bootstrap via the Installer if this is the problem, and retry. Otherwise, note the error code in the *Peripherals* manual.
FROM PID n: (name)	IPC message from another program; e.g., EXEC.	If *name* is EXEC, see the message here in this table or in Table 8-4/8-5. If *name* is another program, see that program's associated manual.
FROM SYSTEM: message	AOS has found a serious error.	Find the *message* in this table.
HARD ERROR	From AOS or support program. A hard error occurred on the specified device (the program tried a read/write operation 16 times without success).	This can occur if a disk goes off line, is shut off, or is write disabled. If so, put it back on line, or write enable it; try to continue normally (if necessary, run ESD or FIXUP and continue). If the device is a disk and it was write enabled, the problem may be a new bad block on the disk; see the Status code in the correct section of the *Peripherals* manual. If the code means a bad sector, the disk has developed new bad block(s). Run a Disk Formatter Partial format, with read-only analysis, on the LDU. If the Formatter reports one or several new bad blocks, this means that new bad blocks probably caused the problem. Have the Formatter update the bad block table, run FIXUP if necessary, and continue. If the Formatter finds many new bad blocks, or none, do not update the bad block table; diagnostics will probably be needed. Consult your DG support organization.

(continued)

Table A-1. Important Errors and What To Do About Them

Message	Source and Meaning	Action
HARD ERROR (continued)		For a hard error on tape, the cause is often a bad tape; it may be dirty or worn, or, on a read, it may have been written by a defective unit. Try another tape if possible; or try cleaning the tape heads with a cotton swab. If you're sure the unit hardware is at fault, consult your DG support organization. For a diskette, remove the diskette; reinsert it, and try again. If the error recurs, the diskette may be unusable. You *can* try a Disk Formatter Partial or Full format on it. For a hard error on any other device, call your DG support organization, and try to run without the device until it is fixed.
HARD ERROR WRITING TO SYSTEM OVERLAY AREA	The system cannot write to the first disk in the master LDU.	Perhaps the disk is not write enabled. If so, turn the write enable switch to "on" and restart AOS.
HARD INTERRUPT FROM board	From SCP. A CPU hardware problem that has halted the main CPU. AOS, if it was up, is frozen.	Record the whole message and contact your DG support organization.
ILLEGAL PROCESS PRIORITY	From the system. You tried to change the priority of a process, but the father process does not have the privilege to change process priorities.	If you need to change the priority of this process, execute PREDITOR and give the privilege to change process priorities to the parent process. Or, log on the system console (PID 2), which has all privileges, and change priority.
ILLEGAL PROCESS TYPE	From the system. You tried to create a resident or preemptible process, but the father process does not have the privilege to change process types.	If you need to change the type of this process, execute PREDITOR and give the privilege to change the process type to the parent process. Or, log on the system console (PID 2), which has all privileges, and change the process type.
INCONSISTENT DIB INFO - DISK FORMAT INCORRECT	From Disk Formatter Partial. This disk's DIB (Disk Information Block) is not what the Formatter expected. Causes may be 1. The disk doesn't belong to the LDU at all; 2. The *sequence* in which you specified the disk is wrong; 3. The revision and/or sequence numbers are invalid.	Mount (if not mounted) and specify the correct disks; or specify the disks in the order in which they were originally Full formatted. If you do these things and the message recurs, the DIB may have been damaged; a Full Format may be needed.
INDECIPHERABLE DUMP FORMAT	From AOS CLI, after CLI LOAD command. The cause may be 1. This is an unlabeled tape and you tried to treat it as labeled (e.g., LOAD/V TAPE:FILE ↵); 2. The file was not dumped to the tape (it may have been written with COPY or by a user program; 3. The file was dumped with a nonstandard (e.g., /BUFF=8192) and you tried to simply LOAD it.	Decide on the cause and correct it. 1. Find the correct tape or try to load it by file number; 2. Try a) COPY/V @NULL & ↵ &) unit:file-number ↵ 3. Check with) LOAD/N/BUFF=n & ↵ &) @unit:file-number ↵ (n is usually 4096 or 8192).

(continued)

Table A-1. Important Errors and What To Do About Them

Message	Source and Meaning	Action
INFINITE xxx FAULT	From SCP. A hardware page or protection fault occurred while the main CPU was trying to process such a fault.	AOS is frozen. Consult your DG support organization.
INTERNAL CONSISTENCY ERROR IN EXEC ... HAVE EVERYONE LOGOFF AND THEN TERMINATE THIS EXEC	From EXEC. EXEC has problems.	Send warning messages to all users; disable all consoles (CX DISABLE). Then, when you are ready to submit an STR, see Chapter 10. Type) TERM/B=file OP:EXEC ↓, and then type UP ↓. If you don't care to submit an STR, type DOWN ↓. Then try typing UP ↓ again.
INVALID REMOTE USERNAME-PASSWORD PAIR - FILE :NET:	From the system. Your remote username/password pair on the remote system differs from that on the local system. The username and password must be the same on both systems for the network and CEO to work. Also, for a CEO user, there must be a CEO profile with a user ID identical to the AOS username on both systems.	Someone on the remote system (or on your system) must change the username and/or password so that they match on both (or all) systems. For network operations of this kind to work, a user's PREDITOR profile also needs the privilege ACCESS LOCAL RESOURCES FROM REMOTE MACHINES.
INVALID USERNAME-PASSWORD PAIR	From the system. If you see this message while trying to log on to a remote system, it means either that there is no profile with the username and password that you typed — or — there is such a profile but it doesn't specify the required privilege for network operations. You can also get this message during a local logon.	Someone on the remote system must create the profile. The profile must include the privilege USE VIRTUAL CONSOLE if the user will log on across a network, or the USE MODEM privilege if the user will dial up.
LD RELEASED, MUST HAVE FIXUP RUN ON IT	From AOS. A file on the LDU just released could not be closed.	Run FIXUP on the LDU.
LOGICAL DISK IN USE, CANNOT RELEASE	From AOS. AOS thinks someone is using the LDU. Perhaps a user with a directory there is logged on, or the directory is in an active user's search list.	Check users and get them to log off as needed; then retry the RELEASE command. If the message recurs, perhaps a batch or print job requires files on the LDU. Check with QDISPLAY. If the system shutdown can't close the LDU, the shutdown will be abnormal (explained in the message *ABNORMAL SYSTEM SHUTDOWN*).
@LPBx COOPERATIVE TERMINATED	From EXEC. EXEC could not maintain the line printer co-operative process, XLPT.	Unless you stopped the printer, see if it is on line — if not, use EXEC commands to start and continue it (see UP.CLI macro for syntax).
@LPB PHYSICAL UNIT OFFLINE	From EXEC. The printer is not on line. The XLPT terminates.	Put the printer on line and use EXEC commands to start and continue it (see UP.CLI macro for syntax.)

(continued)

Table A-1. Important Errors and What To Do About Them

Message	Source and Meaning	Action
NETWORK NOT AVAILABLE	From AOS. Either the network isn't up or the command may require the File Transfer Agent (/FTA switch on the MOVE command).	If you used the MOVE command, retry it with the /FTA switch; otherwise, check for network processes (X.25, etc.) and bring up the network (macro UP.NETWORK.CLI).
NO OPERATOR AVAILABLE	From the CLI. Your command requires an operator (human or CLI), but there is none available.	Type OPERATOR ON ⮐ and retry the command that caused the error. If typing OPERATOR ON ⮐ eliminates the error, this means you tried to use labeled diskettes (pathname @LFD:volid:filename), without turning your CLI operator mode on. Now that you've turned operator mode on, you can proceed with the labeled diskette access. (But if you want to label many diskettes, it's easier to let the CLI do it for you. Turn operator off and turn it on again with the /LABEL switch. Using the CLI command OPERATOR is further described in Chapter 9 and the CLI manual. If turning the CLI OPERATOR command on doesn't eliminate the error, this means that some operation requires a person to be on duty at the system console. To tell the system that a person is available, use the EXEC command OPERATOR. Type) CX OPERATOR ON ⮐ on the system console. Then — from a user console — reissue the command that caused the error. Someone must be present at the system console (to mount tapes, etc.) for this command to work.
OVER TEMP ALARM -- SYSTEM GOING DOWN	From SCP. The temperature in the MV/8000 cabinet has reached the critical point; the SCP is cutting CPU power to prevent damage.	Check for the problem (clogged filters, faulty fans, etc.); try to fix and reboot. FIXUP will be needed on all initialized LDUs.
PHYSICAL UNIT FAILURE	From AOS CLI, SYSBOOT, or AOS utility.	See the *HARD ERROR* message in this table.
PHYSICAL UNIT OFFLINE	From AOS. The tape or diskette unit you specified isn't on line.	For a tape, check the ON LINE status light (if any) and use the ON LINE switch if needed. On a diskette, make sure the diskette is inserted correctly: with the paper label up, or, on a DESKTOP GENERATION system, with the write-enable notch up, and the paper label toward you.

(continued)

Table A-1. Important Errors and What To Do About Them

Message	Source and Meaning	Action
PLEASE INSERT A DISKETTE. UNIT [default] VOLUME ID [default]	From CLI, on labeled diskette access. The CLI is ready to read or write another diskette, or the diskette is full (on a write).	Remove any diskette from this unit. If the diskette lacks a paper label with volid, filename, and date information, be sure to label it. (Use a felt-tipped pen if the blank label is already on the diskette.) Then, insert the next diskette to be read from or written to, and press ↲.
PROFILE NOT FOUND	From EXEC. Someone typed a Q-series command, yet lacks a user profile.	Even the system operator needs a profile to issue Q-series commands. Check for a profile with PREDITOR, creating one if needed (Chapter 5).
READ ACCESS DENIED	From CLI. You lack the needed Read access privilege.	See *DIRECTORY ACCESS DENIED* message.
RUN FIXUP OVER ALL DISKS EVER INIT'ED INTO THIS LDU	From AOS. One or more LDUs that are (or have been) part of this LDU may have inconsistent information on it.	Run FIXUP over all LDUs (including all removable disk pack LDUs and diskettes) that have been initialized into your computer system. (Don't initialize them first.) Run FIXUP even if FIXUP says that fixing is not necessary. For example, you would do this on a diskette LDU that you'd removed from its slot without releasing it.
SBUS message	From SCP. There are problems with the SBUS, which connects the two CPUs. SCP log entries from this point on may not be reliable.	Record the message. Soon, you should shut down AOS and contact your DG support organization.
SOFT ERROR:	From AOS. A soft error occurred on the device (system retried it up to 15 times, and it worked).	If the device is a disk (not diskette), this may be serious. Note the error, and, if it recurs, release the disk or shut AOS down (if needed); ideally, you should run a Disk Formatter Full format on it. If the device is a tape or diskette, some some errors while writing are normal. Frequent soft errors while writing may mean that: the unit heads need cleaning, the medium or unit is wearing out, or improper storate is degrading the medium.
SPECIFIED FILE DOES NOT EXIST	From SYSBOOT during AOS bootstrapping.	Retype the system pathname. It often is in :SYSGEN — and you must include the .SY suffix. If you can't determine the pathname, try pressing ↲ for the installed system; then check filenames from the CLI. Post the system name on the console so this doesn't happen again.
SYSTEM BOOTSTRAP TOO LARGE	From the Installer. The tape or diskette file you specified is too big for its reserved area on the LDU. This probably means you specified the wrong tape file or used the wrong diskette.	If you get the message *MUST RUN FIXUP*, run FIXUP on the LDU. Rerun the Installer and type the correct file number (4 for system bootstrap, 5 for the AOS system), or use diskette 2 for the system bootstrap and diskette 3 for the AOS system. Make sure the tape/diskette is an AOS system tape/diskette.

(continued)

Table A-1. Important Errors and What To Do About Them

Message	Source and Meaning	Action
THE LABEL ON THIS DISKETTE IS NOT THE LABEL REQUESTED. INSERTED:x REQUESTED:y	From the CLI, on labeled diskette access. The diskette you inserted does not have the label the CLI expects. On a dump, the CLI asks if you want to relabel the diskette.	On a dump, if you think you may have made a mistake and inserted a diskette that holds valuable data, press ↓, and then find and insert the correct diskette. If you want the CLI to relabel the diskette (which destroys all information on it and on any subsequent diskettes, if it is part of a sequence), type Y ↓; then, accept the default or type the volid.
THERE HAS BEEN A POWERFAIL *NOW RESTARTING DEVICE n UNIT n*	From AOS auto restart routine. The ac power failed; but the backup battery maintained power to the CPU. The restart routine tries to restart all disks, and restores AOS status at the power fail.	If AOS cannot restart a device, it tells you so. If the device is a disk, run FIXUP on it; for a different device, you may need to shut down and reboot AOS. Regardless, any tape write to an MTA or MTB unit will need restarting and printers will need to be placed on line.
UID ON LAST DISK DOESN'T MATCH	From SYSBOOT during AOS bootstrapping.	The last disk unit you specifed holds a disk that isn't part of the LDU. Mount (if needed) and specify the correct disk.
UNKNOWN ERROR CODE n	From any AOS program.	If you think *n* is a valid error code, the system ERMES file may need rebuilding (Chapter 5).
WARNING message	From the CLI or EXEC. The program continues, but cannot obey the message.	If the message allows you to solve the problem, do so; otherwise, try to find the message text in this table.
WRITE ACCESS DENIED	From the CLI. You lack the required Write access privilege.	See *DIRECTORY ACCESS DENIED* message.
ZERO LENGTH FILENAME SPECIFIED	Your program or command line specified a filename of zero length. This can happen if you accidentally insert a space in a pathname, for example:) X DISPLAY MYDIR: MYFILE ↓	Retype the command to eliminate the space.

(concluded)

AOS Numeric Error Codes

While you are bringing up AOS, or running the stand-among Disk Formatter, error messages may appear as numeric codes, without text explanation. This happens because the active program doesn't have access to the ERMES message file.

Should it happen, you can find the number, and its meaning, in Figure A-1. The first column contains error mnemonic, the second the number, and the third an explanation. For a complete list of AOS numeric codes, see file :UTIL:PARU.SR or the *Advanced Operating System (AOS) Programmer's Manual.*

```
.
ERMEM=   5     ;INSUFFICIENT MEMORY IS AVAILABLE
.
ERDCH=   15    ;DATA CHANNEL MAP IS FULL
.
ERSPC=   21    ;FILE SPACE EXHAUSTED
.
ERDDE=   23    ;DIRECTORY DOES NOT EXIST
ERIFC=   24    ;ILLEGAL FILENAME CHARACTER
ERFDE=   25    ;FILE DOES NOT EXIST
ERNAE=   26    ;FILE NAME ALREADY EXISTS
.
EREOF=   30    ;END OF FILE
.
ERWAD=   32    ;WRITE ACCESS DENIED
ERRAD=   33    ;READ ACCESS DENIED
ERAWD=   34    ;APPEND AND/OR WRITE ACCESS DENIED
.
ERNSW=   43    ;SWAP FILE SPACE EXHAUSTED
ERIBS=   44    ;DEVICE ALREADY IN SYSTEM
ERDNM=   45    ;ILLEGAL DEVICE CODE
.
ERPRN=   51    ;NUMBER OF PROCESSES EXCEEDS MAX
.
ERDAI=   63    ;DEVICE ALREADY IN USE
.
ERPAR=   70    ;PARITY ERROR
EREXC=   71    ;RESIDENT PROCESS TRIED TO CREATE SON AND BLOCK
ERNDR=   72    ;NOT A DIRECTORY
.
ERSNM=   74    ;TOO MANY SUBORDINATE (son) PROCESSES
ERFIL=   75    ;FILE READ ERROR
ERDTO=   76    ;DEVICE TIMEOUT
.
ERPRV=   102   ;CALLER NOT PRIVILEGED FOR THIS ACTION
.
ERPRH=   107   ;ATTEMPT TO ACCESS PROCESS NOT IN HIERARCHY
ERBLR=   110   ;ATTEMPT TO BLOCK UNBLOCKABLE PROCESS
.
ERGES=   112   ;ATTEMPT TO START MULTIPLE AGENTS
ERCIU=   113   ;CHANNEL IN USE
ERICB=   114   ;NOT ENOUGH  CONTIGUOUS DISK BLOCKS
ERSTO=   115   ;STACK OVERFLOW
ERIBM=   116   ;INCONSISTENT BIT MAP DATA
```

Figure A-1. AOS System Error Codes (Selected) (continues)

```
.
ERPUF=    121       ;PHYSICAL UNIT FAILURE (hard error)
ERPWL=    122       ;PHYSICAL WRITE LOCK
ERUOL=    123       ;PHYSICAL UNIT OFFLINE
.
ERMIS=    126       ;DISK AND FILE SYSTEM REVISION NUMBERS DON'T MATCH
ERIDD=    127       ;INCONSISTENT DEVICE INFORMATION BLOCK (DIB)  DATA
ERILD=    130       ;INCONSISTENT LOGICAL DISK
ERIDU=    131       ;INCOMPLETE LOGICAL DISK
.
ERVIU=    134       ;LOGICAL DISK IN USE, CANNOT RELEASE
.
ERARG=    141       ;TOO MANY OR TOO FEW ARGUMENTS TO PMGR
.
ERICN=    147       ;ILLEGAL CHANNEL
.
ERACL=    167       ;ILLEGAL ACL
.
ERFPU=    172       ;FPU HARDWARE NOT INSTALLED
.
ERDCT=    175       ;DISCONNECT ERROR ON MODEM
.
ERSNI=    177       ;SYSTEM NOT INSTALLED
ERLVL=    200       ;MAX DIRECTORY TREE DEPTH EXCEEDED
.
ERRDL=    202       ;RESOURCE DEADLOCK
ERE01=    203       ;FILE IS OPEN, CAN'T EXCLUSIVELY OPEN
ERE02=    204       ;FILE IS EXCLUSIVELY OPENED, CAN'T OPEN
ERIPD=    205       ;INITIALIZATION PRIVILEGE DENIED
.
ERCPD=    237       ;CONTROL POINT DIRECTORY MAX SIZE EXCEEDED
ERNSD=    240       ;SYSTEM  OR BOOTSTRAP DISK NOT PART OF MASTER LD
ERUSY=    241       ;UNIVERSAL SYSTEM, YOU CAN'T DO THAT
EREAD=    242       ;EXECUTE ACCESS DENIED
ERFIX=    243       ;CAN'T INIT LD, RUN FIXUP ON IT
ERFAD=    244       ;FILE ACCESS DENIED
ERDAD=    245       ;DIRECTORY ACCESS DENIED
.
ERLRF=    253       ;RESOURCE LOAD OR RELEASE FAILURE
ERNNL=    254       ;ZERO LENGTH FILENAME SPECIFIED
```

Figure A-1. AOS System Error Codes (Selected) (concluded)

Power Up Fault Codes

This section describes power up fault codes for MV/8000 computers.

Power *up* fault codes indicate faulty CPU components, as detected by EPROM code when you turn power on. A power up fault code is a partial power up message (with one or more characters missing). The power up message is

CONSOLE READY (on MV/8000 systems)

Table A-2 lists the MV/8000 partial power up fault messages.

You can take remedial action for only a few messages (e.g., you can clear the filters if the problem is air flow). But your DG support organization will find the information useful when deciding what material to bring to your site.

Table A-2. MV/8000 Power Up Faults (Partial) ***CONSOLE READY*** Message

Message	Faulty Component(s)
(none)	SCP (MBC CPU and/or TTO circuit boards).
_	SCP (MBC CPU circuit board).
C_	SCP (MBC TTO circuit board).
CO_	CC circuit board.
CON_	Microsequencer or CC circuit board — in order of probability.
CONS_	CC circuit board.
CONSO_	Power supply.
CONSOL_	Power supply (the temperature is too high and the SCP won't allow it to operate).
CONSOLE_	Power supply (air flow is restricted, and the SCP won't allow it to operate).
CONSOLE R_	SCP (MBC RAM circuit board).
CONSOLE RE_	CC circuit board, ATU circuit board, or Microsequencer circuit board — in order of probability.
CONSOLE REA_	Microsequencer circuit board.
CONSOLE READ_	SCP (diskette controller board).

S/280 Power Supply Fault Codes

On the S/280 computer front panel, there are three lights: PWR (power), BATT (battery), and RUN. Normally, when the POWER switch is ON, one or more of these glows steadily, as follows:

PWR light — lit when dc power is normal; off when power is off or the computer is under partial battery backup.

BATT light — lit when computer has transferred from normal power to backup battery (full or partial backup).

RUN light — lit when the computer is executing instructions (AOS, DTOS, etc.); off when the computer is halted.

When the computer is running AOS on normal power, the PWR and RUN lights are lit. When the POWER switch is OFF, all lignts are off.

When one or more lights blink, at about 1 hertz, this indicates a power system or fan fault. The lights will continue blinking until someone turns the POWER switch to OFF or until ac power goes down. In some cases, the power supply will cut power to the entire CPU in 15 seconds to prevent damage.

If you see one or more panel lights blink, note and record the code promptly, in case ac power goes down. The code can help you (and your DG support organization) decide what hardware (if any) needs attention.

The meaning of the blinking lights is shown in Table A-3.

Table A-3. S/280 Power Supply Faults (Blinking Lights)

Blinking Light(s)	Meaning
RUN only	Environment fault. This may be any of the following. · The ac line input voltage is either too high or too low (brownout condition). · There is an overtemperature, either within the power supply or near it (ambient temperature too high).
BATT only	Fan failure. One or more cooling fans is not working. The supply shuts down.
RUN and BATT	VNR or battery backup hardware fault. There is a problem with the battery or its hardware.
PWR only	Power supply fault. An undervoltage condition exists. This may result from a broken supply or a short circuit. The supply shuts down.
PWR and RUN	Overvoltage fault. Output voltage(s) are above safe operating limits. The power supply shuts down immediately. It stays down until someone presses power off and on again.
PWR and BATT	Overcurrent fault. There is a current overload or short circuit. The power supply shuts down.
PWR, BATT, RUN (all)	UPSC (Universal power supply) fault on power up. Normally on power up, all lights glow for about a second; this is a test. · If the lights blink, this means that the USPC self-test found a checksum error in the EPROM code. It hangs without powering up. · If the lights do not blink, this means that the ac-line voltage is too low (power up during brownout). When ac voltage reaches normal, the power-up sequence will resume.

End of Appendix

Appendix B
AOS Peripheral Devices

This appendix lists the standard device names, models, and codes for AOS peripherals. For a full list of AOS peripherals, consult your Data General representative.

AOS Device and Unit Names

During system initialization, the system writes the name of each peripheral device in the peripheral directory (:PER). These standard device names are not reserved exclusively for these devices; you can assign these names to files in directories other than the peripheral directory.

Table B-1 lists the default mnemonics for all AOS devices. Certain AOS programs ask you to specify device or unit names. For consoles and line printers, these names are *device names*. For disks, magnetic tapes, and multiprocessor communications adapters, these names are *unit names*.

Table B-1. Device and Unit Names (Default Mnemonics)

Name	Device
@CON0, @CON1, ... @CONn	First and succeeding CRTs, hard-copy consoles, teletypewriters, or asynchronous communications lines.
@CRA, @CRA1	First and second card reader controllers.
@DPx0, @DPx1, ... @DPxn*	Moving-head disk models.
@DKB0, @DKB1, ... @DKBn	Fixed-head disks running on DKB controllers (see Table B-2).
@LPA, @LPA1	Programmed I/O line printers.
@LPB, @LPB1	Data channel line printers.
@LPC, @LPC1	Non-data channel LP2 printers.
@LPD, @LPD1	Data channel LP2 line printers.
@MTA0, @MTA1, ... @MTA7	First 800-bpi magnetic tape unit controller.
@MTA10, @MTA11, ... @MTA17	Second 800-bpi magnetic tape unit controller.
@MTB0, @MTB1, ..., @MTB7	First model 6026 800/1600-bpi mag tape controller.
@MTB10, @MTB11, ..., @MTB17	Second model 6026 800/1600-bpi mag tape controller.
@MTC0, @MTC1, ..., @MTC3	First mag tape controller: model 6123 or model 6125 1600-bpi "streaming" or model 6230 or 6231 6400-bpi cartridge tape; or model 6270 6400-bpi cartridge tape; or model 4307 1600/6250 tape.

*Replace 'x' with the appropriate letter for the disk controller. For example, if you are using the first model 6060 disk unit on your system, replace 'x' with an 'F' and use the name @DPF0. Table B-2 lists disk model numbers and their corresponding controller names.

Table B-1. Device and Unit Names (Default Mnemonics)

Name	Device
@MCAR, @MCAR1	First and second multiprocessor communications adapter receiver.
@MCAT, @MCAT1	First and second multiprocessor communications adapter transmitter.
@PLA, @PLA1	First and second digital plotter controllers.
@SLN0, @SLN1, ... @SLNn	Synchronous communications lines (i.e., Hamlet).
@TPA, @TPA1	First and second paper tape punch controllers.
@TRA, @TRA1	First and second paper tape reader controllers.

AOS Magnetic Peripherals

AOS magnetic peripherals include standard disk, diskette, and tape models. A standard *disk model* includes a disk controller, one to four disk units, and one to four physical disks. A standard *diskette model* includes a diskette controller, one to four diskette units, and one to four physical diskettes. A standard *tape model* includes a tape controller and one to eight tape units. *Controllers* are the hardware boards that control disk, diskette, or tape units. They control how and when the unit reads from and writes to media. *Units* are the drives that hold media.

Table B-2 below lists the standard AOS magnetic peripherals and their controller names. *Please note*: Each model number represents a category of models; Table B-3 lists each model number in a category.

Table B-2. Standard AOS Magnetic Peripherals and Controller Names

Model Number	Controller Names	Description
4231	DPE, DPE1	Moving-head controller for removable disks that can control up to four units. Each unit can hold a 92-Mbyte disk.
4307	MTC, MTC1	Tape controller that can control up to four 1600/6250-bpi units.
6021	MTA, MTA1	Tape controller that can control up to seven units. Each unit can hold an 800-bpi tape.
6026	MTB, MTB1	Tape controller that can control up to seven units. Each unit can hold a 800/1600-bpi tape.
6030	DPD, DPD1	Diskette controller that can control up to four diskette units. Each unit can hold a 0.3-Mbyte diskette.
6045*	DPD, DPD1	Moving-head controller for top-loading disk cartridges that can control up to four units. Each unit contains two drives. The top drive holds a 5-Mbyte removable disk cartridge. The bottom drive holds a 5-Mbyte nonremovable disk.
6060	DPF, DPF1	Moving-head controller for freestanding disk units with removable packs. Can support up to four disk units. Depending on category (Table A-3), each unit holds one 50-, 96-, 192-, or 277-Mbyte pack.
6063	DKB, DKB1	Fixed-head disk controller that can run up to four units. Each unit holds a 1- or 2-Mbyte disk.
6070	DPG, DPG1	Same as 6045, but each drive has a 10-Mbyte disk, and the controller does not run diskette units.

(continues)

*The only controller that can have more than four unit numbers is model 6045. Units 4 through 7 (and 14 through 17 on the second controller) apply only to this model.

Table B-2. Standard AOS Magnetic Peripherals and Controller Names

Model Number	Controller Names	Description
6098	DPI, DPI1	Moving-head controller for a disk and diskette unit. The disk unit holds a 12.5-Mbyte nonremovable disk. The diskette unit can hold a 1.26-Mbyte (high-density) diskette.
6099	DPI, DPI1	Same as 6098, but without the diskette unit.
6100	DPI, DPI1	Same as 6098, but the disk unit holds a 25-Mbyte nonremovable disk.
6101	DPK, DPK1	A 12.5-Mbyte disk unit with a 1.26-Mbyte diskette unit.
6102	DPK, DPK1	Same as 6101, but without a diskette unit.
6103	DPI, DPI1	Same as 6098, but the disk unit holds a 25-Mbyte nonremovable disk and there is no diskette unit.
6104	DPK, DPK1	A 25-Mbyte disk unit with a 1.26-Mbyte diskette unit.
6105	DPK, DPK1	Same as 6104, but without a diskette unit.
6123, 6125	MTC, MTC1	Tape controller for 1600-bpi tape units (with reels side by side). Each controller can control up to four units.
6160	DPF, DPF1	Moving-head controller for nonremovable disks (mounted in a cabinet). Can support up to two disk units. Each unit holds a 43- or 147-Mbyte disk.
6220	DPK, DPK1	Same as 6220-D, but without the diskette unit.
6222	DPK, DPK1	Same as 6222-D, but without the diskette unit.
6220-D	DPK, DPK1	A 5-Mbyte disk unit with a 1.26-Mbyte diskette unit.
6222-D	DPK, DPK1	A 15-Mbyte disk unit with a 1.26-Mbyte diskette unit.
6224	DPL, DPL1	A 15-Mbyte disk that runs on the burst multiplexor channel (BMC).
6225	DPI, DPI1	Same as 6225-D, but without the diskette unit.
6225-D	DPI, DPI1	A 5-Mbyte disk unit with a 1.26 Mbyte diskette unit.
6227	DPI, DPI1	Same as 6227-D, but without the diskette unit.
6227-D	DPI, DPI1	A 15-Mbyte disk unit with a 1.26 Mbyte diskette unit.
6230, 6231	MTC, MTC1	A 6400-bpi cartridge tape unit.
6234	DPI, DPI1	A moving-head disk unit that holds a 50-Mbyte disk.
6236	DPJ, DPJ1	Moving-head controller for a nonremovable disk. The disk holds 354-Mbytes.
6267, 6268	DPM, DPM1	A 5-1/4 inch diskette unit, capacity 368 Mbytes, for DESKTOP GENERATION systems.
6270	MTC, MTC1	A 6400-bpi cartridge tape unit (DESKTOP GENERATION systems).
6271	DPN0, DPN1	A compact, 15-Mbyte disk unit (DESKTOP GENERATION systems).
6280	DPL0, DPL10	A 50-Mbyte disk unit that runs on the burst multiplexor channel (BMC).
6301	DPN0, DPN1	A compact, 38.6 Mbyte disk unit (DESKTOP GENERATION systems).
6336	DPN0, DPN1	A compact, 71.2 Mbyte disk unit (DESKTOP GENERATION systems).

(concluded)

Table B-3. AOS Model Categories

Category	Model Numbers	Description
4231	4231, 4231A, or 4231B	92-Mbyte moving-head disks.
4307	4307 or 4307-A	1600/6250-bpi tapes.
6021	6021 or 6023	800-bpi tapes.
6026	6026, 6026-A, or 6027	800/1600-bpi tapes.
6030	6030, 6030-A, 6030-B, 6031, 6031-A, or 6031-B	0.3-Mbyte diskettes.
6045	6045, 6046, 6047, 6048, 6050, 6050-F, 4234, 4235, 4236, or 4234-F	5-Mbyte removable disk cartridges with 5-Mbyte nonremovable disks.
6060	6060, 6060-A, or 6060-H	96-Mbyte moving-head disks.
	6061, 6061-A, or 6061-H	190-Mbyte moving-head disks.
	6067, 6067-A, or 6067-H	50-Mbyte moving-head disks.
	6122-A or 6122 (H assumed)	277-Mbyte moving-head disks.
6160	6160, 6160-A, or 6160-H	73-Mbyte moving-head disks.
	6161, 6161-A, or 6161-H	147-Mbyte moving-head disks.
6063	6063, 6063-A, or 6063-H	1-Mbyte fixed-head disks.
	6064, 6064-A, or 6064-H	2-Mbyte fixed-head disks.
	6066, 6066-A, or 6066-H	4-Mbyte fixed-head disks.
6070	6070, 6070-A, 6070-B, 6070-C, or 6070-D	10-Mbyte removable disk cartridges with 10-Mbyte nonremovable disks.
6098	6098	12.5-Mbyte nonremovable disk and a 1.26-Mbyte, quad-density diskette subsystem.
6099	6099 or 6099-A	12.5-Mbyte nonremovable disk.
6100	6100	25-Mbyte nonremovable disk and a 1.26-Mbyte, quad-density diskette subsystem.
6102	6102, 6105 6222	12.5-, 25-, or 15-Mbyte nonremovable disk with 1.26-Mbyte diskette.
6103	6103 or 6103-A	25-Mbyte nonremovable disk.
6224	6224, 6280	15- or 50-Mbyte disk that runs on the burst multiplexor channel (BMC).
6231	6231, 6270	Tape controller for 6400-bpi cartridge tape unit.
6234	6234	50-Mbyte moving-head disk.
6236	6236, 6237	354-Mbyte nonremovable moving-head disk.
6239	6239, 6240, 6290	592-Mbyte nonremovable moving-head disk.
6267	6267, 6268	5-1/4 inch, 368-Kbyte diskettes.
6271	6271	15-Mbyte disk (DESKTOP GENERATION systems).
6301	6301	38.6-Mbyte disk (DESKTOP GENERATION systems).
6336	6336	71.2-Mbyte disk (DESKTOP GENERATION systems).

Models without a suffix or with an 'H' suffix include a high-speed channel controller, a drive adapter, and the specified model. Models with an 'A' suffix are add-on disks. Other suffixes simply specify different model types.

AOS Device Codes

Each controller typically supports a maximum of four disk drives. Each group of four is associated with a single device code and four disk unit names. Table B-4 groups default controller names, device codes, and disk unit names. Whenever you bootstrap the system or a utility program, you are asked to supply a disk unit name. Then you type one of the disk unit names listed below. If the unit's controller is on its default device code (shown below), you do not have to supply a device code; the system associates default disk unit names with the default device codes. If you want to use a nondefault device code, consult the Data General Field Engineering Dispatch Center.

Table B-4. Default Controller Names, Device Codes, and Unit Names

Controller Name	Device Code	Disk Unit Name(s)
DKB	26	DKB0, DKB1, DKB2, or DKB3
DKB1	66	DKB10, DKB11, DKB12, or DKB13
DPD	33	DPD0 / DPD4, DPD1 / DPD5, DPD2 / DPD6, DPD3 / DPD7
DPD1	73	DPD10 / DPD14, DPD11 / DPD15, DPD12 / DPD16, DPD13 / DPD17
DPE	33	DPE0, DPE1, DPE2, or DPE3
DPE1	73	DPE10, DPE11, DPE12, or DPE13
DPF	27	DPF0, DPF1, DPF2, or DPF3
DPF1	67	DPF10, DPF11, DPF12, or DPF13
DPG	33	DPG0 / DPG4, DPG1 / DPG5, DPG2 / DPG6, DPG3 / DPG5
DPG1	73	DPG10 / DPG14, DPG11 / DPG15, DPG12 / DPG16, DPG13 / DPG17
DPI	33	DPI0 and DPI1*
DPI1	73	DPI0 and DPI1*
DPJ	24	DPJ0, DPJ1, DPJ2, or DPJ3
DPJ1	64	DPJ10, DPJ11, DPJ12, or DPJ13
DPK	26	DPK0 and DPK1*
DPK1	66	DPK10 and DPK11*
DPL	25	DPL0
DPL1	65	DPL10
DPM	20	DPM0, DPM1
DPM1	60	DPM10, DPM11
DPM	20	DPM0
DPN	26	DPN0, DPN1
DPN1	66	DPN10, DPN11

*The hard diskette and disk names are determined by the position of the toggle switch behind the disk front panel. Normally the hard disk is unit 0.

Table B-5 lists the standard I/O device codes for AOS. You use these mnemonics to identify a device through AOS.

Table B-5. Standard I/O Device Codes

Device Code	Mnemonic	Device Name
01	---	Writable control store.
02	---	Error checking and correction.
03	---	Memory allocation and protection.
06	MCAT	First multiprocessor communications adapter transmitter.
07	MCAR	First multiprocessor communications adapter receiver.
10	CON0	System Console input.
11	CON0	System Console output.
12	TRA	Paper tape reader.
13	TPA	Paper tape punch.
14	---	Real-time clock.
15	PLA	Incremental plotter.
16	CRA	Card reader.
17	LPA LPB LPC LPD	First line printer.
22	MTA MTB MTC	First magnetic tape controller.
24	DPJ	First moving-head disk controller.
25	DPL	First moving-head disk controller.
26	DKB	First fixed-head disk controller.
	DPK DPN	First moving-head disk controller.
27	DPF	First moving-head disk controller.
33	DPD DPE DPG DPI	First moving-head disk controller (except model 6060).
34	---	First communications system controller (multiplexor).
44	---	Second communications system controller (multiplexor).
46	MCAT1	Second multiprocessor communications adapter transmitter.
47	MCAR1	Second multiprocessor communications adapter receiver.
50	---	Second Intelligent Asynchronous Controller (IAC).

(continues)

Table B-5. Standard I/O Device Codes

Device Code	Mnemonic	Device Name
52	TRA1	Second paper tape reader.
53	TPA1	Second paper tape punch.
54	---	Second real-time clock.
55	PLA1	Second incremental plotter.
56	CRA1	Second card reader.
57	LPA1 LPB1 LPC1 LPD1	Second line printer.
62	MTA1 MTB1 MTC1	Second magnetic tape controller.
64	DPJ1	Second moving-head disk controller.
65	DPL1	Second moving-head disk controller.
	---	First Intelligent Asynchronous Controller (IAC).
66	DKB1	Second fixed-head disk controller.
67	DPF1	Second moving-head disk controller.
73	DPD1 DPE1 DPG1 DPI1	Second disk controller.
77	---	Central processor.

(concluded)

End of Appendix

Appendix C
EXEC Command Summary

This appendix summarizes all EXEC commands, alphabetically, in Table C-1. You can abbreviate any EXEC command to its shortest identifiable string.

Start each EXEC command with CONTROL @EXEC (CX macro).

Table C-1. EXEC Commands, Alphabetically

EXEC Command	What It Does	Example
ALIGN	Stops or continues line printer. Use it for aligning paper.	) CX ALIGN @LPB ↵) CX ALIGN/CONT @LPB ↵
BINARY	Enables binary mode (for word processor printing) on a device.	) CX PAUSE @CON25 ↵) CX BINARY @CON25 CLEANUP ↵) CX CONT @CON25 ↵
BRIEF	Makes batch or spool messages brief.	) CX BRIEF @LPB↵
CANCEL	Cancels a batch or printing request that is enqueued but not yet active. Use FLUSH for an active request.	) CX CANCEL 445 ↵
CLOSE	Closes a queue to user requests.	) CX CLOSE PLT ↵
CONSOLESTATUS	Displays console-user status.	) CX CONSOLES @CON8 ↵
CONTINUE	Continues (resumes processing in) a batch stream (1-4) or device (@LPB). Use after PAUSE.	) CX CONTINUE 2 ↵) CX CONTINUE @LPB ↵
CPL	Changes the maximum number of printed characters per line.	) CX PAUSE @LPB ↵) CX CPL @LPB 85 ↵
CREATE	Creates a device queue.	) CX CREATE PRINT QFILE ↵
DEFAULTFORMS	Sets new printer CPL and LPP parameters.	) CX DEFAULT @LPB FFILE ↵
DELETE	Deletes a device queue.	) CX DELETE QFILE ↵
DISABLE	Disables one or more user consoles for user logon. Use it when you when you plan to shut down or to free an EXEC-owned console for another DG product.	) CX DISABLE @CON8 ↵
DISMOUNTED	Tells EXEC that you have physically dismounted a tape from a unit.	***UNIT DISMOUNT***) CX DISMOUNTED ↵
ELONGATE	Turns LP2/TP2 enlongated printing on or off.	) CX PAUSE @CON25 ↵) CX ELONGATE @CON25 ↵) CX CONT @CON25 ↵

(continues)

Table C-1. EXEC Commands, Alphabetically

EXEC Command	What It Does	Example
ENABLE	Enables a user console for user logon under EXEC; primarily used within UP.CLI macro.	) CX ENABLE @CON45 ⤶
EVEN	Turns even pagination off or on for a printer.	) CX EVEN @LPB1 OFF ⤶
FLUSH	Flushes (terminates) the job that a batch stream or device is processing.	) CX FLUSH 3 ⤶) CX FLUSH @LPB ⤶
FORMS	Allows you to specify a file to be used for special form printing; e.g., bills.	) CX PAUSE @LPB ⤶) CX FORMS BILLS @LPB ⤶) CX CONT @LPB ⤶
HEADERS	Changes the number of header pages printed on a device.	) CX HEADERS @LPB 2 ⤶
HOLD	Holds (suspends) a batch or spool request.	) CX HOLD 4394 ⤶) CX HOLD SACKVILLE ⤶
LIMIT	Enforces user- or operator-defined limits on CPU time or printed pages.	) CX LIMIT 2 1:30 ⤶) CX LIMIT @LPB 25 ⤶
LOGGING	Turns EXEC logging on or off.	) CX LOGGING/START XLOG ⤶
LPP	Changes the maximum number of printed lines per page.	) CX PAUSE @LPB ⤶) CX LPP @LPB 60 ⤶) CX CONT @LPB ⤶
MESSAGE	Writes a text message to EXEC's log file.	) CX MESSAGE 4 STREAMS ⤶
MOUNTED	In response to user mount request, tells EXEC that a tape is physically mounted on a tape unit.	***UNIT MOUNT***) CX MOUNTED @MTA1 ⤶
MOUNTSTATUS	Displays mount request status.	) CX MOUNTSTAT ⤶
OPEN	Opens a device queue to user requests.	) CX OPEN MYQUEUE ⤶
OPERATOR	Tells EXEC that an operator is on or off duty.	) CX OPERATOR ON ⤶) CX OPERATOR OFF ⤶
PAUSE	Pauses a batch stream or device in an orderly way, after the current request is done. Prepares for normal shutdown or change in device specifications. To resume, use CONTINUE.	) CX PAUSE ⤶) CX PAUSE @LPB ⤶
PREMOUNT	Tells EXEC that a person has physically mounted a labeled tape before a MOUNT request for it occurred.	) CX PREMOUNT @MTA2 &⤶ &) VOL1 SAM ⤶
PRIORITY	Sets a new system priority and/or process type for batch streams or cooperative processes (XLPT).	) CX PRIORITY 3 1 ⤶
PROMPTS	Removes or adds time of day from EXEC message display.	) CX PROMPTS OFF ⤶

(continued)

Table C-1. EXEC Commands, Alphabetically

EXEC Command	What It Does	Example
PURGE	Deletes entries in a stopped device queue.	) CX PURGE PLT ↵
QPRIORITY	Sets a new range of batch/print priorities that will be accepted by a batch stream or device.	) CX QPRIORITY 3 1 26 ↵) CX QPRIORITY @LPB 1 99 ↵
REFUSED	Tells the user that you refused a mount request	) CX REFUSED ↵
RESTART	Restarts a printing request.	) CX RESTART @LPB ↵
SILENCE	Silences EXEC; suppresses all batch or device messages to the system console and EXEC log file. Useful on hardcopy system consoles.	) CX SILENCE ↵) CX SILENCE @LPB ↵
SPOOLSTATUS	Displays queue-device association and status.	) CX SPOOLS @LPB ↵
STACK	Tells EXEC to read card images from an input device or file.	) CX STACK @CRA ↵
START	Associates a queue with a device. CX CONTINUE then activates the device.	) CX START LPT @LPB ↵) CX CONT @LPB ↵
STATUS	Gives information about batch streams or devices. The CLI command QDISPLAY/V is also handy for this.	) CX STATUS ↵) CX STAT @LPB ↵
STOP	Stops a device or dissociates a queue. To resume, use CX START.	) CX STOP @LPB ↵
TERMINATE	Terminates the user process associated with a console.	) CX TERMINATE @CON5 ↵
TRAILERS	Changes number of printed trailer pages.	) CX TRAILERS @LPB 1 ↵
UNHOLD	Negates HOLD command.	) CX UNHOLD 4394 ↵) CX UNHOLD SACKVILLE ↵
UNITSTATUS	Describes the mount status of tape units.	) CX UNITSTATUS ↵
UNLIMIT	Negates LIMIT command.	) CX UNLIMIT 2↵) CX UNLIMIT @LPB↵
UNSILENCE	Negates SILENCE command.	) CX UNSILENCE ↵) CX UNSILENCE @LPB ↵
VERBOSE	Makes batch/spool messages verbose.	) CX VERBOSE 4 ↵
XBIAS	Sets a new EXEC small job versus large job bias factor, for a batch stream or device.	) CX XBIAS 1 -100 ↵) CX XBIAS @LPB 100 ↵
XHELP	Describes all EXEC commands or a specified command (omit the leading CX).	) XHELP ↵) XHELP ALIGN ↵

(concluded)

End of Appendix

Glossary

This is the glossary of terms used in the book.

abort — the result of a serious error condition. When a program (like the CLI) hits an error, it may display a warning, error, or abort message. The abort message is the most serious of the three: it means the error was so serious that the program couldn't continue.

access control list (ACL) — a list of privileges, associated with every directory and file, that specifies the type of access allowed for any user. The privileges are O (Owner, can change ACL); W (Write, can modify or delete file); A (Append, can add files to directory); R (Read, can read file or list filenames in directory); and E (Execute, can execute file or use directory name in pathname).

accumulator — a hardware register within the CPU. Accumulators are used for arithmetic, value comparisons, address manipulations, and other things. An ECLIPSE CPU can have eight user-accessible accumulators: four fixed-point 16-bit accumulators, and four floating-point 32-bit accumulators.

ACL — see "access control list."

ADES (Advanced Diagnostic Executive System) — a system of hardware diagnostics for peripherals. It includes tests, exercisers, formatters, and alignment test programs.

ANSI — American National Standards Institute, a committee that publishes standards for a large range of things, including computer languages and tapes, machine screws, and copiers.

AOS — Advanced Operating System, DG's Advanced Operating System for 16-bit ECLIPSE computers.

AOSGEN — a program supplied with AOS that allows you to create and tailor AOS operating systems.

AOS/VS — Advanced Operating System/Virtual Storage, DG's Advanced Operating System for 32-bit ECLIPSE computers.

archive — see "backup."

argument — something that is acted upon by a command, statement, or instruction. For example, in QPRINT MYFILE ↵, MYFILE is an argument to the QPRINT command. In PRINT "Hello", "Hello" is an argument to the PRINT statement.

ASCII — the American Standard Code for Information Interchange. this code establishes standard numeric values for each character used in text. The numbers range from 000 for the null character to 177 (octal) for the DEL character. An international character set extends the ASCII set with numbers from 200 (octal) to 377 (octal); these numbers indicate non-U.S., language-specific characters (for example, the U.K. currency symbol).

ASLM — Asynchronous-Synchronous Line Multiplexor. This board, for microECLIPSE computers, is available with either one or four lines. The line(s) can be asynchronous or synchronous (you specify this at AOSGEN). (ASLM is called "USAM" in DG DESKTOP GENERATION computer documentation.)

asynchronous line — a communications line that uses an asynchronous structure to transmit characters. In such a structure, each character has its own "framing" information: traditionally one start bit (before the character) and one stop bit (after the character). Asynchronous lines are generally used for consoles and for intersystem communication. They consume less system overhead than synchronous lines, but are not as fast.

backup — the procedure of copying disk-based information for safekeeping, done daily on most computer sites. The copy medium is usually magnetic tape, but can be disk or diskettes. AOS aids to backup are the DUMP command, the PCOPY program, and the OPERATOR command.

bad block — on the magnetic surface of a disk, a flawed area that won't hold information. The Disk Formatter notes such areas in a bad block table so the operating system will avoid them; and it creates a remap area of good blocks for the system to use instead of bad ones. If a new bad block develops on the disk, and AOS tries to use the block for I/O, it will detect the bad block and abort the command with a HARD ERROR message.

batch — the technique of processing in a continuous, autonomous stream. Batch jobs don't require a console and they run without human interaction (for example, overnight); they are ideal for big, well-defined tasks, like large sorts. You can tell AOS to run an operation in batch via the QBATCH command.

baud — the rate at which a line (or modem) can transfer data, in bits per second. Normally, on asychronous lines, each character requires 10 bits, so characters are transferred at one tenth the baud rate. The standard (and default) baud rate for consoles is 9600 (960 characters per second). For modems it is 1200.

bitmap — an area on each logical disk that indicates which disk blocks are free and which are used for data storage.

block — see "disk block."

blocked process — a process whose execution is suspended, waiting for an event that may or may not occur. By default, a user process blocks when it creates a (son) process. A process can block voluntarily when it creates a son, via the PROCESS/BLOCK command. Or, another (Super) process can block it, or AOS can block it.

boot, bootstrap — to load a program from a device (tape or disk) into memory and transfer control to it. Bootstrap programs are designed to load other, larger programs like operating systems.

bpi — bits per inch; a measure of data density on magnetic tape. The standard DG tape densities are 800, 1600, and 6250 bpi.

break sequence — a control sequence that involves pressing the CMD and BREAK/ESC keys (or the BRK or BREAK key). Typed on the system console of an MV/8000 computer, if the CPU is unlocked, the break sequence removes the console from AOS control and gives control to the SCP CLI. You can restore control to AOS by typing TTY ↵. Typed on other programmed consoles, the break sequence will display a *!* prompt. Restore control by typing P ↵. Typed on a user console, the break sequence can break binary mode (used by certain programs when they write to the console), or do other things like log the user off.

bus — a connection between hardware components. For example, a CPU has a memory bus and an I/O bus.

byte — 8 bits, can store one ASCII character (e.g., A) or one of 256 different integers.

CEO — DG's Comprehensive Electronic Office System, which includes electronic mail, calendar, filing, and word processing.

checksum — a test used to verify data integrity. One checksum method is to compute the sum of all data bytes and compare it to the known sum. If the values do not match, the data has been corrupted. Checksums are used to verify that data has been loaded correctly (for example, microcode or FIXUP) and as a system sanity check on system constants.

CLI (Command Line Interpreter) — the AOS (and SCP-OS) command languages. AOS CLI commands allow people to communicate with AOS. When AOS is brought up, it automatically runs a CLI process as PID 2 on the system console. This is the *master CLI*, from which all subordinate (son) processes are created. Typically, AOS runs a CLI process for each user who logs on under EXEC. The CLI files are CLI.PR, CLI.ST, and CLI.OL, located in the root directory. On an MV/8000, the SCP CLI runs under the SCP operating system, in the microprocessor CPU (if any). The SCP CLI commands control the main processor and boot the AOS operating system.

CMD key — a key on the terminal keyboard that, in conjunction with other keys (like BREAK/ESC), can do things like produce the break sequence.

cold start — start up when power to the CPU has been off.

console — an interactive device with a keyboard for input and a screen or printer for output. The system console connects directly to the SCP and main CPU and can issue commands to them. User consoles interact with the computer through a multiplexor. The filename of a user console is @CONn; the person or program using any console can address it by the name @CONSOLE.

control point directory — see "directory."

control store — the part of a computer where microcode is stored.

converting nonDG tapes for AOS — see "magnetic tape."

co-operative process — a process created by a system utility to communicate with it, and other processes. EXEC, for example, creates and uses a co-operative process named XLPT to manage printing devices for it.

crash — see "panic."

cursor — on a CRT console screen, the cursor indicates the current position on a line. It is either a box superimposed on a character position or an underscore beneath a character position.

DCU (Data Control Unit) — a device used to manage synchronous communications lines. Also see "ISC."

database — a central location (one or more disk files) for data, which can be shared by more than one user or program. DG has two systems for creating and managing databases: INFOS II and DG/DBMS.

deadlock — a situation in which the AOS scheduler cannot react to all the demands on it; the system does not respond, and seems frozen.

dedicated line — a phone line installed and used exclusively for communication between computer systems. Dedicated lines can remain open and available though day and night. Communication over a dedicated line is more precise, faster, and more expensive than communication over a switched line.

DEDIT — a disk file editor utility supplied with AOS.

default, by default — a value or parameter that a program uses if you do nothing about it. Two examples — the PROFILE macro has the default answer of :CLI.PR for each user's initial program; and the SED text editor displays line numbers by default.

diagnostic program — a program run to determine the source of a hardware failure (or incipient failure). DG's ADES system is a series of diagnostic programs.

DIB (Disk Information Block) — a block of information written to each physical disk in an LDU by the AOS Disk Formatter during a Full format. The information includes the disk type, LDU unique ID, and the types and sequence of all disks in the LDU. AOS and all its programs require a valid DIB before they can access the disk as an LDU.

DIP (dual in-line package) switch — a very small switch, often in groups of eight, that enables or disables a hardware function; for example, 8-bit operation on a console. An alternative to DIP switches is hardware (wire) jumpers, but DIP switches are far more convenient. You can change the setting of DIP switch with a pencil point.

directory — a file that contains (points to) other files. AOS uses directories extensively. A *standard directory* has no set size; it grows according to the files in it. A *control point directory* is a directory created with a maximum space size (in 512-byte blocks). You can create a noncontrol point directory with the CLI command CREATE/DIR dirname ↵; you can create a control point directory with the command CREATE/DIR/MAX=size dirname ↵. The PREDITOR utility creates each user's initial directory as a control point directory. On the master LDU, the root (:) is the master control point directory. You can tell the amount of space remaining in a control point directory via the command SPACE cpd-pathname ↵.

disk block — an area on a disk that includes 512 bytes of storage.

diskette — a medium used for software distribution and file backup. The standard diskette used with ECLIPSE systems holds 737,000 bytes. DESKTOP GENERATION systems use a 5-1/4 inch diskette that holds 368-Kbytes.

Disk Formatter — a program supplied with AOS, in both stand-alone and stand-among versions, used to create LDUs, change LDU specifications, and check disk surfaces for flaws (bad blocks).

dump — to copy information onto another medium. Most dumps are done for backup. Sometimes a dump is done to provide diagnostic information, as in a memory dump (which copies computer memory so programmers can examine it, try to reproduce the problem, and solve it).

element, file — see "file element."

emulator — a program that defines computer instructions that aren't defined in the hardware. Functionally, it resembles microcode. A DESKTOP GENERATION Model 10/SP system that uses non-English language character sets must have an emulator installed and running in its IOC board before it can be used.

EPROM — erasable programmable read-only memory. Also see "RAM."

ESD — a part of AOS that does an Emergency Shutdown, closing all open files and allowing immediate restart. ESD is most useful after a panic *(FATAL AOS ERROR)*. You can run it after a panic as described in Chapter 6. (ESD is also an acronym for Electro-Static Discharge — static electricity — and its effect on computers.)

EXEC — a utility supplied with AOS that manages user consoles, batch and spool queues, and user mount requests. EXEC accepts commands from any process with username OP and sends messages to the system console. EXEC files are EXEC.PR, EXEC.ST, and EXEC.OL, usually in directory :UTIL; but the EXEC process's initial working directory, when EXEC is run, must be directory :PER (shorthand @).

FCU (Forms Control Utility) — a utility program that inserts control characters in files for special-form printing.

field engineer — a DG engineer whose primary responsibility is hardware.

file — a collection of information stored as a unit, under a filename. Some device filenames are rigidly defined (e.g., @MTB0 for tape, @LPT for the line printer queue, @CONSOLE for the console); but disk filenames and LDU filenames are flexible (also see "names").

file element — a set of contiguous 512-byte disk blocks that AOS allots to each file as the file grows. The default file element size is one block, but users can specify larger element sizes via the CREATE/ELEMENTSIZE= switch when they create files.

fileset — a name for one or more disk files written (usually dumped) to a set of one or more labeled magnetic tape/diskette volumes. For example, if a user writes one or more volumes to a three-volume set in a single operation, the fileset includes the three volumes. The fileset ID is written to the labels and allows the system to keep track of the volumes that contain the fileset.

firmware — instructions that control some operations of a computer-based device; similar to microcode.

FIXUP — an AOS utility program that restores file integrity and streamlines the file structure on an LDU. You must run FIXUP after an abnormal shutdown in which ESD failed, but you can run it anytime to clean up the disk. You should run it when you suspect system data has been corrupted (for example, if the system panics whenever someone accesses a specific directory).

floating-point unit (FPU) — one or more circuit boards that speed up computations with floating-point numbers.

formatting (disks) — an operation that prepares the magnetic surface of a disk or diskette for data storage. *Hardware formatting* divides the disk surface into sectors (disk blocks); then checks the disk surface for flaws. All DG disks and diskettes are hardware formatted before they are shipped. *Software formatting* creates tables that an operating system needs on a disk or diskette, including a block table that allows the system to find good substitutes for bad blocks (including those found during hardware formatting). These acts create a Logical Disk Unit (LDU) from one (or more than one) physical disk. Software formatting is not needed for diskettes that you don't want to use for directories; for example, it isn't needed for diskettes you'll use for labeled tape backup. Hardware formatting is done by the MV/ADES system; software formatting is done by the AOS Disk Formatter (DFMTR).

fragmentation — Storage in fragments. For fast access on disk, the elements of a file should be stored sequentially on the disk surface. AOS does this when there are no other file elements in the way (when an LDU is relatively empty). But after an LDU begins to fill up, fragmentation begins: AOS must seek farther for space for new file elements. For example, one element is near an outer cylinder, and the next available space is near an inner cylinder. To write the next element, the system must move the read/write head all the way across the disk. File fragmentation often occurs when an LDU is nearly full. It can significantly slow access to the disk involved.

FRU (Field Replaceable Unit) — a circuit board that can be replaced on site; or a diagnostic test that checks one or more of these circuit boards.

FTA — the XODIAC network file transfer agent. It helps copy files from one computer system to another.

function key — one of the keys in the topmost row of a console keyboard. Each key, alone or in conjunction with the SHIFT and/or CTRL keys, can represent a command. (Pressing a key is easier than typing a command.) A product's function keys (if it has any) are identified by a shaped *template* that fits over them.

generic file — a category of files, some of whose names you can set with CLI commands. For example, the generic LISTFILE can be set to @CONSOLE or a disk file; and generic file @NULL is a useful destination file when you don't care about reading the data written.

Ghost — the heart of the user interface in AOS. In conjunction with the operating system, peripheral manager (PMGR) and the CLI, the Ghost preprocesses most system calls, validating input when it can, and acting on behalf of each user to communicate with AOS. Files are GHOST.PR, GHOST.OL, and GHOST.ST in the root directory.

GIS — graphics instruction set, a hardware option available with Data General DS/4000-series computers. It works in conjunction with DG's GKS (Graphics Kernal System) to produce graphics and enable "mouse" operations.

hang — see "deadlock."

hardware diagnostic — a test program designed to identify current (and potential) problems with peripheral hardware, like disks. DG's ADES (Advanced Diagnostic Executive System) includes series of such tests (as does DG's DTOS, another diagnostic system).

hash frame size — a value associated with a directory, used to index into the data structure that contains the directory filenames. The default value is 7. This is appropriate for a medium-size directory. The proper hash frame size for any directory can reduce the number of disk accesses needed to find a filename in the directory.

heuristic treatment — AOS studies swappable processes and assigns them priority based how much time passes between blocking events. AOS raises the internal priority of processes with many blocking events and lowers the internal priority of processes with few blocking events. This is called heuristic treatment (also see "timeslice").

hierarchy (process) — all processes are related in a structure that resembles an inverted tree. The highest processes are the peripheral manager process (PMGR, PID 1) and the master CLI process (PID 2).

host — a computer system that's connected to one or more other systems. The system you are on is called the *local* host; any other system is called a *remote* host.

IAC (Intelligent Asynchronous Controller) — a muliplexing device that handles user terminals.

IACRS — see "peripheral manager."

IBM tapes — see "labeled tape" or "magnetic tape."

INFOS II — a file management system, available with AOS, that lets users create, maintain, and use large databases, via COBOL, FORTRAN 77, PL/I, RPG II, and other application programs. INFOS II is a superset of an ISAM file system, with an ISAM extension called DBAM (database access method). It has an interactive graphics and report-generator query program named PRESENT.

initialize — a general-purpose computer term meaning "introduce" or "open" in the context of hardware and software. For example, to initialize AOS means to give it vital information (like date and time) at startup. To initialize a logical disk unit (LDU) means to open it to the AOS file system.

Installer — an AOS utility program that places a disk bootstrap, system bootstrap (SYSBOOT), and/or AOS system on a system LDU.

ISC (Intelligent Synchronous Controller) — a device that handles synchronous communications lines (often used for communication with IBM systems). An alternative to a DCU.

LABEL utility — creates ANSI or IBM labeled tapes or or diskettes; also see "labeled tape/diskette" and "magnetic tape."

labeled tape or *labeled diskette* — a magnetic tape or diskette with labels. Each label is is information — including a volume ID (volid) — that describes the contents of the tape or diskette. For tape, this information is written by the LABEL utility (when you execute LABEL) and by the operating system (when you or a user program issue a command that writes to the tape). You can label and write a tape in either DG, ANSI, or IBM format. For diskette, label information is written — usually — by the CLI, and by the operating system (when you or a user program issue a command that writes to the diskette). You can read the label applied by the LABEL program (which includes the volid) with the command X DISPLAY @unitname ⤶. Labeled tape or diskette has many advantages over the unlabeled variety; we recommend it for all your backups.

laser document printer — a printer that produces high-quality copy that resembles typeset copy.

LDU — logical disk unit: one or more physical disks, processed by the Disk Formatter into one logical disk.

letter-quality printer — a printer that produces copy suitable for a business letter. The copy resembles typewritten copy.

line (communications) — see "asynchronous line" or "synchronous line. "

line (of text) — a sequence of ASCII characters that ends with either a NEW LINE, form feed, or null character.

link entry — a file whose sole function is to indicate another file's pathname, created with the CREATE/LINK command. Thus, a link named MAR to :UDD:CHRIS:MARCH_REPORT makes access to MARCH_REPORT easy; for example, you can type just TYPE MAR ⤶.

local area network — a network of computer systems that are relatively close to one another — up to a mile apart. This is a good arrangement for one or more MV/Family systems (as hubs of the network), and DESKTOP GENERATION systems as remote hosts.

local (item) — an "item" (like system, console, or printer) that is managed by your computer system without a communications line. The opposite of local is *remote*. For example, in a XODIAC network, from your local console, you can log on to a remote system and use the remote printer. Or, when you use a modem line, your local console communicates with the remote system.

logical address — the address that a user or user program sees. For CPU addresses, the main processor translates each logical address to a physical address to access main memory. For disk addresses, AOS does the translation.

logical disk unit (LDU) — one or more physical disks, processed by the Disk Formatter into one "logical unit."

magnetic tape — a medium used for software distribution and file backup. Types of tape unit are MTAn (800 bpi), MTBn (1600 or 800 bpi), and MTCn (1600 or 6250 bpi, reel or cartridge). Unlabeled tape files are numbered 0, 1, 2, 3, ... and so on. Users append the number to the device name to specify the tape file; e.g., @MTB0:0. Tapes can be labeled with the LABEL utility. Users specify labeled tape filenames via their own linkname and tape filename; e.g.,TAPE:FILE. All AOS software is shipped on 800- and 1600-bpi magnetic tape; the SYSTAPE.CLI macro produces your own tailored system tape in the same format as the supplied starter tape. Tapes written on an IBM or other non-DG system can be converted and loaded into an AOS system with the AOS DISPLAY utility or the Sort/Merge program.

master CLI — see "CLI."

master LDU — the logical disk unit that holds the currently running AOS system.

MCA (Multiprocessor Communications Adapter) — a device that allows network communications between DG systems that are hardwired (not connected via phone line) to one another.

MCP1 — Multicommunications Processor, one PC board that includes an asychronous line controller that supports eight lines, a synchronous line controller that supports two lines, and a data channel line printer controller. You identify the async lines, sync lines, and printer controller to AOSGEN as if they were separate devices.

microcode — control sequences that implement the instruction set of a computer. Microcode consists of a series of microinstructions.

modem — a device that connects an asynchronous line from a computer to a telephone line, and connects the other end of a telephone line to a console. Two modems are needed for each active remote console.

multiplexor (mux) — a general term for a device that sorts and controls multiple signals. In this book, a multiplexor is part of a user-console handling device called an ALM, IAC, IOP, ASLM, or ULM. On each device (except IOP), the mux is an integral part of the device.

mux — see "multiplexor."

names — AOS filenames can be from 1 through 31 characters, including letters (A,a...), numbers (1,2...), underscore (_), period (.), dollar sign ($), and question mark (?). The LDU names (assigned with the Disk Formatter) can be 1 through 15 of the above characters. Labeled tape volid names can be 1 though 6 characters. Usernames (assigned via PREDITOR) can be 1 through 15 characters; and passwords can be 3 though 15 characters. AOS devicenames are rigidly defined; they begin with three letters (e.g., MTB, DPF, LPB), sometimes followed by a one- or two-digit number to indicate the controller and unit number; e.g., MTB0, DPF0.

NBA (Network Bus Adapter) — a device used by DG's XODIAC networking system for a network of computers, within 1 mile of one another.

overlay — a section of executable program code that can be called into a reserved area of memory as needed, and then overwritten by another overlay called into memory. Overlays associated with a program usually reside in an overlay file that has the suffix (or extension) .OL.

page — in memory, a 2,048-byte quantity.

page-milliseconds — indicates process memory usage in relation to time; formed by multiplying the number of memory pages used by the number of CPU milliseconds used. The PED and REPORT utilities both give page-millisecond figures for processes.

panic — an error that halts AOS processing with a fatal error message on the system console.

patch — a correction or update made to a program on disk. DG provides patch files in updates for AOS programs like EXEC, AOS systems, and the CLI. The patch filename describes the program. You can apply the patches easily with the PATCH utility, described in Chapter 4.

pathname — a path, usually including one or more directory names, to a file. For example, :UDD:JACK:MYDIR:MYFILE is a pathname.

PCOPY — a fast stand-alone copy program supplied with AOS, used to back up an LDU by making a physical copy of it, onto another LDU, magnetic tape, or diskettes.

PED (Process Environment Display) — an AOS utility program that can display all vital statistics of a process.

peripherals directory (PER) — the AOS directory that holds all device entries. Its full pathname is :PER (or shorthand prefix @). The prefix @ that you use with devicenames specifies the peripherals directory.

peripheral manager — a set of programs supplied with AOS that supervises all character I/O (for example, with user consoles). The base program always runs as PID 1. The files are all in the root directory. They are PMGR.PR, PMGR.ST, IACRS.PR, IACRS.ST, IOPMGR.PR, IOPMGR.ST, ALDCU.PR, and ALDCU.ST.

physical address — The address that the hardware sees. For a CPU, it points to a specific address in main memory. For a disk, it is an address on the physical disk (as opposed to a logical address on the LDU).

PID — the Process ID that the system creates and associates with each process.

PMGR — see "peripheral manager."

PREDITOR — the user profile editor supplied with AOS, which creates user profiles that identify system users to EXEC, and allows them to log on and off.

process — a program set up for execution; or more specifically, the memory, CPU, disk, and other system resources used by an executing program. AOS runs each program as a process. Also see "blocked process."

queue — a file that stores print and batch requests until the printer and system are ready to process them. The system runtime queue directory and file are :QUEUE:Q. The default line printer queue is @LPT.

RAM — random access memory. Most of a computer's memory is RAM memory. Programs can read from *and write to* RAM.

record — a series of one or more characters written to or read from a file. Records can be read and written by a text editor (they are lines of text like these). Or, records can be read and written by your own applications programs.

Release Notice — notice of recent software changes that DG hasn't yet been able to include in pertinent manuals, supplied with AOS and other software as a printed listing and a disk file in :UTIL (pathname :UTIL:RELEASE.n.nn; e.g., :UTIL:RELEASE.5.00).

remote (item) — an "item" (like a system, console, or printer) managed by another computer or by your computer over a communications line. The opposite of remote is *local.* For example, a remote console is one attached to your system via a modem. But when you're using the modem, it is local and the system is remote.

revision — a new version of AOS (or other software) and manuals. DG issues a new revision of AOS about every 6 months, sending it to customers on DG's Software Subscription Service.

ROM — read-only memory. Another form of ROM, called PROM, is programmable; and still another, called EPROM, is erasable and programmable. The part of the SCP that does the power up checking is EPROM. All forms of ROM are nonvolatile: they retain their values when power is shut off.

root directory (:) — the system master directory that contains and gives access to all other directories.

SCP (System Control Processor) — a microcomputer that provides a user interface (called the "programmed console") to the main processor on MV/8000s. The SCP has its own operating system (SCP-OS) and CLI (SCP-CLI). The SCP program loads from disk or tape, can detect and log hardware errors, and run CPU diagnostics.

scratchpad memory — a high-speed, RAM memory outside (but accessible to) the main CPU. It contains constants and temporary storage locations.

SDCU — the combination of a Data Control Unit (DCU) with one or more synchronous line multiplexors; used to manage synchronous communication lines.

search list — a list of directories that AOS searches when it cannot find a file in the working (current) directory; set with the CLI command SEARCHLIST. User searchlists are often set in the user LOGON.CLI macro; the master CLI, PID 2, searchlist is usually set in the UP.CLI macro. If you get a *FILE DOES NOT EXIST* error and you know that the file exists, check your searchlist.

sector — see "disk block."

Software Subscription Service — a service that provides new revisions of AOS and support software as DG creates them. A year's membership is included with purchase of AOS; membership is available thereafter.

Software Trouble Report (STR) — a formal report, made by a customer to DG through a DG systems engineer, about a serious problem that the customer is having with the software. The cause may be a user or DG error. DG personnel try to duplicate the problem to solve it, and need as much information about the problem as possible. Or, instead of reporting errors, an STR can simply offer suggestions.

stand-alone program — a program that runs by itself, without an operating system to manage its I/O, supervise scheduling, etc. Each AOS operating system is itself a stand-alone program. There are also stand-alone versions of the Disk Formatter, Installer, FIXUP, and PCOPY.

stand-among program — a version of stand-alone program reconfigured to run under AOS. There are stand-among versions of the Disk Formatter, Installer, FIXUP, and PCOPY. The advantage of the stand-among programs is that you need not shut down AOS to run them (but they cannot be run on any LDU that is part of the system LDU's file structure).

STR — see "Software Trouble Report."

subslice — the time interval between Scheduler runs. A hardware timer, called a PIT (Programmable Interval Timer), generates an interrupt every 32 milliseconds, telling the Scheduler to run. Nearly always, a different process gets CPU control after each subslice expires.

Superprocess — a privilege that allows a user process to control any process on the system. You can turn on Superprocess mode (if you have the privilege) via the command SUPERPROCESS ON).

Superuser — a privilege that allows a user to bypass file access controls and access any file on the system. You can turn on Superuser mode (if you have the privilege) via the command SUPERUSER ON.)

support organization — the DG group or person with whom you have a contract for help and support; inside the U.S. this is usually the Atlanta Support Center. Outside the U.S., it is often your local systems engineer.

swapping — action that AOS takes to resolve competition for memory. When a set of processes is too large to fit into memory at once, AOS makes room as follows: it fits some processes into memory, stores the others in the swap file (SWAP.SWAP), and rotates processes in and out of memory.

SWAT — DG's high-level language debugger, that works with AOS COBOL, FORTRAN 77, and PL/I.

switch — a slash (/) followed by a value. Switches change the meaning of a command or action performed on its arguments; they are a major feature of AOS and AOS/VS.

switched line — a normal telephone line, which makes connections via normal telephone switching stations. It is less expensive (and slower) than a dedicated telephone line.

symbol table file — a file, built by the Link utility when it creates a program file, and which identifies all symbols used in the program. These symbol files are needed by the PATCH utility, assembly language debugger, and disk file editor (DEDIT). The files are not needed for normal program execution. The standard symbol table files have the suffix (or extension) .ST; the SWAT debugger symbol table files have the suffixes .DS and .DL.

synchronous line — a communications line that uses a synchronous protocol to transmit or receive data. Synchronous lines are frequently used for high-speed and/or long-distance communication between computer systems. They are faster than asynchronous lines, but require more system overhead.

system console — the console connected directly to the CPU. User consoles are not connected *directly*, but through a multiplexor that sorts incoming incoming and outgoing data flow.

system engineer — a DG engineer whose primary expertise is software, secondarily hardware. *Field engineers* generally handle hardware.

system tape (AOS) — see "magnetic tape."

template — there are two meanings. First, a template is shorthand for part of a filename, used with one of the template characters (+, −, *, \, #), to access one or more files. For example, the template FOO+ matches all filenames that begin with the characters FOO in a specified directory. Second, a template is a cardboard or plastic shape that fits over or above the topmost group of keys on the keyboard (function keys), labeling them.

timeslice — A unit of time that includes from 1 to 64 subslices. It is the amount of time AOS expects a swappable process to last without blocking. If a process lasts without blocking for more than its timeslice, AOS reduces its internal priority, thus penalizing it in relation to other swappable processes. If a process blocks before its timeslice expires, AOS raises its internal priority, giving it CPU preference over other swappable processes. Therefore, the scheduling arrangement for swappable processes favors highly interactive processes (which have many blocking events as people pause for thought). This arrangement is not ideal for noninteractive (batch type) processes, which have *few* blocking events. The latter processes will run better as preemptible processes. You can have the PED program display a process's timeslice *exponent* via argument ?PSSL (subslice power of 2; e.g., 3 would mean that AOS expected the process to last for eight subslices without blocking).

thrashing — a condition in which the Scheduler has the system spending most of its time reading or writing memory pages. The system is doing little or no useful processing.

user directory — the directory created and maintained for each interactive user. Usually becomes the working directory when the user logs on. It can have subordinate directories.

user directory directory (:UDD) — the AOS directory that contains each user directory.

user profile — a disk file, created via the PREDITOR utility, that contains each user's username, password, disk space allowance, and other privilege specifications.

username — the name under which an AOS user logs on; specified to the PREDITOR utility. The username is also the name of the user directory.

user, system — anyone who (in any capacity) has logged onto an AOS system. This can be programmer, manager/operator seeking information, or nontechnical person.

UTIL — the utilities directory, contains most (if not all) utility programs on the system. UTIL's pathname, :UTIL, is often found in search lists.

utility, utility program — a program supplied by DG to help you generate systems or build programs; for example, the CLI, Disk Formatter, AOSGEN, PED, and Link.

virtual console — a device entry on an AOS system that allows remote AOS users to log on as if they were on a local console; provided by DG's XODIAC Network VTA (Virtual Transfer Agent). VTA's virtual consoles are unrelated to real consoles; for example, an AOS system running 40 real consoles for 40 local users can also run virtual consoles for remote users.

volatile memory — memory whose contents are lost when power goes down. Semiconductor RAM memory (which holds the SCP, on MV/8000s, and AOS operating systems) is volatile. ROM memory is nonvolatile.

volid — the six-character volume ID that identifies each labeled tape, written with the LABEL utility.

warm start — startup in which power has remained on to the CPU.

working directory — the directory where you are; the current directory.

X.25 — the XODIAC network management support process, which runs all other network operations. X.25 is the name of an international standard for intercomputer communications.

XLPT — name of EXEC's co-operative process to manage printing devices. EXEC's co-op for plotters is named XPLT.

XODIAC — DG's networking system, which allows a large DG system to communicate with one or more DG systems in a private or local area network. XODIAC also allows DG systems to participate in a Public Data Network like TELENET.

End of Glossary

Index

Within this index, "f" or "ff" after a page number means "and the following page" (or "pages"). Commands, calls, and acronyms are in uppercase letters (e.g., CREATE); all others are lowercase.

A

C

M

N

O

P

Q

User Documentation Remarks Form

Your Name ______________________ Your Title ______________________

Company ______________________

Street ______________________

City ______________________ State __________ Zip __________

We wrote this book for you, and we made certain assumptions about who you are and how you would use it. Your comments will help us correct our assumptions and improve the manual. Please take a few minutes to respond. Thank you.

Manual Title ______________________ Manual No. __________

Who are you? ☐EDP Manager ☐Analyst/Programmer ☐Other __________
☐Senior Systems Analyst ☐Operator __________

What programming language(s) do you use? ______________________

How do you use this manual? *(List in order: 1 = Primary Use)* ______________________

___ Introduction to the product ___ Tutorial Text ___ Other
___ Reference ___ Operating Guide __________

About the manual:

	Yes	Somewhat	No
Is it easy to read?	☐	☐	☐
Is it easy to understand?	☐	☐	☐
Are the topics logically organized?	☐	☐	☐
Is the technical information accurate?	☐	☐	☐
Can you easily find what you want?	☐	☐	☐
Does it tell you everything you need to know	☐	☐	☐
Do the illustrations help you?	☐	☐	☐

If you have any comments on the software itself, please contact Data General Systems Engineering.
If you wish to order manuals, use the enclosed TIPS Order Form (USA only).

Remarks:

Date

134-664

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 26 SOUTHBORO, MA. 01772

POSTAGE WILL BE PAID BY ADDRESSEE

DataGeneral

User Documentation, M.S. E-111
4400 Computer Drive
Westborough, Massachusetts 01581

Notes

Notes

Notes

Notes

Notes

Notes

$1\frac{1}{8}$ = ESD

ALM — ASYNCHRONOUS LINE MULTIPLEXER
ASLM — ASYNC/SYNC LINE MUX
BBU — BATTERY BACK UP
BMC — BURST MUX CHANNEL
CON — CONSOLE
CRA — CARD READER
DK
DP } — DISKS
IAC — INTELLIGENT ASYNC CONTROLLER
IAP — INTEGRAL ARRAY PROCESSOR
LP — LINE PRINTER
MCA — MULTIPROCESSOR COMMUNICATIONS ADAPTOR
MT — MAG TAPE
PLA — PLOTTER
SDCU — SYNCH DATA CONTROL UNIT
SLM — SYNCHRONOUS LINE MUX
TPA — PAPER TAPE PUNCH
TRA — PAPER TAPE READER
ULM — UNIVERSAL LINE MUX

Data General Corporation, Westboro, MA 01580

093-000217-06