Data General

DESKTOP GENERATION™

Using AOS on DESKTOP GENERATION™ Systems

Notice

DGC reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult DGC to determine whether any such changes have been made.

The terms and conditions governing the sale of DGC hardware products and the licensing of DGC software consist solely of those set forth in the written contracts between DGC and its customers. No representation or other affirmation of fact contained in this document including but not limited to statements regarding capacity, response-time performance, suitability for use or performance of products described herein shall be deemed to be a warranty by DGC for any purpose, or give rise to any liability of DGC whatsoever.

In no event shall DGC be liable for any incidental, indirect, special or consequential damages whatsoever (including but not limited to lost profits) arising out of or related to this document or the information contained in it, even if DGC has been advised, knew or should have known of the possibility of such damages.

Data General Corporation has acquired rights in the product it calls COMPUCALC from Concentric Data Systems, Inc., Westboro, MA. Within the terms of that agreement, Data General may have used and adapted material belonging to Concentric Data Systems, Inc., in creating portions of this manual. Concentric Data Systems, Inc., is an independent company and is neither an agent, representative, partner, joint venturer, nor subsidiary of Data General Corporation.

Using AOS on DESKTOP GENERATION^{TM} Systems
Ordering No. 069-000058-01

Revision History:	Effective with:
Original Release - October 1983	
First Revision - February 1984	
Addendum 086-000071 - July 1984	AOS Rev. 6.0

About this Book

AOS is Data General's (DG's) Advanced Operating System for ECLIPSE® computers — including the DESKTOP GENERATION™ Model 10/SP, 20, and 30 computers.

In an hour or two, you can install AOS with support software. And in another hour, you can develop a working sense of AOS by using it. This book shows how — with enough information to get you started. Then, it lists other DG manuals that give more information.

Before you can start or run AOS (as shown in this book), you must have

- unpacked your DESKTOP GENERATION hardware;
- connected the components (computer, hard disk, keyboard and display screen, printer, and user terminals, if any);
- plugged system components into appropriate ac outlets; and
- run a hardware test.

For Model 10/SP systems, these installation tasks and hardware operations are described in

Installing Model 10 and Model 10/SP Systems, part number 014-000901

Testing Model 10 and Model 10/SP Systems, 014-000902

Operating Model 10 and Model 10/SP Systems, 014-000900 (accompanies this manual for Model 10/SP systems).

For Model 20 and 30 systems, installation and hardware operations are covered in

Installing Model 20 and 30 Systems, part number 014-000904

Testing Model 20 and 30 Systems, 014-000905

Operating Model 20 and 30 Systems, 014-000903 (accompanies this manual for Model 20 and 30 systems).

What Do You Want to Do?

This book assumes that you want to learn about using (or installing) AOS or related software on your desktop computer. To use the CP/M®-86* or MS™-DOS** operating system, see the appropriate book, listed later in the Preface.

To install AOS software, read Chapter 1 (for background) and Chapter 2. Or, to update an existing AOS system or add hardware to it, see Chapter 8.

To write text — memos, letters, reports, books, or other material — you can use DG's CEO® Electronic Office or CEO Word Processor-Independent. See Chapter 6.

For general AOS operations, like creating files and directories, moving files, printing them, and backing them up for safekeeping, you'll need the command language (CLI), and other AOS-related programs. Read Chapter 1 for background. Then, for the CLI, see Chapters 4 and 5. To back up files, see Chapter 7.

To write computer programs, you may need Chapter 1 for background. Then, for BASIC, read Chapter 10. For a language other than BASIC, you need a text editor (Chapter 9), and Chapter 10. To understand the examples, you need some experience with the language involved.

To hardware or software format your diskettes, or to copy one diskette to another, see Chapter 14.

To recover from an error condition, or whenever you don't know what to do, see Chapter 15.

To find the meaning of a computer-oriented word you don't know, see the Glossary.

* *CP/M®-86 is a registered trademark of Digital Research.*

***MS™-DOS is a trademark of Micro Soft.*

How is this Book Organized?

Chapter 1 introduces the AOS software; describes the programs available with it; and offers some cautions and hints.

Chapter 2 tells how to install AOS on a blank hard disk, then install other software on it (like the CEO Electronic Office or XODIAC™ networking system). Read it when you want to install your first AOS system or other software products on a blank hard disk.

Chapter 3 explains the steps you take to start up and shut down AOS on your DESKTOP GENERATION system.

Chapter 4 shows how to use the AOS command language (CLI) in a hands-on session.

Chapter 5 explains the file system, then describes common CLI commands, and the macros and programs shipped with AOS.

Chapter 6 shows how to start the CEO Electronic Office or CEO Word Processor-Independent and how to exit from CEO.

Chapter 7 shows you how to back up your system's files for safekeeping and how to restore these files.

Chapter 8 explains how to handle hardware upgrades and software updates (new hardware and software acquired from DG).

Chapter 9 shows how to run the text editor named SED and describes other programs in the optional General Language Development Package.

Chapter 10 shows how to build computer programs in the following five computer languages: BASIC, FORTRAN, COBOL, Pascal, and PL/I.

Chapter 11 explains record management with the Sort/Merge program and INFOS® II file management system. This chapter may interest you if you have some programming experience.

Chapter 12 introduces the optional programs — like XODIAC networking — that enable your system to communicate with another computer system.

Chapter 13 outlines DG's optional graphics hardware and software available with AOS for DESKTOP GENERATION systems.

Chapter 14 describes how to prepare your diskettes for use, and how to copy one diskette to another.

Chapter 15 describes error conditions, error messages, and how to recover from errors. It also gives the DG phone numbers to dial for help.

Glossary defines pertinent terms, like *batch* and *byte*. When you see a term you don't know, check the glossary.

Fast-Reference Summary Card summarizes AOS commands, macros, and programs; and startup and shutdown steps. It is before the back cover.

For quick reference, insert and use the *tabbed dividers* packed with this book.

Reader Please Note:

In this book, *system console* means specifically the operator's terminal — the one from which you bring up AOS.

We use these conventions for command formats:

```
COMMAND required [optional]...
```

Where **Means**

`COMMAND` You must enter the command (or an acceptable abbreviation) as shown.

`required` You must enter some argument (such as a filename). Sometimes, we use:

$$\left\{\begin{matrix}\texttt{required}_1\\ \texttt{required}_2\end{matrix}\right\}$$

which means you must enter one of the arguments. Don't type the braces; they only set off the choice.

[optional] You have the option of entering this argument. Don't type 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 key on your terminal's keyboard.
□	Be sure to insert a space here. (We use this only where we must; normally, you can see where to put spaces.)

All numbers are decimal.

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.

If you have comments on this manual, please use the prepaid Remarks Form that appears after the Index.

Related Documentation

After you read this book, you may want to look at other Data General manuals. AOS, CEO, communications, COMPUCALC, CP/M-86, graphics, MS-DOS, programming languages, and record management manuals are listed below.

AOS Manuals

AOS Debugger and Disk File Editor User's Manual (093-000195-02)
AOS Link and File User's Manual (093-000254-01)
AOS Macroassembler Reference Manual (093-000192-03)
AOS Programmer's Manual (093-000120-04)
AOS and AOS/VS SPEED Text Editor User's Manual (093-000197-03)
Command Line Interpreter (CLI) User's Manual (093-000122) (included in PREGEN AOS)
How to Generate and Run Your AOS System (093-000217) (for SYSGEN AOS)
Learning to Use Your AOS System (069-000018) (for SYSGEN AOS)
SED User's Manual (093-000249) (for SYSGEN AOS and GLDP)

BASIC

BusiGEN™ User Guide (069-705011)
Business BASIC Commands, Statements, and Functions (093-705005)
Business BASIC Subroutines, Utilities, and BASIC CLI (093-705006)
Business BASIC System Management (093-705007)
Business BASIC Technical Concepts (093-705004)
Guide to Using Business BASIC (069-000028)
MP/BASIC Reference (093-400005) (included with PREGEN AOS)
Extended BASIC User's Manual (093-000065)

CEO Manuals

Getting Started with CEO® (069-000036)
Managing CEO® Word Processing - Independent (093-000271)
Managing Your CEO® System (093-000286)
Using CEO® Word Processing (093-000285)
Using CEO® Word Processing - Independent (093-000220)

COBOL

COBOL Reference Manual (AOS COBOL) (093-000223)
Interactive COBOL Programmer's Reference Manual (069-705013)
AOS Interactive COBOL User Guide (069-70515)
AOS Interactive COBOL Utilities (069-705021)
IC/EDIT Interactive COBOL Editor (055-004)

Communications

Generating, Running, and Using DG/XAP™ (093-000352)
Generating, Running, and Using DG/GATE™ (093-000353)
XODIAC™ Guide for Operators and Network Managers (093-000260)
XODIAC™ Network Management System User's Manual (093-000178)
X.25 Protocol User's Manual (AOS and AOS/VS) (093-000175)
How to Use DG/BLAST (069-100006)
HASP Workstation Emulator User's Manual (AOS and AOS/VS) (093-000158)
RCX70 Terminal Operator's Guide (093-000170)
RCX70 Reference Manual (AOS) (093-000172)
RJE80 Control Programmer User's Manual (AOS and AOS/VS) (093-000157)
SNA/RJE Operator's and User's Guide (093-000301)

DG/SNA Programmer's Reference Manual (093-000282)
DG/SNA Operator's Guide (093-000283)
SNA/APILU2 User's Manual (093-000302)
SNA/3270 Operator's Guide (AOS and AOS/VS) (093-000287)
DG/SNA DG/3278 User's Terminal Guide (093-000284)
SNA/3278/APL Operator's Guide (093-701004)

COMPUCALC™

The COMPUCALC™ Electronic Spreadsheet User's Guide (069-705031)

CP/M®-86

Using CP/M®-86 on DESKTOP GENERATION™ Systems (069-100007)

FORTRAN

FORTRAN 77 User's Manual (093-000162)
AOS FORTRAN 77 (F77) Environment Manual (093-000273)
FORTRAN IV User's Manual (093-000053)
FORTRAN 5 Reference Manual (093-000085)
FORTRAN 5 Programmer's Guide (AOS) (093-000227)
FORTRAN Commercial Subroutine Package (093-000107)
FORTRAN QCALLS Reference Manual (093-000239)

Graphics

Graphical Kernel System Reference Manual (093-000355)
Model 10/SP Monitor and Keyboard User's Manual (014-000770
PRESENT Information Presentation Facility User's Manual (093-000168)
TRENDVIEW® Graphics Charting Package User's Manual (069-700008)

MS™-DOS

Using MS™-DOS on DESKTOP GENERATION™ Systems (069-100025)

Pascal

SP/Pascal Programmer's Reference Manual (069-400203)

PL/I

Plain PL/I (093-000216)
PL/I Reference Manual (093-000204)

Record Management Software (all included with GLDP)

INFOS® II System User's Manual (AOS) (093-000152)
INFOS® II Storybook (069-000019)
Sort/Merge Utility User's Handbook (AOS) (093-000176)
Sort/Merge with Report Writer User's Manual (093-000155)
SWAT® Debugger User's Manual (093-000258)

Contents

1 Introducing AOS Software

2 Installing AOS for the First Time

4 Using the CLI: A Session with AOS

5 Understanding AOS File Structure and Commands

6 Starting CEO

7 Backing Up and Restoring Files

8 Adding New Software, Software Revisions, and New Hardware to Your System

9 Using the SED Text Editor and Other GLDP Products

10 Programming with DG Computer Languages

13 Using Graphics

14 Formatting and Copying Diskettes

15 Responding to Errors and Error Conditions

Glossary

Tables

Table

Illustrations

Figure

Introducing AOS Software 1

Read this chapter when

- you want some background on the Advanced Operating System (AOS);
- you want to know what programs are included and available with AOS, and what these programs do.

The major sections in this chapter are

- What is the AOS Operating System?
- How Do I Install AOS?
- How Do I Work with AOS?
- What is a File?
- What Programs are Available with AOS?
- Cautions and Keyboard Control Characters
- If You Make a Mistake
- What Next?

What is the AOS Operating System?

An operating system is a large group of computer instructions that allows you to communicate with a computer. You type *commands* on a terminal keyboard; the operating system translates these for the computer; the computer does what the operating system directs, then tells the operating system it is done; and the operating system displays a suitable message on the terminal screen.

The operating system is the lowest level of computer software; it *supports* higher levels, like word processors or computer languages.

AOS is a general-purpose operating system that runs on DG ECLIPSE® computers — including DESKTOP GENERATION™ Model 10/SP, 20, and 30 computers.

AOS is a timesharing system. It can serve up to five people at one time — each using a terminal and doing word processing or other operations. AOS also supports *batch* operations, running jobs without human interaction or attention.

AOS can do these things because it is a multiprogramming system: it can run many programs simultaneously. Each running program is called a *process*.

Each process is like a complete computer system. It has up to 64 Kbytes of main memory; it often has its own terminal; and it can use devices like a disk or printer. AOS can manage up to 64 concurrent processes.

AOS is a secure system. It allows only authorized people to log on to user terminals. An authorized person is one for whom a *user profile* has been created by the user profile editor program.

But AOS does not *require* multiple terminals. It can manage a single-terminal desktop system very well.

What Hardware Does AOS Support?

AOS requires

- a Model 10/SP, 20, or 30 computer with at least 1/2 megabyte (Mbyte) of main memory;
- a model 6271 15-Mbyte hard disk;
- a model 6267 single-slot or model 6268 dual slot 368-Kbyte diskette unit; and
- a system console.

Over the minimum hardware, AOS can support

- a 16-color system console, on Model 10/SP systems;
- a multiplexor (USAM, model 4463) with up to four lines, to communicate with another system, a printer, a plotter, or user terminals;
- additional main memory, up to a total of 2 Mbytes;
- a second 15-Mbyte hard disk, for a total of 30 Mbytes of hard disk storage;
- a model 6301 38.6-Mbyte hard disk;
- a second model 6301 38.6-Mbyte hard disk, for a total of 77.2 Mbytes of hard disk storage;
- a dot-matrix serial printer, model 4434;
- a letter-quality printer, model 4518;
- a color plotter, model 4435;
- a cartridge tape unit;

and other devices.

How Do I Install AOS?

Your desktop hardware must be connected and ready to run as described in the hardware manuals named in the Preface.

Then, using the AOS diskettes from DG, you'll format your hard disk(s), install AOS, and install the rest of the AOS and support software (like the XODIAC™ networking system and CEO® Electronic Office) on the hard disk. The next steps are to configure all products and bring them up together; and, optionally, to create *user profiles* for other people who will use the system.

Generally, you will need to install all AOS software on the hard disk only once. You may need to repeat parts of the procedure if you acquire new hardware or new software.

How Do I Work with AOS?

If the system is not running, you'll start it up — a simple procedure that involves pressing a switch and typing a number, system name, date, and time on the system console.

Then — using either the system console or a user terminal — you'll type commands that instruct the system to do what you want. (To use a terminal other than the system console, you will *log on* first.)

The first command you type will tell the Command Line Interpreter (CLI) to ready the system for users. CLI commands can also execute other programs — like the CEO Electronic Office. If you choose to execute another program, you'll then give commands to *that* program.

Eventually, you'll want to stop working with the computer. You can simply walk away; or you can log off (if on a user terminal); or you can shut the system down by typing commands on the system console.

That's all there is to it.

What is a File?

A *file* is a collection of information stored under a name called a *filename*. This information can be *records*, like customer names, sales figures, and/or accounts receivable. It can also be the text of a report; or it can be a program: a series of computer instructions. The CEO system, the CLI, and the operating system itself are programs.

Files are stored on hard disks and diskettes. You'll be using them extensively. A file can be very large or very small: the largest file can store four billion bytes; the smallest, zero bytes.

Filenames

Each file is identified by a *filename*. AOS filenames can be from 1 to 31 of the following characters: upper- and lowercase letters, numbers, period (.), dollar sign ($), question mark (?), and underscore (_). For example,

```
trans_$income_received?may.10
```

is a valid (if cumbersome) filename.

A period and the last two or three characters in a filename often identify the contents of the file. For example, program files that you can execute usually end in

```
.PR
```

And CLI macro filenames end in

```
.CLI
```

A macro is one or more commands placed in a file for easy execution. Generally, the .PR and .CLI end characters (suffixes) are the only ones you need to know about. Filename suffix conventions are further described in *Learning to Use Your AOS System*, 069-000018, and the *Command Line Interpreter User's Manual*, 093-00010122 (under "extensions").

Directory Files

A *directory* is a file that contains other files. The main advantage of directories is that they allow you to group files by *category.* You can create and use them at will to help organize your files. For example, a directory named LETTERS could contain all letters; a directory named REPORTS all reports, and so on. Within each directory, you can create other, subordinate directories for more explicit subcategories.

AOS allows you to create directories within other directories, up to a total of eight levels. Directories are extemely useful; in fact, AOS uses them for its own files.

When you are using the CEO Electronic Office, files are called *documents.* Directories are called *drawers* and *folders.* CEO uses these terms for compatibility with the office environment.

What Programs are Available with AOS?

With your desktop computer, you bought the AOS operating system. Many other software programs are available with AOS.

The programs shown in Table 1-1 are included with every AOS system. The names proceed alphabetically.

The programs shown in Table 1-2 are included in the optional AOS *General Language Development Package (GLDP).*

The *products* shown in Table 1-3 are available as options with desktop systems.

Table 1-1 Programs included with AOS (continues)

Program Name	Comments
BASIC language	This is a modern, ANSI-standard BASIC called MP/BASIC. It is introduced in Chapter 10.
Command Line Interpreter (CLI)	The CLI is the main program you use to communicate with the AOS system. When AOS starts up, it runs the CLI on the system console. It can also run the CLI on user terminals. The CLI has over 100 commands, some of which execute other programs. The primary CLI chapters are 4 and 5.
Disk Formatter programs	Any hard disk or diskette must be formatted before it can be used with AOS. For this, there are two Disk Formatters. One formats hard disks; you will use it rarely. The other formats diskettes; you may use it pretty often.
DOWN.CLI macro	A CLI macro file (not program) that brings down the multiuser environment (shuts down the printer, etc.). You will always use DOWN before shutting down AOS.
Error message file	This file, named ERMES, contains text messages for AOS to match with error numbers. The text messages are more helpful than error numbers.
EXEC (Executive) program	EXEC manages the multiuser environment, the printer, user logon, and so on. The UP.CLI macro starts EXEC; DOWN.CLI stops it.
FIXUP disk fixer	When a computer system shuts down abnormally (as on a power failure), any disk or diskette that was in use ("open") must be closed (fixed) before it can be used. FIXUP does this: it closes the hard disk and any diskettes that were in use when the abnormal shutdown occurred, allowing AOS to be restarted. One version of FIXUP closes the hard disk; another closes diskettes (if any were open).

Table 1-1 Programs included with AOS (concluded)

Program Name	Comments
INFOS® II file manager	INFOS_II is a file management system that allows record access by any of many different criteria. It allows certain data management programs and CEO to run. You will probably not be using INFOS II directly. INFOS II comes on its own set of diskettes.
Printer manager program	The printer manager (spooler) program (XLPT), works with EXEC to ensure that the printer prints files in an orderly way.
Profile editor program	The user profile editor (PREDITOR) creates user profiles: files that allow people to and log on to user terminals (if any) and use the printer. The profiles also enforce file access rules — preventing one user from trespassing on another's files. A macro named PROFILE.CLI simplifies profile creation.
Sort/Merge	The Sort/Merge program allows you to sort and edit records, but it requires some computer experience to run. Sort/Merge is introduced in Chapter 11. It comes on its own set of diskettes. CEO requires Sort/Merge.
System	The AOS operating system manages devices, like terminals, disks and printers, for people.
System installer	The Installer program, named INSTL, reads the AOS system from a DG-supplied diskette and installs it on the hard disk.
System support program	These include a program named the Ghost that helps AOS serve users, and a program named PMGR (peripheral manager) that manages terminals.
UP.CLI macro	This is a CLI macro file that brings up the EXEC program, printer queues, and enables user terminals (if any) for people to log on to. You will usually type UP↲ after starting AOS.
Utilities directory	UTIL, the utilities directory, holds copies of most AOS utility programs. Other DG software products are installed in UTIL or a directory below UTIL.

Table 1-2 Some programs in the General Language Development Package

Program Name	Comments
Debugger	A debugger is a system program that allows you to isolate problems in programs you have created. DG's high-level debugger is named SWAT®. SWAT can be helpful with programs written in a compiled language like FORTRAN; it isn't needed for BASIC programs.
Display program	The DISPLAY program can read any binary, text, ASCII, or EBCDIC file — including a diskette — and write the contents on the terminal or to a disk file.
Error message file	This is a larger version of the standard ERMES file, including messages for SED, Link, SWAT, and some computer languages not included in the standard ERMES.
File compare programs	There are two file compare programs: SCOM, which compares two text files and displays the lines that differ, and FILCOM, which compares two binary files and displays the differences by file address.
Library file editor (LFE)	A library file editor (LFE) allows you to create your own programming libraries.
Link program	The Link program is used to build programs written in a compiled language (like FORTRAN).
Macroassembler (MASM)	The macroassembler is needed for assembly language programming or for creating your own central error message file.
Text editor program (SED)	A text editor allows you to create and edit text; it is similar to a word processor. You will use the SED editor if you want to write programs. SED is introduced in Chapter 9.

Table 1-3 Some other software products available with AOS (continues)

Program Name	Comments
BASIC	Aside from the supplied MP/BASIC, there are two BASICs available on desktop systems. Extended BASIC is a traditional BASIC, and Business BASIC is a business-oriented BASIC with screen handling, ISAM file handling, and graphics capability. Using any BASIC with AOS is explained in Chapter 10.
CEO or CEO.WP	CEO is the CEO Comprehensive Electronic Office system, with Electronic Mail, Word Processing, Filing, and Calendar. CEO.WP is the Word Processor - Independent, without Mail, Filing, or Calendar. Both are introduced in Chapter 6.
COBOL	There are two COBOLs available on desktop systems: AOS COBOL and Interactive COBOL (ICOBOL). AOS COBOL works with the INFOS II file management system and runs on Model 30 systems only; ICOBOL has its own file management features and runs on any desktop system. Both COBOLs are introduced in Chapter 10.
CP/M-86 operating system	CP/M®-86*, the popular operating system for microcomputers, and dependent programs like WordStar can run under AOS. Details are in *Using CP/M on DESKTOP GENERATION Systems.*
FORTRAN	There are three FORTRANs available with desktop systems: FORTRAN 77 (F77), FORTRAN 5, and FORTRAN IV. They are introduced in Chapter 10.
MS-DOS operating system	MS™-DOS**, the operating system used by IBM's personal computer, and languages like GW BASIC can run under AOS. Details are in *Using MS-DOS on DESKTOP GENERATION Systems.*

**CP/M®-86 is a registered trademark of Digital Research.*

***MS™-DOS is a trademark of Micro Soft.*

Table 1-3 Some other software products available with AOS (concluded)

Program Name	Comments
PASCAL	The Sp/Pascal language is introduced in Chapter 10.
PL/I	The PL/I language is introduced in Chapter 10.
TRENDVIEW	TRENDVIEW® is a graphics and charting package, introduced in Chapter 13.
XODIAC	The XODIAC networking system allows your desktop computer to communicate with another DG system. It is described in Chapter 12.

Cautions and Control Characters

This section gives some cautions and hints to help you as you create and run an AOS system. Simply read it; don't do anything yet.

Power Switches and Connections

While AOS is running, *don't* press the computer power switch to off, or disconnect or unplug any wires connected to the computer or terminals.

If power stops to the computer when AOS is running, system users may lose work. Also, when the power is turned back on, you will need to run a disk fixing program (FIXUP) before you can restart AOS.

When AOS has been shut down normally, the system console displays the message ***SYSTEM SHUTDOWN***, followed by a number and *!* prompt. At this point, you can turn off power to the computer and user terminals (if any) if you want. Be sure to remove any diskettes that are inserted before you turn off the computer.

System Console

The system console is the terminal you use to start up, control, and shut down the AOS system. This terminal must be turned on and on line for startup, control, and shutdown.

When both the computer and the system console are powered on, the Model 20 or 30 system console should display an exclamation point (*!*) prompt. On Model 10/SP systems you see multiple messages; finally, the console says *SELECT LOAD DEVICE: 20H (FOR DISKETTE) OR 26H (FOR DISK)* and displays the *!* prompt. For example,

```
!
```

If you don't see this *!* prompt, make sure the system console is turned on. Press the NEW LINE key (shown in this book as ↲). If the *!* prompt appears, fine; you can proceed to start up AOS as described later in the book. If the *!* prompt doesn't appear, consult the hardware *Testing* manual shipped with this one.

While AOS or any program is running, *don't* type the break sequence (CMD and BREAK/ESC) on the system console, unless you want to stop everything. If a *!* prompt appears on the system console while AOS is running, someone may have typed the break sequence on this console. AOS is stopped (halted). You can have AOS continue by typing

```
P
```

next to the *!* prompt. Accidental breaks can be annoying, but do no real harm as long as you know how to recover from them; type P next to the *!* prompt.

Keyboard Control and Command Sequences

There are several keyboard control sequences that govern terminal display, interrupt program execution, and so on. You will probably need one or more of these as you work with AOS. In any case, it's helpful to know about them.

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

Table 1-4 lists the major control characters, special keys, and their functions.

Table 1-4 Control characters and special keys (continues)

Key(s)	What it Does
CMD key	In conjunction with the BREAK/ESC key, CMD creates a break sequence that stops the computer. To have AOS continue, type P↵). The on-line and break sequence issues are covered above.
CTRL-C CTRL-A	Interrupts execution of a CLI command. You'll find this sequence useful.
CTRL-C CTRL-B	Aborts the current program process, like the CEO Electronic Office or BASIC interpreter. Generally, avoid this sequence.
CTRL-C CTRL-E	Creates a memory-image break file and aborts your current process. Avoid this sequence unless you are debugging a program.
CTRL-D CTRL-D	Signals an end of file, which usually aborts your current process. Generally, avoid this sequence.
CTRL-O	Cancels terminal display. The command or program continues running, but does not display characters. To resume display, type CTRL-O again.
CTRL-Q	Resumes terminal display after it has been stopped by CTRL-S. If display was stopped by CTRL-O, CTRL-Q does nothing.
CTRL-S	Suspends terminal display. To continue display from where it stopped, type CTRL-Q. CTRL-S and CTRL-Q are useful when you want to read long files on a screen, or anytime the display is too fast to read. The command or program keeps running, but saves characters instead of displaying them.
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.

Table 1-4 Control characters and special keys (concluded)

Key(s)	What it Does
DEL key	Erases the last character typed. In certain programs, DEL shows an underline (_) for each character erased.
NEW LINE key	Tells the computer to execute your command. Usually, when you type a command, nothing will happen until you press NEW LINE (shown as ⏎ in this book).

If You Make a Mistake

AOS and its companion software have good error messages and error recovery. Your mistakes (everyone makes them) will usually provoke an error message of the form

CAN'T OPEN DISK, ERROR CODE n

or

FILE DOES NOT EXIST

which enable you to identify the problem and retry. But if you make what appears to be a critical mistake, you can usually restart the program from the beginning without problems. At worst, you'll need to run a disk fixing program (FIXUP).

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. For other error situations, go to the Chapter 14, "Responding to Errors and Error Messages".

What Next?

To create your first AOS system, proceed to Chapter 2. If you already have a system running, go to the topic you want. Chapter 6 explains CEO and Chapter 10 presents DG computer languages.

Installing AOS for the First Time

2

Read this chapter when

- you have just connected and tested your desktop system hardware and you want to install AOS on it;
- you want to install software other than AOS (for example, CEO or INFOS II);
- you want to install AOS on a blank hard disk.

This chapter tells you — step by step — how to install AOS and support software on your hard disk. The major sections are

- Getting Ready
- Formatting the Hard Disk
- Installing AOS on the Hard Disk
- Installing Other DG Software
- Defining the System Configuration
- Bringing It All Up
- Terminating the Multiuser Environment and Shutting Down
- Step Summary
- What Next?

Getting Ready

1. If the system console is off, turn it on: use the switch behind the screen, on the right.

   Turn on power to the cartridge tape module (if present) and the second hard disk (if present).

   Turn computer power on, using the main power switch on the upper right of the computer module. On a Model 10/SP, the system console screen should show *SELECT LOAD DEVICE* message. Then — on any system — the console should display a *!* prompt.

   If nothing shows on the screen, check the brightness (contrast) control under the right front corner of the screen unit: put the control in a central position. If this doesn't help, make sure that the unit is plugged in and that connections are secure. If you still don't get any result, see Chapter 15, the error chapter. (Terminals other than the system console don't matter yet.)

2. Have all your AOS and other software diskettes ready, in order, as shown by the diskette labels. Each label has a UPC bar code and several lines of text. The first line gives the diskette part number, product model number, and diskette serial number. The second line gives the product name. The last line gives the diskette sequence number in the set and the product revision (REV). For example, for the first AOS diskette:

```
                                   product   diskette serial
                  diskette         model     number (unique for
                  part number      number    each diskette)

                  082-000375-00    30506T    nnnnnnnnnn
   product name   AOS PREGEN
     copyright    COPYRIGHT 1976 ....................
                  ...................................
                  DISKETTE  1  OF 4    REV 6.nn

                  diskette sequence   product
                  in set              revision
```

For a Model 10/SP system, you'll use the diskette labeled AOS D200 EMULATOR right after the last AOS PREGEN diskette. The names of most software products available with AOS start with AOS; for example, AOS INFOS II RUNTIMES, AOS SORT/MERGE RUNTIMES, and AOS MP/BASIC. The part of the name that follows AOS identifies each product.

For certain Model 10/SP systems (for example, a 10/SP with a German keyboard), a diskette labeled "BOOTABLE D200 EMULATOR" is packed with the computer. If you find such a "BOOTABLE" diskette, this is the first diskette you'll use. If there is no "BOOTABLE" diskette, the first diskette you'll use is AOS diskette 1 of 4.

3. Insert the appropriate diskette as follows. (If there are two diskette slots, use the rightmost one.)

3a. Turn the control arm beside the slot counterclockwise until the arm is vertical.

3b. Remove the diskette from its outer envelope. Don't try to remove the inner envelope — the diskette must remain in this.

3c. Hold the diskette by the edges and examine it. One side has a paper label (saying AOS) and the other is blank. On each side, the envelope is cut away to expose part of the diskette surface. Just a reminder — *don't touch the diskette surface.* The oil on your finger could make that part of the diskette unreadable. One edge of the diskette has a small notch (about 1/4 x 1/4 inch). This is the *write-enable* notch. When this notch is uncovered, data can be written to the diskette. When the notch is covered (as with opaque tape), the diskette is write-protected: it cannot be written to. Do not write-protect any diskettes from which you intend to install software.

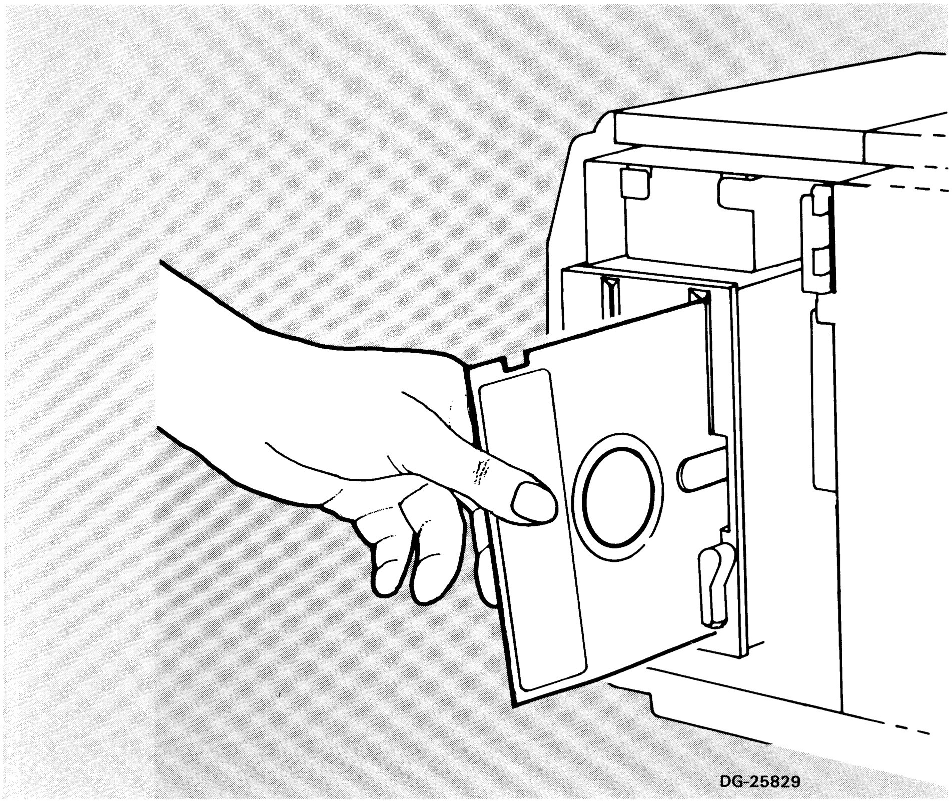


Figure 2-1 Inserting a diskette

3d. Hold the diskette with the write-enable notch up and your fingers on the label. Slide it into the unit slot as shown in Figure 2-1. (With two units, use the rightmost one.) The diskette should slide in smoothly and come to a firm stop.

3e. Turn the control arm beside the slot clockwise until the arm is horizontal. This locks the diskette in the unit.

4. Go to step 5 (unless you found a diskette labeled "BOOTABLE D200 EMULATOR" in step 2). If you found this "BOOTABLE..." diskette, program load from it: 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 beep and display the *!* prompt again.

4a. Remove the emulator diskette from the unit and replace it in its envelope. Now insert AOS diskette 1 in the unit in the same way you inserted the emulator diskette (step 3).

You're ready to format the hard disk.

Formatting the Hard Disk

Before AOS can use a disk, a program called the Disk Formatter must run on the disk. The Formatter prepares the disk by writing information that AOS needs. It also checks the disk surface for *bad blocks* (flawed areas that won't hold information).

Running the Disk Formatter is easy, but takes some time. It takes about 40 minutes to check the entire surface of a 15-Mbyte disk, and even longer for the 38.6-Mbyte disk.

Mistakes and Errors

If you type an incorrect answer to a question, and have not yet pressed ↲ (NEW LINE key), press the DEL key to erase the wrong characters. The program will display an underline (_) for each character erased.

To restart the program, type CTRL-C CTRL-A (press the CTRL key, hold it down, then press C, then press A). And go to step 6. To abort the program, type the break sequence (CMD and BREAK/ESC keys) and return to step 5.

If the Disk Formatter reports a disk or other error, check the error message in Chapter 15, the error chapter.

Disk Formatter Dialog

5. Now you need to program load from the diskette. Next to the *!* prompt, type

```
!   20H
```

The diskette run light should glow, to indicate motor action. Wait 20 to 30 seconds. The screen displays

```
WELCOME TO THE DG DESKTOP GENERATION

PROGRAM NAME?
```

(If nothing happens after a minute or so, the diskette may not be inserted correctly. Remove it from the slot and reinsert it, with the write-enable slot up and your fingers near the label edge. Then turn the control arm up to engage the diskette. Type the break sequence (CMD and BREAK/ESC keys) and repeat step 5. If this doesn't help, remove the diskette, turn power off and on, insert the diskette, and repeat step 5. If this doesn't help, consult Chapter 15, the error chapter.)

6. Type the Disk Formatter filename, which is DFMTR, and press ↵ (NEW LINE key).

   ```
   DFMTR ↵
   ```

   Wait 20 seconds or so. The Disk Formatter starts up and displays

   ```
   AOS DISK  FORMATTER (DFMTR) REV x.xx

   DO YOU WANT TO SAVE ANY INFORMATION ON YOUR DISK ?
   TYPE Y (YES) TO SAVE INFORMATION, OR
   N (NO) TO ERASE YOUR DISK.
   ```

7. You are just starting out, so the hard disk is blank and has no information on it. To prepare it for AOS, you must answer N↵ for no. Type

   ```
   N ↵
   ```

   (If you type an incorrect answer to this or the next question, type the break sequence (CMD and BREAK/ESC keys) and return to step 5.)

8. The Disk Formatter asks for confirmation (just in case you made a mistake and your hard disk *does* have files):

   ```
   THIS WILL ERASE ALL FILES FROM YOUR DISK.
   TYPE Y (YES) IF YOU ARE SURE YOU WANT TO DO THIS, OR
   TYPE ANYTHING ELSE TO START OVER.
   ```

   To confirm, type Y↵

   ```
   Y  ↵
   ```

9. Next, the Disk Formatter wants to know

   ```
   HOW MANY HARD DISKS DO YOU HAVE?
   ```

   If you don't know how many hard disks your system has, check for a separate disk module with its own power supply and power switch. If there are two disks, make sure the second one is turned on.

Most desktop systems have only one hard disk; if this is true for yours, type 1↵. If you are unsure of the number, try 2↵. If the Formatter prints *FATAL DISK ERROR, STATUS...* this means that it found only one disk running. If there are two hard disks, turn on the second disk. If you can't find another disk to turn on, assume that you have one disk. Return to step 5.

The Disk Formatter now checks the disk surface for flaws. You will see the display

```
ANALYZING YOUR DISK FOR BAD BLOCKS
NUMBER OF OPERATIONS REMAINING: n
```

n starts at 1946 for the 15-Mbyte disk, and at 4715 for the 36.6-Mbyte disk. If you have two hard disks, the Disk Formatter finishes one, then does the next. It usually takes 40 to 50 minutes per 15-Mbyte disk, and more than double that for the 38.6-Mbyte disk: (80 to 100 minutes with two hard 15-Mbyte disks). When the Formatter finishes the surface-checking operations, it will be done. A good way to spend the waiting time would be to read the rest of this chapter — or other parts of the manual. Or, you can walk away and leave the system console if you want.

When it has finished the checking operations, the Formatter displays

```
TOTAL BAD BLOCKS: n

DONE!
n
!
```

Ignore the number *n* that appears after *DONE*.

You've formatted the hard disk(s).

Installing AOS on the Hard Disk

Next, you'll run the Installer program to copy the AOS system from diskette onto the hard disk. (AOS must be on a hard disk to run; it can't run from a diskette.) Then, you'll bring up AOS and load some files from diskette. Diskette 1, the same diskette with DFMTR on it, should be in the right unit.

10. The *!* prompt is showing on the system console. You must program load from diskette again. Type

    ```
    !    20H
    ```

    Wait about 20 seconds. The system console displays

    ```
    WELCOME TO THE DG DESKTOP GENERATION

    PROGRAM NAME?
    ```

    (If nothing happens after a minute or so, type the break sequence (CMD and BREAK/ESC keys) and repeat this step. If break doesn't help, flip the diskette retaining arm up and down, try another break sequence, and repeat this step. If these don't work, consult Chapter 14, the error chapter.)

11. Type the Installer filename, INSTL

    ```
    INSTL⤶
    ```

    Wait 20 seconds or so. The Installer starts up and displays

    ```
    AOS INSTALLER (INSTL) REV x.xx

    -- INITIALIZING THE DISK
    ```

    The Installer now copies two loader programs (called bootstrap programs) from diskette to the hard disk. This takes 20 to 30 seconds. Then it displays

    ```
    -- STARTING SYSTEM INSTALLATION
    ```

    The Installer copies the AOS system from diskette to the hard disk. This takes 2 to 3 minutes. When done, the Installer says

    ```
    -- SYSTEM INSTALLED

    DONE!
    n
    !
    ```

Ignore the *n* — here and from now on.

The AOS system is installed on your hard disk.

12. Now you will bring up AOS. The *!* prompt is showing on the system console. You must program load again, this time from the hard disk. Type

    ```
    !   26H
    ```

    ```
    WELCOME TO THE DG DESKTOP GENERATION
    ```

    ```
    PROGRAM NAME?
    ```

    Note the faster response — about a second from the hard disk, instead of about 20 seconds from diskette.

13. To specify the AOS system, type

    ```
    AOS ↵
    ```

    (If you see an *ABORT* error message, don't worry. You may have made a mistake typing the name AOS. Return to step 12 and try again — type AOS and ↵.)

    This starts AOS initial operations. It displays

    ```
    INITIALIZING SYSTEM DATA ON DISK --
    ```

    After a minute or so, it displays

    ```
    INITIALIZATION COMPLETE

    AOS REV xx.xx

    DATE (MM/DD/YY)?
    ```

14. Type the date as numbers for month, day, and year. Insert a space or slash before the day and year. For example, for November 16, 1984, type

    ```
    11  16  84 ↵
    ```

    ```
    TIME (HH:MM:SS)?
    ```

15. Type the time, based on a 24-hour clock, in hours, minutes, and seconds. Insert a space or colon before the minute and second numbers. (Actually, minutes and seconds are optional. If you omit them, the system uses 00:00.) For example, for 2:30 p.m., type

```
14 30 ↵
```

Wait 20 or 30 seconds. Then

```
DO YOU WANT TO INSTALL AOS SYSTEM SOFTWARE [N] ?
```

You do want to install AOS software, so type

16.
```
Y ↵
```

```
MOUNT AOS DISKETTE 2 -- TYPE NEW-LINE WHEN READY
```

17. Change diskettes: remove AOS diskette 1 and insert AOS diskette 2 (rotate the control arm down, remove the diskette, place it in its envelope, and then insert diskette 2, as described in step 3.)

18. Press

```
↵
```

19. AOS now reads the second diskette and copies files from it to the hard disk. This takes 1 to 2 minutes. Then

```
AOS  CLI  REV  n  date  time
)
```

Congratulations! You've brought up AOS. The Command Line Interpreter (CLI) is running on the system console. The CLI's prompt,), tells you that it is ready for a command.

(If you see a *FATAL ERROR 25* message, a needed file wasn't loaded. Perhaps you forgot to answer `Y↵` to the *INSTALL AOS SOFTWARE* question. In any case, run Emergency Shutdown by typing I then `14R`; and return to step 12 to try again.)

The CLI is more versatile than the Disk Formatter or Installer. It has many commands and fine error handling. It also has a HELP feature you can use after loading the next diskettes. You can interrupt executing CLI commands by typing CTRL-C CTRL-A (press the CTRL key, hold it down, and press the other character). As always, you can delete characters with DEL and erase bad lines with CTRL-U; and you can type CTRL-S to stop the screen display and CTRL-Q to continue it (useful for reading long files on a terminal).

20. Change diskettes again: remove AOS diskette 2 and insert AOS diskette 3, the same way you did it before.

21. Now to install AOS support software from the remaining diskettes. Type

```
) INSTALL  AOS ↲
```

Please insert the diskette with the desired DG-supplied AOS software into your primary diskette unit -- @DPM0.

Press NEW-LINE when you are ready to begin installation

22. To continue, press

```
↲
```

Beginning AOS software installation...

. (Filenames copied from the diskette. Don't worry if the
. names appear sporadically in groups — this is normal.)
. If you see the message *PHYSICAL UNIT FAILURE...Disk 1 must be mounted*
. *first,* this means you inserted the wrong diskette. Check the
. diskette numbers, swap diskettes, and try step 21 again.)

Installing files from a diskette takes about two minutes. After all the diskette's files have been installed, the system console prompts for a diskette number that is 2 *less* than the number on the AOS diskette label. For example, as shown, it prompts

This disk is exhausted. Mount disk 2 and press NEW-LINE to continue.

The AOS diskette you insert is labeled 4 in the AOS diskette set. (This inconsistency occurs because the installation program starts installing with AOS diskette 3, not AOS diskette 1.)

23. Remove the diskette from the primary unit and insert the next AOS diskette (if any). Press ↲.

24. Repeat step 22 and 23. The program then says

```
AOS software installation is complete. Please remove diskette from the
primary unit.
```

(If you insert diskettes in the wrong sequence, the program will display a *Wrong disk...* message. Remove the diskette, insert the correct one, and press ↵ to try again.)

25. With a Model 20 or 30, skip to step 26. With a Model 10/SP, you must install the emulator *file* on the hard disk. Remove the AOS diskette from its unit and insert the diskette labeled "AOS D200 EMULATOR".

25a. To install the emulator, type

```
) INSTALL  EMULATOR ↵
```

```
Please insert....
Press NEW LINE when ready to begin installing EMULATOR.
```

Press ↵

The installation macro will ask you the following questions about your system console and character set.

25b. *Is your system console monochrome? {Y N} ''Y''*

If you have a 13" color display system console, type N ↵. In most cases, the console screen can display only one color (it is monochrome); if so, press

↵

*

25c. Then the system displays a list of 8 character set options

```
1. DANISH-NORWEGIAN
2. FRENCH
3. GERMAN
4. ITALIAN
5. SPANISH
6. SWEDISH-FINNISH
7. SWISS-GERMAN
8. UNITED STATES-UNITED KINGDOM

Which character set will you need? { 1 2 3 4 5 6 7 8 } ''8''
```

Type the number of the desired national character set. For example if you want the FRENCH character set answer

```
) 2 ↵
```

25d. Your answers to the preceding questions tell the macro which emulator file to install. When the macro loads the emulator you specified onto the disk, the following message appears.

```
EMULATOR Software Installation is complete.....
```

From now on, AOS can load the emulator automatically on startup, you don't need to install it again. Remove the diskette from the unit and return it to the outer envelope.

26. If you have a model 20 or 30 system, you're done! You've built AOS on your hard disk(s). Make sure all the AOS diskettes are in their covers; and store them in a safe place. You will need them again if you want to restore any of these AOS files to your hard disk. To learn more about these files, see Chapter 1, Table 1-1. Aside from INFOS II and Sort/Merge, all programs shown in Table 1-1 are on your hard disk.

27. Now, if you have a Model 10/SP system, you must shut down AOS, power down, and restart to load the emulator. Type the following commands:

27a. `) BYE ↵`

27b. `DO YOU REALLY WANT TO SHUT ... DOWN? Y ↵`

```
SYSTEM SHUTDOWN
!
```

27c. Turn power off to the cartridge tape (if present), the second hard disk (if present), and then the computer unit. Power up the devices first and then the computer unit.

27d. `!    26H`   (Type 26H)

```
WELCOME TO THE DG DESKTOP GENERATION
```

27e. `PROGRAM NAME?       AOS ↵`   (Type AOS ↵.)

```
AOS REV n
```

27f. `DATE(MM/DD/YY)?        12 14 84 ↵`   (Type the correct date.)

NOTE *With a French or Italian terminal, use the numeric keypad on the far right of the keyboard to answer the DATE and TIME questions. The normal numbers will not work.*

27g. `TIME (HH:MM:SS)        15 20 ↵`   (Type the correct time, 24-hour clock.)

27h. `DO YOU WANT TO INSTALL AOS ...SOFTWARE [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 won't appear in lowercase, return to step 25.)

Installing Other DG Software

The next steps are to install other DG software products you received with your AOS system. For AOS and certain products to work together, the products must be installed on the hard disk *before* you run the configure macro to tailor your AOS system. If you have any of the following products, you must install them now.

XODIAC Network System
INFOS II File Management System (ships with AOS)
Sort/Merge Program (ships with AOS)
CEO Electronic Office or CEO Word Processing - Independent

The products are relatively easy to install, using the supplied INSTALL.CLI macro. (This and another macro supplied with the product do nearly all the work involved.) Because you may not have all the products described, we will stop numbering the steps here. But you'll still be able to follow the directions.

Installing the XODIAC Networking System

To install XODIAC, get the diskette labeled AOS XODIAC PREGEN and insert it in the rightmost diskette unit (as described in step 3). Then type

```
) INSTALL  XODIAC ↲

Please insert the diskette with the desired DG-supplied XODIAC
software into your primary diskette unit -- @DPM0.

Press NEW-LINE when you are ready to begin installation    ↲

Beginning software installation...

.
. (time passes as XODIAC files are copied to the hard disk)
.
XODIAC software installation is complete. Please remove diskette
from the primary unit.

)
```

You've installed XODIAC. Remove the diskette from the unit, insert it in the envelope, and put it in a safe place.

Now you can install INFOS II, Sort/Merge, and/or CEO (if you have them).

Installing INFOS II

INFOS II is a file management system. The CEO Electronic Office requires INFOS II; but the CEO Word Processor - Independent (the word processor only) does not require it. The AOS COBOL language requires INFOS II; but neither Interactive COBOL, nor MP, Business, nor Extended BASIC, nor the FORTRAN(s), nor CP/M®-86-based or MS-DOS-based software require it.

You should install INFOS II if

- you have CEO (the full system, not CEO Word Processing - Independent);
- you have AOS COBOL; or
- if you want sophisticated indexed sequential (ISAM) file management, via a FORTRAN, BASIC, or PL/I program.

If you need to install INFOS II, you should do it now. If you don't need to install INFOS II, put the INFOS II diskettes in a safe place and skip the rest of this section.

To install INFOS II: Get the diskette labeled AOS INFOS II RUNTIMES and insert it in the primary (rightmost) unit. Then type

```
) INSTALL  INFOS ↲

Please insert the diskette ...

Press NEW-LINE when you are ready ...

.
. (system verifies INFOS files copied to the hard disk)

INFOS software installation is complete. Please remove diskette
from the primary unit.
)
```

You've installed INFOS II on your hard disk. Remove the last diskette from the unit and place it with the others.

Now you can install Sort/Merge, if needed.

Installing Sort/Merge

The Sort/Merge program allows you to rearrange and reformat records; it requires some computer experience to run directly. But, Sort/Merge is required by any CEO product (either the CEO Electronic Office or Word Processing - Independent), and by AOS COBOL. So you should install Sort/Merge if you have CEO or AOS COBOL, or if you want to sort, rearrange, or reformat records. Sort/Merge must be installed before CEO software.

To install Sort/Merge, insert the diskette labeled AOS SORT/MERGE RUNTIMES in the primary (rightmost) diskette unit. Then type

```
) INSTALL  SORT ↵

Please insert the diskette ...

Press NEW-LINE when you are ready to begin installation    ↵

Beginning SORT software installation...
.
. (time passes as SORT files are copied to the hard disk)
.
SORT software installation is complete. Please remove diskette
from the primary unit.

)
```

You've installed Sort/Merge. Remove the diskette from the unit, insert it in the envelope, and put it in a safe place.

Before proceeding, type

```
) SPACE  : ↵
```

The `SPACE :` command tells you about disk space on the hard disk: how much you have (MAX), how much is currently used (CUR), and how much remains free for storage (REM). This is very useful information. And, by typing SPACE : after installing each product, you'll learn the amount of space it consumes. This may be useful in the future.

Write "After AOS, INFOS, and Sort" and the MAX, CUR, and REM figures on a piece of paper.

Now you can install CEO, if you have it.

Installing CEO

There are two versions of CEO — the CEO Electronic Office (full product) and the CEO Word Processor - Independent. Both products require multiple diskettes, but the full CEO product requires many more diskettes than the Word Processor.

To install CEO, get the first AOS CEO product diskette and insert it in the primary unit. Then

With CEO Electronic Office, type	**With CEO Word Processing-Independent, type**
`) INSTALL  CEO ↵`	`) INSTALL  CEO.WP  ↵`

```
Please insert the diskette labeled ...
.
Press NEW-LINE when you are ready to begin installation. ↵
.
. (System verifies CEO files copied to the hard disk. If you see the
. message ILLEGAL DUMP FORMAT... Disk 1 must be mounted first,
. remove the current diskette, insert CEO diskette 1, and try
. the INSTALL step again.) .
.
This disk 1 is exhausted. Mount disk 2 and type NEWLINE to continue.
```

Remove the diskette from the unit, replace it in the envelope, and insert the next diskette in the unit. Then press ↵.

```
.
(system verifies files copied)
.
```

When the program prompts for another diskette, remove the diskette and replace it with the next one. After all the CEO files have been installed, the INSTALL macro says

```
CEO...software installation is complete. Please remove diskette
from the primary unit.
)
```

All DG-supplied CEO files have been installed in the CEO directory. You can remove the last diskette from the unit and place it with the others.

You've installed CEO on your hard disk. CEO is the last product that *must* be installed before you configure your system. You can install other products now, or skip them and go to "Defining Your System Configuration", and install them later. Before proceeding, type `SPACE :↵` again; and note "After CEO" and the disk space usage figures on your paper.

Installing the General Language Development Package (GLDP)

The General Language Development Package (GLDP) is designed for people who want to write programs in a compiled language (FORTRAN, COBOL, and so on). GLDP includes a text editor named SED, a Link program, a debugger named SWAT, and other programs.

There are several components to the GLDP package, including the entire development versions of INFOS II and Sort/Merge. Thus, installing GLDP requires several steps. To start, get the General Language Development Package diskettes, insert the first one in the primary unit, and type

```
) INSTALL  GLDP ↵
.
Please insert the diskette labeled ...
.
Press NEWLINE when you are ready to begin installation. ↵
.
. (system verifies program and other files copied to the hard disk)
This disk is exhausted. Mount disk n+1 and type NEWLINE to continue.
```

Remove the diskette from the unit, return it to the envelope, and insert the next diskette. Then press ↵. Repeat this step until you see

```
.
The standard error message file contains runtime error
messages for applications written in FORTRAN 77.
.
Do you want to be able to modify and/or add to the
standard set of runtime error messages?

Please type Y to retain the necessary files.
Type N to free the disk space required by those files? {Y/N}  "N"
```

Adding your own custom codes and messages to the standard message set requires many .OB files, which consume a lot of disk space. Doing this also requires planning and expertise; it's generally useful only when you envision a large application that consists of *many* programs.

By default, runtime errors in any program are handled well; and your programs can have whatever text you write displayed on an error condition. So, in most cases, you should say no, by pressing ↵ or typing **N**↵. You can always install GLDP later and say yes if you need the files.

Depending on your answer, the GLDP macro displays explanatory text and deletes files.

Installing the Development Version of INFOS II

The next step is installing the development version of INFOS II. If you already installed INFOS II, the macro says

```
INFOS II is already installed. To develop programs with INFOS,
you need the entire INFOS II product. I am proceeding to install
the entire product.

Please insert the diskette labeled ...

Press NEW-LINE when you are ready to begin software installation.
```

Insert the first of the GLDP INFOS diskettes and press ↵. You may see some `DELETED` messages as development INFOS files replace INFOS files already installed. Proceed through all of the GLDP INFOS diskettes, until the macro says

```
INFOS software installation is complete...
```

Now skip to the message that begins `GLDP software installation is complete...`

If INFOS II has not been installed, the macro asks whether you want to install it:

```
Do you want to install the INFOS II system {Y/N}?
```

You need INFOS II for the AOS COBOL language, or for ISAM file management with any other language. (For CEO, the streamlined INFOS II supplied with AOS is preferable because it consumes less space.) INFOS II is *not* needed for Interactive COBOL, any BASIC, or any program that does not attempt ISAM file access. The entire INFOS II takes a substantial amount of disk space, but this is okay if you really plan to use it — with relatively sophisticated programs. If your computer is not a Model 30, and you don't have CEO, you probably don't need the development INFOS II system. You can always install it later — if needed — by reinstalling the whole GLDP package. If you don't want the full INFOS II installed, type `N↵`. To install the full INFOS II, type `Y↵` and follow directions as the install macro and program prompt for INFOS II diskettes.

```
GLDP software installation is complete. ....

)
```

You've installed the General Language programs. Remove the last diskette from the unit, replace it in the outer envelope, and put all diskettes in a safe place.

Type `SPACE :↵` again. Then note "After GLDP" and the disk space usage on your paper. Later, you may want to delete the general language programs you don't need, as described in Chapter 9.

Now — if you have them — you can install other DG products, or you can skip ahead to configure your system.

Installing Other DG Products

Desktop system software includes a BASIC language interpreter called MP/BASIC. (The diskettes are labeled AOS MP/BASIC HFP.) You might want to install this next — since BASIC is such a useful language. Do it the same way as XODIAC — but specify `INSTALL MBASIC↵` instead. There are two diskettes.

Other DG software available with desktop systems includes FORTRAN 77, Interactive COBOL, and others. For CP/M-86 or MS-DOS see the pertinent manual.

Generally, you install any product just like you installed INFOS II or CEO software. You put the first (or only) product diskette in the primary diskette unit, then type `INSTALL productname↵`, where `productname` is one of these product names:

Product	**Name to Install by**
BASIC (Business)	BBASIC
COBOL (AOS)	COBOL
COBOL (Interactive)	ICOBOL
FORTRAN 77	F77
SP/Pascal	PASCAL

Other product names are shown in Chapter 5, under INSTALL.CLI. During installation, the program will prompt for other diskettes as needed. If you insert the wrong diskette, it will tell you of the error and allow you to change diskettes.

When done, the installation macro will say

```
xxxx software installation is complete....
```

And return you to the CLI prompt:

```
)
```

Put the diskette(s) back in their envelopes — and, if desired, type `SPACE :`↵ and note the new space usage figures on the original piece of paper. An overview of system file structure, with XODIAC, INFOS II, and CEO installed, appears at the beginning of Chapter 5.

To *use* the software, see the appropriate chapter: for the CLI, see Chapters 4 and 5; for CEO, see Chapter 6; for languages, see Chapter 10, and so on.

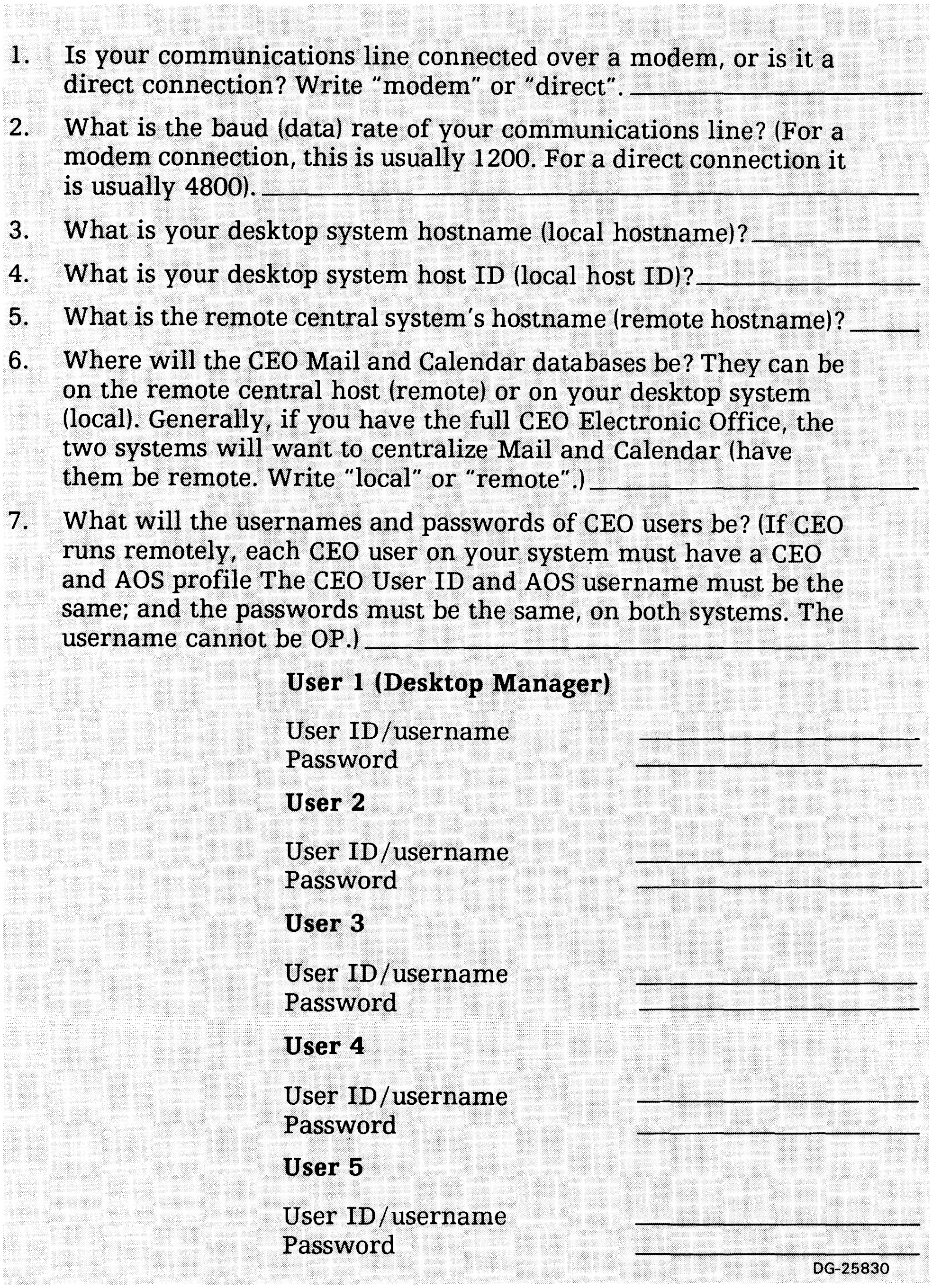
Defining the System Configuration

The next task is to define your system configuration: diskettes, communications lines, printers, plotter, and terminals, using the CONFIGURE macro. You can also use CONFIGURE to change an existing configuration (for example, if you acquire new hardware or want to change an existing configuration).

Part of the configuration procedure involves communication with another (central host) system. If your desktop system will be connected to another DG system (perhaps a larger, MV/Family system), the two systems must agree on certain things, like baud rates, hostnames, CEO User IDs, and AOS usernames and passwords. If your system will not be connected to another system, skip to the next section, "Configuration Dialog".

For the two systems to work properly together, you will need answers to the following questions. If you know the answers, write them in the space provided below, in Figure 2-2. If you don't know the answers, phone the person in charge of remote system operations, get the answers, and write them down. Use pencil for the passwords if desired (you can then erase them for security).

Fill in all of the blanks that apply. Having the answers handy will make your configuration, and future configurations, much easier.



1. Is your communications line connected over a modem, or is it a direct connection? Write "modem" or "direct". ________________

2. What is the baud (data) rate of your communications line? (For a modem connection, this is usually 1200. For a direct connection it is usually 4800). ________________

3. What is your desktop system hostname (local hostname)? ________________

4. What is your desktop system host ID (local host ID)? ________________

5. What is the remote central system's hostname (remote hostname)? ________

6. Where will the CEO Mail and Calendar databases be? They can be on the remote central host (remote) or on your desktop system (local). Generally, if you have the full CEO Electronic Office, the two systems will want to centralize Mail and Calendar (have them be remote. Write "local" or "remote".) ________________

7. What will the usernames and passwords of CEO users be? (If CEO runs remotely, each CEO user on your system must have a CEO and AOS profile The CEO User ID and AOS username must be the same; and the passwords must be the same, on both systems. The username cannot be OP.) ________________

User 1 (Desktop Manager)

User ID/username ________________
Password ________________

User 2

User ID/username ________________
Password ________________

User 3

User ID/username ________________
Password ________________

User 4

User ID/username ________________
Password ________________

User 5

User ID/username ________________
Password ________________

DG-25830

Figure 2-2 Summary of answers to remote configuration questions

AOS Configuration Dialog

To start the configuration procedure, type

```
) CONFIGURE ↲
```

Now, if AOS has already been configured, it asks if you want to change the AOS configuration. You need not change this to change a XODIAC or CEO configuration; so, if you want to change a XODIAC or CEO specification, say no (↲ for the default). If you want to change an AOS configuration, say yes (Y↲). To change any AOS configuration, you must respecify them *all.*

The dialog proceeds

```
In the following questions the values in braces are the
valid choices for the response. The value in the quotation
marks is the default response.

** Defining System Configuration **

Please type your system model number?
{10SP 20 30} "10SP"
```

Respond with the model of your desktop system: 20↲, 30↲, or ↲ for Model 10/SP. For a Model 10/SP, it skips the next question.

```
Is the system console D410-D460 compatible?
{Y N} "N"
```

DG DASHER® D410 and D460 terminals have special graphics capabilities. To use these capabilities, you must identify the terminal to AOS. If system console is a D410 or D460, type Y↲. Otherwise, for a D210, D211, or G300, press ↲ to say no.

```
How many diskette units do you have?
{1 2}  "1"
```

Desktop systems are available with one or two diskette units. Respond with the number your system has: type 2↲ or press ↲ for the default.

```
Do you want to reserve line 0 for communications?
{Y N} "N"
```

This question pertains if your system has an optional model 4463 USAM, a universal synchronous/asynchronous multiplexor. A desktop system can have no USAM, a one-line USAM, or a four-line USAM. A USAM line can be used for communications, a printer, a plotter, or a user terminal. "Communications" means interaction with another system, for example, via XODIAC networking or DG/GATE™ software.

If your system has a USAM, and you want to be able to communicate with another system (needed for the full CEO product or to log on to the other system), type Y↵. The macro then asks *Is line 0 is connected via a modem.* Based on the answer you wrote in Figure 2-2, type Y↵ (for yes) or ↵ (for no). It then asks *What is the baud rate of line 0 ....* Again, based on the value you wrote in Figure 2-2, press ↵ or type the correct baud rate.

If your system does not have a USAM, or if it will not communicate with another system, press ↵ or type N↵ to say no.

How many printers do you have?
{0 1} "0"

If your system has a printer, answer 1↵; otherwise press ↵ or type 0↵. If you answer 0↵, skip the next questions.

What is the maximum number of characters you want printed per line?
{16 - 255} "80"

This sets the *maximum* characters per line. If any line exceeds the maximum length, it will be truncated (cut off). Lines less than or equal to the maximum length will print normally. A model 4434 dot-matrix printer can print up to 160 characters in compressed mode, so, generally, for a 4434, specify 160↵. A model 4518 letter-quality printer can print up to 203 characters (compressed), so, for a model 4518, specify 203↵. A model 4433 printer can print up to 233 characters (compressed), so specify 233↵ for a model 4433 printer. To make use of these maximums, you may need to load paper that's wide enough. But — usually — it does no harm to specify the maximum for your printer.

The system default, if you don't give a number but press ↵, is 80 characters per line.

What is the maximum number of lines that can be printed per page?
{6 - 144} "66"

This determines the number of lines per page for printing. Generally, a printer can print 66 lines per page. Most printers print 6 lines per inch, which, for 11-inch paper, comes to 66 lines per page. If you have CEO, you *must* specify the maximum number of lines a page can accept; if you specify a different number, CEO users won't be able to print documents.

The *characters* and *lines* answers can be changed later. To temporarily change them, use the PRINTER_REDEFINE macro (Chapter 5); you need not rerun CONFIGURE. However, the values specified in CONFIGURE will be used every time the UP macro is used. To permanently change the values you must rerun CONFIGURE.

The Model 4434, multifunction dot matrix printer has firmware that allows you to change printer characteristics — specifically, form length, lines per inch, characters per inch, and the character set. See *Operating Model 10 and Model 10/SP Systems*, 014-000900 or *Operating Model 20 and 30 Systems}* 014-000903 for a full description.

Next, the macro displays

How many plotters do you have?
{0 1} "0"

If you have a model 4435 color plotter, type 1↵. Otherwise, press ↵ or type 0↵. *

How many user terminals do you have?
{0 - m} "0"

This answer gives the number of *user terminals* — in addition to the system console — connected to the system. The connection can be either direct or over a modem. The maximum number of user terminals a desktop system can support is four (less one for each communication line, printer, plotter, mouse, or data tablet). Type the number of user terminals — for example, 2↵ — or press ↵ if there is none.

Is the terminal on line n D410-D460 compatible?
{Y N} "N"

Line *n* means the terminal line number — 0, 1, 2, or 3 (for a four-line USAM board) on the cable connector attached to the USAM. If the terminal on this line is a DASHER D410 or D460, you must type Y↵ to enable use of its graphics features. If the terminal is not a D410 or D460, press ↵.

Is line n connected via a modem?
{Y N} "N"

A terminal can connect directly to your computer, or it can connect remotely via a modem. The modem connects the terminal to a phone line — allowing a person to use the computer from his home, for example. If you know that the terminal on this line will be connected via a modem, type Y↵; otherwise, press ↵.

Next, depending on your answer, the macro asks

What is the baud rate of ....?
{300 1200 2400 4800 9600} "n"

Baud is the rate at which a line or modem can transfer data, in bits per second. Each character requires 10 bits. The default for a terminal connected directly (not to a modem) is 9600 baud (960 characters per second). For a local terminal, unless you *know* that you want a different rate, press ↵ for the default.

The default rate for a modem is 1200 baud. Traditionally, a modem has one of two baud rates: 300 or 1200. Older and less expensive modems are often 300 baud; newer ones are 1200 baud. Be sure to type the correct rate of the modem here; or, for 1200 baud, press ↵. (At the remote site, the terminal's baud rate must be the same as the modem's. You can set the rate with very small switches — using a pencil point — on the terminal back panel.)

After you answer this baud rate question, the macro asks about the next terminal (if any), until you have specified all the user terminals.

XODIAC Configuration Dialog

Next, if XODIAC is installed, a XODIAC configuration macro executes automatically. (If not installed, the macro doesn't execute; skip this section.)

If XODIAC has never been configured, its macro asks about your local hostname, ID, and the remote hostname as in the following dialog.

If XODIAC *has* been configured, the macro gives you the option to keep the old answers. If XODIAC functions properly, and you don't want to change a hostname or ID, then keep the old answers. If you elect to change any answer, you must specify all three answers.

The XODIAC configuration questions are

Please specify your system's local hostname

This is the name by which the remote system will access your system. This name should be agreed upon by both systems, because it must be the same on both systems for the network to work. From Figure 2-2 above that you filled in, read the "local hostname" answer, and type it here. For example, SALES3↵.

Please specify your system's local ID

The host ID allows the network software to identify your system. It must be correct on both systems for the network to "know" your system. From Figure 2-2 above, get and type your local ID.

Please specify the remote system's hostname

This is the name by which your system, and CEO, will access the remote one. Like the local hostname, the name must be the same on both systems. Read the remote hostname from Figure 2-2 above, and type it here. For example, HOST↵.

This is the last XODIAC configuration question; the network macro now builds the needed files in directory :NET and returns control to the AOS CONFIGURE macro.

CEO Configuration Dialog

Now, if any CEO product is installed, the AOS macro executes a CEO configuration macro. If CEO isn't installed, the dialog ends; skip to "Creating AOS User Profiles."

If CEO is installed and has never been configured, its configuration macro asks about letter-quality printers, and so on, as in the following dialog.

If CEO *has* been configured, the macro asks whether you want to reconfigure. To keep CEO the way it is, answer NO↵. This retains the old CEO configuration (useful when you have run CONFIGURE to change an AOS or XODIAC configuration, and don't want to change CEO). If you answer NO↵, the dialog ends; skip to "Creating AOS Profiles".

The CEO configuration macro asks

```
Does your Desktop System have a letter-quality printer?
YES/NO
```

CEO needs to know this because it treats a letter-quality printer differently from other types of printers. If you have a model 4518 letter-quality printer, answer YES↵; otherwise type N↵.

Now, for the CEO Word Processor-Independent, the macro ends; skip to "Creating AOS User Profiles". For the full CEO product, the macro gives some background information, and asks about the desired configuration:

```
On a small system CEO can have two possible configurations.
Would you like to use CEO in the remote configuration?
YES/NO
```

If your desktop system is connected to a larger DG system (like an MV/Family machine), you can use CEO in one of two ways.

If you answer Yes to this remote configuration question, CEO will be configured to use the Mail and Calendar databases on the central host system. This has the advantage of centralizing Mail and Calendar information and allowing desktop systems to mail messages and schedule events with users on the central system and other connected desktop systems. CEO Mail and Calendar work as usual — they just run remotely.

For the remote CEO arrangement to work, it must have been agreed upon with the other system — as shown in your answers in Figure 2-2. To specify remote configuration, type YES↵. The macro may then ask for the hostname of the central host system; if so, type the same name you typed for "remote hostname" (in Figure 2-2 above).

If your computer will *not* be connected to a large central host (or if the central host is not running CEO), you should specify a local configuration to have CEO keep its Mail and Calendar databases on your system. If not, CEO won't run properly. For the local configuration, type NO↵. CEO then sets up for local operation and gives control to the AOS macro, which ends. Skip to the section "Creating AOS Profiles".

If you answered YES↵ to specify remote operation, the CEO macro asks

```
Would you like to create, delete, or modify CEO user profiles now?
YES/NO
```

A CEO profile is required for every person who will use CEO. The first time you configure CEO, you must create a profile for yourself. And you should also create a profile for every person who will use CEO. So type YES↵.

(For remote CEO, you can create, delete, or modify profiles during configuration only — it won't work properly any other time. Also notice that CEO and AOS user profiles have to be created on both local and remote systems before you can run CEO in a networked configuration.)

After a pause, CEO displays its Main Menu. Select the number for the "Utilities" choice. Then, from the Utilities menu, select "Office Manager" functions.

Proceed to create a CEO profile for each user you mentioned in Figure 2-2. The first CEO profile you create should be for yourself (type your agreed-upon User ID). Give yourself Office Manager privileges and the right to use the CLI. The "Alias", "User Directory", and "Description" questions are meaningless if you're connected to another system — they must be defined on the *central* system. Default these questions by pressing ↵.

After you finish your profile, create others for the other users (if any), with the agreed-upon User IDs. Don't give Office Manager privileges. For details on creating CEO user profiles on CEO (full product), see the section on creating a new profile in *Managing Your CEO System.*

Be sure to create all CEO profiles needed. If you forget or make a mistake while creating profiles for a remote CEO, you will need to run CONFIGURE all over again.

When you have finished all the CEO profiles needed, leave CEO: Press the CANCEL/EXIT function key (the 11th function key on the top row of keys) three times. When it asks *Do you wish to exit from CEO?*, answer **Y↵**. This stops CEO.

The CEO configure macro yields control to the AOS CONFIGURE macro; and the AOS configuration macro ends:

```
System definition is complete.....
```

Creating AOS User Profiles

Next, you should create AOS user profiles for yourself, and for every person who will use your system. (There is an OP profile supplied, but it has limitations, as described later in this chapter.)

An AOS user profile gives each person a convenient, protected place for his or her own files. It also allows the person to print files. If your system is connected to a remote system and the person has an AOS and CEO profile on both systems, he or she can use CEO remote Mail and Calendar. And, if the desktop system has multiple terminals, the profile allows the person to log on to and use any user terminal (in addition to the system console).

You can create AOS profiles with the PROFILE macro. This macro requires you to be the master CLI, PID 2, on the system console. (This prevents unauthorized people from creating profiles for themselves.)

At the system console, check your PID number with the **WHO↵** command. If not PID 2, type **BYE↵**. In this case, having just run CONFIGURE, you are PID 2, as you can see:

```
) WHO ↵
PID : 2  OP     OP  :CLI.PR (Running PID 2, the master CLI, username
                             OP. The program pathname is :CLI.PR.)
```

To run the profile macro, type

```
) PROFILE ↵
```

The macro announces itself, and gives choices:

```
User Profile Options

C Create a new user profile
D Delete a user profile
R Rename a user profile

If you want to change an old profile, delete it with D and create it with C.

Type C or D or R -- based on your preference:
```

To create a profile, type C↵.

The macro asks the following questions:

```
Please type the username -- 1 to 15 characters -- to create:
```

The username is the name by which the system — and other users — will address the person. It can be from 1 to 15 filename characters (A through Z, 0 through 9, ?, _, ., or $). Spaces aren't allowed. The name you type cannot already exist.

If you are creating an AOS profile to match a CEO profile created above, find the "User ID/username" that you filled in earlier (Figure 2-2), and type *exactly the same name* here.

The username you specify cannot already exist. The name you select can be changed — if needed — by the PROFILE Rename option.

Often, a person's first name, initials, or last name are used as the username. For example, for Chris F. Smithson, you might type CFS↵.

```
Please enter xxxx's password -- 3 to 15 characters:
```

If your system is connected to a remote system, type the user's password that was agreed upon by both systems — as written in the user's Password item in Figure 2-2. A user's password must be the same on both systems for CEO Mail and Calendar to serve that user.

If your system isn't connected to a remote system, you can make up a password. (For a single-terminal system that isn't connected to another system, the password is immaterial: type the same password as the username you give above, and press ↵.)

For a multiterminal system, you will need to tell each user his or her password — so that he or she can log on to a user terminal. Also, the password does not show (echo) as a person types it. Therefore — if you have a choice — you should use a word you can remember easily. In a multiterminal desktop system that is not connected to a remote system, an easy way to handle the password issue is to use the username as the password, and tell the user to change his or her password at logon as described in Chapter 4.

A user password can be 3 to 15 filename characters long, with no spaces. (For legal filename characters, see username detail above.) The password doesn't echo as you type it. For example, for the username CFS, you might type `CFS↵`.

```
The standard initial macro is SETUP.CLI.
Do you want to specify a different macro? {Y N} "N"
```

This answer selects the macro file that will be executed for the user after he or she logs on to a user terminal. This issue — and all remaining questions — are irrelevant for someone who is using the system console, since people can't truly log on to the system console.

If you don't want this user to start up in CEO, we recommend that you select the default file, SETUP.CLI, by pressing ↵. A user in the CLI can execute other programs (including CEO, by typing `CEO↵` or `CEO.WP↵`. To select the CLI, press ↵.

But if you want to have CEO executed for the user, type `Y↵`. It asks

```
Type the full pathname of the macro to be executed:
```

If you have the CEO Electronic Office (full product), and want it executed for the user, type `:UTIL:CEO:CEO.STARTUP.CLI↵`. If you have the CEO Word Processor - Independent, and want the Word Processor executed, type `:UTIL:CEO:CEO.WP.STARTUP.CLI↵`. This will execute CEO — full product or Word Processor — immediately at logon. If you specify a file that doesn't exist, your answer will be ignored and the default macro (:SETUP.CLI) will be used. If you really *want* to specify the CEO macro and you make a typing mistake, you will need to delete the profile (PROFILE Delete) and the user directory. Then create the profile again and type the CEO macro name correctly.

After you answer this question, the macro says

```
The standard initial program executed at logon is CLI.PR.
Do you want to specify a different program? {Y N} "N"
```

Your answer specifies the *program* that will execute when this person logs on to a user console. The CLI is a good choice — and it will execute CEO for the user if you specified CEO.STARTUP.CLI above. Generally, you should take the default — CLI — by pressing ⤶.

If you want the user to come up in a different program, type Y⤶. The macro then asks for the full pathname of the program; and you must type the full pathname, from the root (:) of the desired program. The file you specify must already exist; if not, your answer will be ignored and the default program (CLI.PR) will be used.

After you answer this question, the macro types a *Starting* message, then tells the profile editor to build the profile.

After a profile has been built, note the username and password on a piece of paper so that you can try it later. Try the username on the system console by typing **LOGON username**⤶, then type **BYE**⤶ to return to the master CLI. A summary of a PROFILE Create dialog appears in Figure 2-3.

While you're familiar with the procedure, you should create a profile for every person who will use the system. If your system is connected to another, use the username and profile for the user, as described in your Figure 2-2. As you finish each profile, add the username and password to Figure 2-2, if they are not already recorded. And try each profile with **LOGON username**⤶, then, if you see no errors, type **BYE**⤶ to return to the master CLI. If you see an error, this means that the new profile does not have the name you thought; create it again. You can discover all profile names by typing:

```
FSTAT  :UDD:+ ⤶
```

Whenever needed, from the system console, you can run the PROFILE macro again to create any other AOS profile. The PROFILE macro will run only from the master CLI, PID 2. So, if you typed **LOGON username**⤶ on the system console, you must type **BYE**⤶ to return to the master CLI.

Note that usernames are public information, but all passwords should be private.

If you make a mistake when creating an AOS profile, delete the profile and create it again. Type PROFILE⤶ and D⤶; and the macro will say

```
Delete User Profile

Please type username to delete.
```

Type the username; for example, ADRIAN. The macro says

```
Deleting ADRIAN's profile.....
The profile for ADRIAN has been deleted.

The user directory -- :UDD:ADRIAN -- has not been deleted.
You can delete it via the DELETE command if you so desire.
```

After deleting the profile you don't want, you should delete the corresponding user directory. To do this, type

```
) SUPERUSER ON ⤶
```

Then type

```
*) DELETE/V :UDD:name ⤶          (Name is the username of the directory.)

Deleted :UDD:name

*) SUPERUSER OFF ⤶
```

Then you can recreate the profile with the specifications you want.

```
) PROFILE ↵

         User Profile Options

      C Create a new user profile

      .
   Type C or D or R -- based on your preference:    C ↵

Please type the username -- 1 to 15 characters -- to create:    ADRIAN↵
Please enter ADRIAN's password -- 3 to 15 characters:    ADR↵

The standard initial macro is SETUP.CLI.
Do you want to specify a different macro  {Y N}? "N"    Y ↵

Type the full pathname of the macro to be executed:    :UTIL:CEO.STARTUP.CLI ↵

The standard initial program executed at logon is CLI.PR.
Do you want to specify a different program  {Y N}? "N"      ↵

Starting to create the profile...

ADRIAN's profile has been created.
```

DG-25831

Figure 2-2 Creating a user profile

Bringing It All UP

Now, you're ready to start up the multiuser environment via the UP macro. UP finishes the startup actions begun by the CONFIGURE macros, and starts up other DG products, like XODIAC and CEO (if you have them). You'll run UP as matter of routine, whenever you bring the system up.

If you have a printer, make sure it is ready. It should have paper available (loaded as described in the hardware *Operating* manual); and it should be on-line. If it is not on-line, press the on-line button or switch.

If you have terminals other than the system console, make sure each one is turned on and on line, with the on-line light lit. The terminal ON LINE status light glows red when the terminal is on line. If a terminal is not on line, press the CMD key and hold it down. Then, while holding CMD down, press the ON LINE key. The light should glow.

When ready, type

```
) UP ↲
```

The first message you see is about EXEC — the multiuser management program that handles the printer(s) and enables user terminals for log on.

```
Creating the EXEC process...

PID: 3
FROM PID 3 :(EXEC) READY              (Messages from EXEC...)
FROM PID 3 :(EXEC) time

Creating the XODIAC processes...      (If XODIAC is installed.)
... (XODIAC messages)...
```

If your communications line is configured to use a modem, the XODIAC UP macro tells you that it's time to dial your central DG host:

```
To use the modem line for communications, you must dial up now.
If you don't dial now, software that needs the line (like CEO) cannot
use it. Dial up or not. When ready to continue, press NEW LINE.
```

It's time to dial up the central host and make the connection. Dial the number, and when you hear the high-pitched tone, connect the modem (via the button or by inserting the receiver). When ready, press ↲.

Now the XODIAC UP macro returns control to the AOS macro, which brings up other products (if you have them).

```
Creating the INFOS process...                    (if INFOS II is in-
                                                 stalled)
... (INFOS II messages)...
Creating the CEO processes...                    (if CEO is installed)
... (CEO messages)...
FROM PID 3 : (EXEC) @PRINTER COOPERATIVE INITIATED  (If you have a printer.)
FROM PID 3 : (EXEC) ENABLED CONSOLE, @CONn       (If you have a USAM,
                                                 @CONn is a user ter-
                                                 minal.)
.
.
```

Press ↵. And, if you saw no ERROR messages, skip to the next section.

If you saw the message *NO CONSOLES ENABLED* , and you have no user terminals (just the system console), this is normal — EXEC is describing user terminals when it says *NO CONSOLES...* Skip to the next section.

If you saw messages like *FILE DOES NOT EXIST* , and *NO SUCH COOPERATIVE* , you probably made a mistake with the CONFIGURE macro. To fix it, type

```
) DOWN ↵
```

and restart the configuration sequence by typing

```
) CONFIGURE ↵
```

specify the correct answers, don't change the XODIAC or CEO answers, and try UP↵ again. If these errors recur, consult Chapter 14, the error chapter.

The CONFIGURE and UP macros, together, not only set up all your system software; they create files that allow you to use friendly names for diskette units and printers. After you run UP, all the device names are

Device	Names
Diskette unit (primary, rightmost)	@DPM0 (formal name), @DISKETTE, @DISKETTE1, or @RIGHT_DISKETTE
Diskette unit (secondary, leftmost)	@DPM1 (formal name), @DISKETTE2 or @LEFT_DISKETTE
Plotter	@PLOTTER
Printer (Model 10/SP)	@LPT (queue name), @CON1 (formal devicename), or @PRINTER
Printer (Model 20 and 30)	@LPT (queue name), or @PRINTER. The formal devicename, without a communications line, is @CON2; with a communications line, it's @CON3.
Tape unit	@MTC0 (formal name) or @TAPE

After the UP macro works without errors, you're nearly done with installation procedures. Bear with them for a little longer.

Creating CEO Profiles

If you specified a remote CEO, you've done this; skip to "Testing CEO". If you don't have CEO, skip to "Becoming a Standard User".

If you have the CEO Electronic Office (full product) and will run it locally, or if you have the CEO Word Processor - Independent, then you must create CEO profiles for every person who will use your system. For either product, you can create CEO profiles at any time. But you might as well do it now.

With the CEO Electronic Office (full product), type

```
) CEO ↵
```

and with the CEO Word Processor-Independent, type

```
) CEO.WP ↵
```

After a pause, CEO starts up and displays its Main Menu. Select the number for the "Utilities" choice. Then, from the Utilities menu, select "Office Manager" functions. Proceed to create a CEO user profile for each person who will use CEO.

The first CEO profile you create should be for yourself. As a "User ID," repeat your AOS username. Give yourself Office Manager privileges and the right to use the CLI. If asked, the "path to personal drawer" should be **:UDD:username:CEO_DRAWER** — make this a CPD directory, of 5000 (blocks) or more. The "Alias", "User Directory", "Description", and other questions are explained in the CEO *Managing* books. For the *full* CEO product, see the section on creating user profiles in *Managing Your CEO® System*. For details on CEO Word Processing - Independent, see the section on creating user profiles in *Managing CEO® Word Processing - Independent.*

After you finish your profile, create profiles for the other users (if any), with the same user ID as the AOS username. Don't give Office Manager privileges. When you have finished all the CEO profiles, leave CEO: Press the CANCEL/EXIT function key (the eleventh function key on the top row of keys) four times. When it asks ***Do you wish to exit from CEO?***, answer Y↵. This stops CEO and returns you to the CLI.

Becoming a Standard User

Thus far, you have been running a process with the username OP, and a process ID (PID) of 2 — known as the master CLI — on the system console. This CLI has super powers, and can be used to read or delete anyone's files or do other malicious things. Also, a process with the username of OP has certain limitations: it cannot use a CEO that runs on a remote system, for example, or log on to a remote system.

The next step is to become a standard user, with a standard CLI, under your own username. To do it, use the profile you created for yourself above by typing

```
) LOGON  username↵          (Where username is the AOS username;
                             for example, LOGON ADRIAN↵.)

AOS CLI REV n date time
)
```

The LOGON macro creates a standard CLI process with a username other than OP (needed for CEO and communications). And it puts you in your user directory for normal operations. The standard CLI process also adds some security to the system (covered below in "The System Console and Security").

You should type **LOGON username↵** every time you bring the system up (UP macro).

Testing CEO

Next, if you have CEO, try starting it up under your own username. (If you don't have CEO, skip this section.)

If you have the CEO Electronic Office (full product), running remotely, phone the remote central system and make sure matching profiles have been created there. And make sure the remote system is up (not down for maintenance, or some other reason). Then, type

```
) CEO ↵
```

With CEO Electronic Office (full product) running locally, you don't need to phone. Just type

```
) CEO ↵
```

For the CEO Word Processor - Independent, type

```
) CEO.WP ↵
```

After a pause, the CEO Main Menu should appear on your screen. If the bottom line on the screen says: *If you need assistance here (or on any other menu or question), press the HELP key* , this means that CEO is running normally. Congratulations! To help use CEO, get a CEO template (a plastic strip that names the CEO function keys, supplied with the CEO product). Fit the template over the function keys: the topmost row of keys. To leave CEO, press the CANCEL/EXIT key, then confirm by typing **Y↵**.

If the screen says *You do not have a CEO profile, contact your CEO manager to obtain one* , this means there isn't a CEO profile for the username you typed in LOGON above. Perhaps you made a mistake when you created CEO profiles. In any case, you will need to create another CEO profile. Type **BYE↵** to return to the master CLI with username OP. Then, if you have either the full CEO running locally or Word Processor - Independent, create the profile ("Creating CEO Profiles", above); then continue from there (**LOGON** username↵).

If you have the full CEO product running remotely, you must run CONFIGURE again. Don't change the AOS or XODIAC specs: all you need do is create a CEO profile. Before starting, you may want to phone the central site to verify usernames.

If the bottom line on the screen says *The Post Office is Closed* , this means CEO could not access its Mail database. The cause may be no connection (you may have misdialed for the modem connection), or that the remote system is not up and running CEO. Check with the remote system; if it is not up, you won't be able to use Mail and Calendar until it is up. If the remote *is* up, type **DOWN↲**, then **UP↲** and redial the modem line.

If the bottom line is *Cannot find your Inbox. See your CEO manager,* this means that your profiles on the remote system are wrong or missing. Check with the remote system to make sure that matching AOS and CEO profiles have been created (as in Figure 2-2). After making sure that all is okay on the remote system, retry CEO (type **CEO↲**).

The System Console and Security

The system console communicates directly with the computer, while user terminals (if you have any) are managed (through the USAM multiplexor hardware) by EXEC. EXEC allows only people who type a username and password to use a user terminal. Thus, user terminals are relatively safe from unauthorized people.

But the system console is always available to anyone when AOS is up, and it doesn't enforce a log-on procedure. The system console has access to the master CLI, which has super powers: it can shut down the system, delete files indiscrimately, or create user profiles.

So, even if you are the only person who will use the system console, you should run the LOGON macro. It allows you to run CEO and do most standard operations, except things like

- create AOS profiles;
- install software;
- back up the entire hard disk;
- delete other users' files; and
- shut the system down.

When the standard CLI created by the LOGON macro terminates (as when someone types **BYE↲** to it), the master CLI returns. You can then use the master to shut down the system or to do other privileged operations like create profiles, or back up the disk.

Trying the Printer

If you have a printer, try a QPRINT command:

```
) QPRINT UP.CLI ↲
QUEUED, SEQ = 1, QPRI=127
)
```

The printer should now print the text of the system UP macro. (If not, conditions may be wrong or there may be another problem — see the hardware *Operating* manual.)

You needn't examine the printed file now — it served only to test the printer and printer queue. When you print a file, you don't need to wait for the printer to finish — AOS can run your terminal and the printer at the same time.

Logging On as a User

If your system has only one terminal (the system console), skip this section.

To log on to a user terminal, a person needs a user profile — created by the PROFILE macro and user profile editor. For this log-on example, use the username and password you gave yourself earlier with PROFILE (shown in example as username **ADRIAN** and password **ADR**).

To log on, go to the nearest terminal that isn't the system console. The user terminal screen should show

```
*** AOS REV n  / TYPE NEW-LINE TO START LOGGING ON ***
```

Press ↲; then answer the USERNAME/PASSWORD questions as follows:

On a user terminal:

```
AOS EXEC REV n date time
```

USERNAME:	username ↲	(Type your username and press ↲.)
PASSWORD:	password ↲	(Type your password and press ↲. For privacy, the password doesn't show on the screen.)

```
AOS CLI  REV n  date  time
)
```

If you make a mistake, the system console displays *INVALID USERNAME/PASSWORD PAIR.* Try again.

You have now logged on as a user and a CLI process is running on your terminal. This is your own CLI, independent of the CLI on the system console. Your user directory is the current (working) directory. Its pathname is :UDD:username and it's a good place to keep all your computer-based work.

Terminate the CLI process by typing

```
) BYE ↵

AOS CLI TERMINATING  date  time

*** AOS REV n  / TYPE NEW LINE TO BEGIN LOGGING ON ***
```

Next, try all the usernames and passwords you created with PROFILE above. If one doesn't work, return to the system console, type BYE↵ to access the master CLI; then rerun PROFILE to delete and recreate the profile before the user ever sees it. When done, type LOGON username↵, at the system console, and return to a user terminal to try the profile.

Whenever anyone logs on to a user terminal (using username and password), the user directory becomes the person's current (working) directory; all files he or she creates are created here.

Now you know that EXEC's log-on function works, and that your terminal lines are properly connected.

The log-on procedure is the same for any user, on any user terminal: type ↵, then username↵, then password↵. To log off, type BYE↵.

About the Operator Profile

The operator profile shipped with AOS has the username of OP and password of ABC — this is a general-purpose profile whose primary use is that it allows you to print files from the master CLI.

Generally, any process with the username of OP cannot use CEO Mail or Calendar (if these are on a remote host connected to the desktop system). Nor can such a process log on to a remote system. These restrictions limit the usefulness of the OP profile.

However, if your system is not connected to another DG system, and you want to use the OP profile as a personal profile, you can do so. In a multiterminal system, you may want to change the OP password to protect your files; you can do this from a user terminal as described near the end of Chapter 4. Also, each time you bring up the system, or finish some privileged operation, you should type LOGON OP↵ to safeguard the system console (just as if you had another username).

Shutting Down the Multiuser Environment and System

You must shut the multiuser environment down before shutting AOS down. Do this from the system console, using the DOWN macro. The DOWN macro brings down CEO (if up), and other software; then it terminates EXEC, which terminates all user processes. Type

```
) DOWN ↵

Error - Only the master CLI can run DOWN.
)
```

To execute DOWN, you must be running the master CLI, PID 2. Since you ran LOGON, this is not true. Check again with the WHO command:

```
) WHO ↵

PID: 18  OP  018  :CLI.PR
```

To return to PID 2, the parent, as before, type

```
) BYE ↵

AOS CLI TERMINATING  date  time

You are now
PID: 2  OP  OP  :CLI.PR
```

When you sign off the standard CLI on the system console, it displays the PID (2) of the master CLI. This is a convenience — it tells you that the master is running (as you can verify with the WHO command):

```
) WHO ↵

PID: 2  OP  OP  :CLI.PR
```

From PID 2, DOWN will work. Try it:

```
) DOWN ↵

Bringing down CEO...
```

... (CEO messages)...

```
Bringing down INFOS...
```

... (INFOS II messages)...

```
Bringing down XODIAC...
```

... (XODIAC messages)...

```
Bringing down EXEC...

DOWN processing complete at  time  on  date
.
)
```

You can now shut AOS down. You do this by typing BYE↵ to the master CLI:

```
) BYE ↵

DO YOU REALLY WANT TO SHUT THE SYSTEM DOWN?
```

Confirm the shutdown by typing

```
Y ↵

STARTING SYSTEM SHUTDOWN

SYSTEM SHUTDOWN

n
!
```

Congratulations! You have installed AOS, all support software, configured them, created profiles, and tested them. AOS and its multiuser environment are complete on the hard disk.

You can bring AOS up again by typing **26H**, the date, the time, and **N↵**; and you can bring the multiuser environment up again by typing **UP↵**, as explained in Chapter 3. Using this environment is much easier than creating it.

WARNING *Orderly shutdown is very important. Never turn computer power off when AOS is running. Turning power off while AOS is up produces an abnormal (incomplete) shutdown. System users may lose a lot of work. And you will have to run a disk fixing program (FIXUP) before you can restart AOS.*

Step Summary

To summarize this chapter: Figure 2-4 briefly reviews all the steps you've taken — from turning on the computer, to creating the last profile, to shutting down.

Getting Ready

1. If the system console is off, turn it on. It may beep. Turn the computer on, if off. If nothing shows on the screen, check the brightness (contrast).

2. Have the DG-supplied diskettes handy, in order. For certain Model 10/SP systems (for example, a 10/SP with a German keyboard), 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. For most systems, use the first AOS diskette: AOS PREGEN, DISKETTE 1 OF 4.

3. Insert the first diskette. With two slots, use the rightmost.

4. If you have a diskette labeled "BOOTABLE D200 EMULATOR", program load from it: next to the *!* prompt, type 20H. Then, wait for the *!* prompt, remove the diskette, and insert the first AOS diskette.

Formatting the Hard Disk

5. *!* `20H` (Type 20H, wait 20 seconds or so.)

6. *PROGRAM NAME?* `DFMTR ↵` (Type DFMTR↵, wait 20 seconds or so.)

   AOS DISK FORMATTER (DFMTR) REV x.xx

   DO YOU WANT TO SAVE ANY INFORMATION ON YOUR DISK ?
   TYPE Y (YES) TO SAVE INFORMATION, OR

7. *N (NO) TO ERASE YOUR DISK.* `N ↵`

   THIS WILL ERASE ALL FILES FROM YOUR DISK.
   TYPE Y (YES) IF YOU ARE SURE YOU WANT TO DO THIS, OR

8. *TYPE ANYTHING ELSE TO START OVER.* `Y ↵`

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (continues)

```
 9. HOW MANY HARD DISKS DO YOU HAVE?   1↲          (or, with two disks, 2↲)

    ANALYZING YOUR DISK FOR BAD BLOCKS               (Operations decrease to
    NUMBER OF OPERATIONS REMAINING: n                0. It  takes 40-50 minutes per
   .                                                 15-Mbyte disk and
   DONE!                                             more than double that
   .                                                 for the 38.6-Mbyte
   !                                                 disk.)
```

Installing AOS on the Hard Disk

```
10. !    20H              (Wait 20 seconds.)

   WELCOME TO THE DG DESKTOP GENERATION

11. PROGRAM NAME?   INSTL↲       (Wait 20 seconds.)

   AOS INSTALLER (INSTL) REV x.xx
   -- INITIALIZING THE DISK          (Wait 30 seconds.)

   -- STARTING SYSTEM INSTALLATION   (Wait 2-3 minutes.)

   -- SYSTEM INSTALLED

   DONE!
   .

12. !    26H

   WELCOME TO THE DG DESKTOP GENERATION

13. PROGRAM NAME?    AOS ↲

   INITIALIZING SYSTEM DATA ON DISK --   (Wait a minute or so.)

   INITIALIZATION COMPLETE

   AOS REV xx.xx

14. DATE (MM/DD/YY)?    11  16  84 ↲          (Current date.)

15. TIME (HH:MM:SS)?    14:30 ↲  (Current time, using 24-hour clock)

16. DO YOU WANT TO INSTALL AOS SYSTEM SOFTWARE [N] ?    Y ↲

   MOUNT AOS DISKETTE 2 -- TYPE NEW-LINE WHEN READY
```

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (continues)

17. Change diskettes: remove AOS diskette 1 and insert AOS diskette 2.

18. ↵

19. Wait 1 to 2 minutes.

```
AOS  CLI  REV  n  date  time
)
```

20. Change diskettes: remove AOS diskette 2 and insert AOS diskette 3.

21. `) INSTALL  AOS ↵`

```
Please insert the diskette with the desired DG-supplied AOS...
```

22. *Press NEW-LINE when you are ready...* ↵

```
Beginning AOS software installation...
```

. (Filenames copied. Each diskette takes about 2 minutes.)

```
The disk is exhausted. Mount disk 2 and press NEW-LINE...
```

23. Remove the diskette and insert the next AOS diskette (if any); press ↵.

24. Repeat step 22 and 23;

. (filenames)

.

```
AOS software installation is complete. Please ...
```

25. For Model 10/SP systems *only:* Remove diskette, insert diskette labeled "D200 EMULATOR"; and type

`) INSTALL  EMULATOR ↵`

- *Press NEW LINE when you are ready...* ↵
- * Answer the questions about monochrome (as opposed to color), and your language.

. (installs emulator file)

```
EMULATOR software installation is complete...
```

Remove the diskette from the unit.

If you have a Model 20 or 30 system, you are done!

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (continued)

26. You've installed AOS on your hard disk(s).

27. Now, for Model 10/SP systems only: shut down AOS, turn the power off, and restart to load the emulator. Type

```
27a. ) BYE ↵

27b. DO YOU REALLY WANT TO SHUT ... DOWN?     Y ↵
     SYSTEM SHUTDOWN
     !
```

27c. Turn computer unit power off and on again.

```
27d. ! 26H          (Type 26H)
     WELCOME TO THE DG DESKTOP GENERATION

27e. PROGRAM NAME?      AOS ↵     (Type AOS ↵.)
     AOS REV n

27f. DATE (MM/DD/YY)? 12 14 84 ↵ (Type the correct date.)
```

NOTE *With a French or Italian terminal, use the numeric keypad on the far right of the keyboard to answer the DATE and TIME questions. The normal numbers will not work.*

```
27g. TIME (HH:MM:SS) 15 20 ↵ (Type the correct time, 24-hour clock.)

27h. DO YOU WANT TO INSTALL AOS ...SOFTWARE[N] ↵
     AOS CLI REV n date time
     )
```

To test for the correct emulator, type a lowercase letter. If you can make a lowercase letter appear, an emulator is loaded and working correctly. Continue to the next section. If a lowercase letter doesn't appear, return to step 25.

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (continued)

Installing XODIAC (skip if you don't have XODIAC)

- Get the XODIAC diskette and insert in primary unit.
- `) INSTALL  XODIAC ↵`
- *Press NEW-LINE when you are ready...* ↵

  Beginning software installation...

  .
  . (copies to hard disk)
  .

  Xodiac software installation is complete. Please remove diskette...
- Remove the XODIAC diskette.

Installing INFOS II

- Get the INFOS II diskette and insert it in the primary unit.
- `) INSTALL  INFOS ↵`
- Press NEWLINE when you are ready... ↵

  .
  . (copies to hard disk)
  .

  INFOS software installation is complete. Please remove diskette...
- Remove the INFOS diskette.

Installing Sort/Merge

- Get the SORT diskette and insert it in primary unit.
- `) INSTALL  SORT ↵`
- *Press NEW-LINE when you are ready...* ↵

  .
  . (copies Sort to hard disk)
  .

  SORT software installation is complete. Please remove diskette..
- Remove the last Sort/Merge diskette.
- `) SPACE  : ↵`   (Note "After AOS, INFOS, and Sort" and the figures on paper.)

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (continued)

Installing CEO (skip if you don't have CEO)

- Get CEO diskettes and insert the first diskette in the primary unit.
- `) INSTALL CEO ↵`
  or
  `) INSTALL CEO.WP ↵`
  .
- *Press NEWLINE when you are ready to begin installation.* ↵

  The disk is exhausted. Mount disk n+1 and press NEWLINE...
- Remove the diskette from the unit, insert next diskette, press ↵. Repeat until you see

  CEO...software installation is complete. Please remove diskette from the primary unit.
- `) SPACE : ↵` (And note "After CEO" and figures on paper.)
- Install other products (like GLDP, described earlier, and others, named in Chapter 5, INSTALL command) as desired.

Configuration Dialog

- If you will use CEO remotely or communicate with another system, call the remote system and get information to fill any pertinent blanks in the list (given in Figure 2-2).
- `) CONFIGURE ↵` (Starts AOS configuration)
- If your system has ever been configured, it asks if you want to skip AOS configuration. The rest of the AOS configuration dialog goes

  *** Defining System Configuration ***
- *Please type...model ...? {10SP 20 30} "10SP"* ↵ (or 20↵, or 30↵)

  For a Model 20 or 30, it asks
- *Is the system console...D410-D460 compatible? {Y N} "N"* ↵ (or Y↵)
- *How many diskette units do you have? {1 2} "1"* ↵ (or 2↵).

  Do you want to reserve line 0 for communications...? {Y N} "N"

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (continued)

- To set up for communications with a remote system, type Y↵. Otherwise press ↵ and skip the next two questions.

  Is line 0 is connected via a modem? {Y N} "N"

- Based on your Figure 2-2, answer Y↵ (for modem) or ↵ (for direct).

  What is the baud rate....?{300 1200 2400 4800 9600} "n"

- Based on your Figure 2-2, answer ↵ (for default) or the other rate.

- *How many printers do you have? {0 1} "0"* ↵ (or 1↵)

  If you have a printer:

- *What is the maximum number of characters you want printed per line? {16 - 255} "80"* 160 ↵

- *What is the maximum number of lines that can be printed per page? {6 - 144} "66"* ↵

- *How many plotters do you have? {0 1} "0"* ↵ (or 1↵)

- * *How many user terminals ...? {0 - m} "0"* ↵ (or 1↵ or 2↵ and so on)

  And for every user terminal, it asks

- *Is the terminal...D410-D460 compatible? {Y or N} "N"* ↵ (or Y↵ for D410-D460.)

- *Is line n connected via a modem? {Y, N} "N"* ↵ (or Y↵ if this terminal will be remote.)

- *What...baud rate...? {300 1200 2400 4800 9600} "n"* ↵ (or, for a modem, ↵ or 300↵.)

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (continued)

XODIAC Configuration Dialog (skip if you don't have XODIAC)

If XODIAC has ever been configured, it gives you the option to skip. The XODIAC dialog goes

Please specify your system's local hostname

- Type your desktop system's hostname, from your Figure 2-2.

  Please specify your system's local ID

- Type your desktop system's local ID, from your Figure 2-2.

  Please specify the remote system's hostname

- Type the remote (distant) system's hostname, from your Figure 2-2.

CEO Configuration Dialog (skip if you don't have CEO)

If CEO has been configured, it gives you the option to skip. The CEO dialog goes

Does your Desktop System have a letter-quality printer?
YES/NO

- Type YES↵ if you have a model 4518 letter-quality printer, or NO↵ if you don't have one.

  The CEO Word Processor macro ends. The full product CEO macro explains about configurations and asks

  On a small system CEO can have two possible configurations.

  Would you like to use CEO in the remote configuration?
  YES/NO

- To run CEO with its Mail and Calendar remote, on a central DG system, type YES ↵. Remote operation should be mentioned on your Figure 2-2. To have everything local, type NO ↵.

  For remote operation, the CEO macro asks the following questions. For local operation, it ends: skip to "Creating AOS User Profiles".

  Would you like to create, delete, or modify CEO user profiles now?
  YES/NO

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (continued)

For remote CEO, you can create, delete, or modify profiles during configuration only — it won't work properly any other time.

- To create one or more CEO profiles (required the first time), or to delete or change a CEO profile, type `YES ↵`.

  (If you type `NO ↵`, it ends: skip to the next section.)

- From the CEO Main Menu, select "Utilities", then "Office Manager".

- Starting with yourself, create profiles for everyone on your system who will use CEO. Get the User IDs from your Figure 2-2. The "Alias", "User Directory", and "Description" must be defined on the *central* system. Default them by pressing ↵.

- Leave CEO: Press the CANCEL/EXIT function key (the eleventh function key on the top row) four times. When it asks *Do you wish to exit...?,* type `Y↵`.

  System configuration is complete.....

Creating AOS User Profiles

Next, create AOS user profiles for yourself, and for every person who will use your system:

- `) PROFILE↵`

  User Profile Options

  C Create a new user profile
  D Delete a user profile

- *Type C or D or R -- based on your preference:* `C ↵`

- *Please type the username ... to create:* `xxxx ↵` (Type the username, same as CEO User ID, if any.)

- *Please enter xxxx's password ...* `yyy ↵` (Type the password, same as the remote password, if any.)

  The standard initial macro is SETUP.CLI.
  Do you want to specify a different macro ? ... "N"

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (continued)

- To have the user start up in CEO, type `Y↵`; otherwise, answer ↵ and skip the next question.

```
Type the full pathname of the macro..
```

- For the full product CEO, type `:UTIL:CEO:CEO.STARTUP.CLI↵`. For the Word Processor - Independent, type `:UTIL:CEO:CEO.WP.STARTUP.CLI↵`.

- *The standard initial program executed at logon is CLI.PR.*
  Do you want to specify a different program ? ..."N" ↵

```
Starting to create the profile...
xxxx's profile has been created.
```

- Create an AOS profile for everyone who you know will use the system.

Bringing It All UP

- Turn the printer on, and have all user terminals on and on line (if your system has these).

- `) UP ↵`

```
Creating the EXEC process...
```

Creating the XODIAC processes... (If XODIAC is installed.)

... (XODIAC messages)...

If your communications line is on a modem, the XODIAC macro asks

```
To use the modem line for communications, you must dial up now.
.
... When ready to continue, press NEW LINE.
```

- Dial the central host and connect the modem (if asked). Press ↵.

Creating the INFOS process... (if INFOS II is installed)

... (INFOS II message)...

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (continued)

Creating the CEO processes... (if CEO is installed)

... (CEO messages)...

... (EXEC messages) ...

.

- Press ↵. With no ERROR messages, skip to the next section. With error messages, rerun CONFIGURE: type DOWN↵ and CONFIGURE↵. Then give the correct AOS answers (don't change the XODIAC or CEO answers), and try UP↵ again.

Creating CEO Profiles (skip if you don't have CEO)

- With a remote CEO (Mail and Calendar on another system), skip to the next section.
- With a local CEO or CEO Word Processing - Independent, create profiles. Type

) CEO ↵ (or, with the Word Processor, type CEO.WP ↵.)
- From CEO Main Menu, choose the "Utilities", then "Office Manager" functions. Starting with yourself, create a CEO user profile for each person who will use CEO.
- As a "User ID," repeat the AOS username. Give yourself (but not other users) Office Manager privileges and the right to use the CLI. If asked, the "path to personal drawer" should be :UDD:username:CEO_DRAWER — this should be a CPD directory of 5000 (blocks) or more. The "Alias", "User Directory", "Description", and other questions are explained in the CEO *Managing* book.
- Leave CEO: Press the CANCEL/EXIT function key (the eleventh key on the top row). When it asks *Do you wish to exit...* , type Y↵.

Becoming a Standard User

-) LOGON username ↵ (Where username is your AOS username; for example, LOGON ADRIAN↵.)

```
AOS CLI  REV  n  date  time
)
```

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (continued)

Testing CEO (skip if you don't have CEO)

-
  ```
  ) CEO ↵              (For the full CEO product)
  ```

  or

  ```
  ) CEO.WP ↵           (For the Word Processor - Independent)
  ```

  The CEO Main Menu appears on your screen. If the bottom line says *If you need assistance here (or on any other menu or question), press the HELP key* , this means success! Congratulations. Find and fit a CEO template.

  If the Main Menu doesn't come up, or the bottom line has an error message, find the message in Chapter 16, the error chapter.

Trying the Printer (skip if you don't have a printer)

-
  ```
  ) QPRINT  UP.CLI ↵
  QUEUED, SEQ = 1, QPRI=127
  )
  ```

  If the printer doesn't print, check for paper, and so on (in the *Operating* manual).

Logging on as a User

- With only one terminal (the system console), skip this section.
- Go to the nearest terminal that isn't the system console. The screen shows

  ```
  *** AOS REV  n  / TYPE NEW-LINE TO START LOGGING ON ***
  ```

- Press ↵.
- *USERNAME:* `username ↵` (Type your username and press ↵.)
- *PASSWORD:* `password ↵` (Type your password and press ↵. The password doesn't show.)

  ```
  AOS CLI  REV  n  date  time
  )
  ```

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (continued)

If you made a mistake, it says *INVALID USERNAME....* Retry.

-
  ```
  ) BYE ↵
  ```

  ```
  AOS CLI TERMINATING...
  ```

- Check all the usernames and passwords you created with PROFILE above. If one doesn't work, return to the system console, type BYE↵ to access the master CLI; then rerun PROFILE to delete and recreate the profile.

Shutting Down the Multiuser Environment and System

- At system console, type

  ```
  ) DOWN ↵
  ```

  ```
  Error - Only the master CLI can run DOWN.
  )
  ```

-
  ```
  ) WHO ↵
  PID: 18  OP  018  :CLI.PR
  ```

-
  ```
  ) BYE ↵              (Return to the master CLI.)
  ```

  ```
  AOS CLI TERMINATING  date  time

  You are now
  PID: 2  OP  OP  :CLI.PR
  ```

-
  ```
  ) DOWN ↵
  ```

  ```
  Bringing down CEO...
  ```

  ... (CEO messages)...

  ```
  Bringing down INFOS...
  ```

  ... (INFOS II messages)...

  ```
  Bringing down XODIAC...
  ```

  ... (XODIAC messages)...

  ```
  Bringing down EXEC...

  DOWN processing complete...
  ```

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (continued)

```
•   ↵
    )

•   ) BYE ↵

•   DO YOU REALLY WANT TO SHUT THE SYSTEM DOWN?   Y ↵
    .
    SYSTEM SHUTDOWN
    .
    !
```

Congratulations! You have installed AOS, all support software, and configured all support software that needs configuration.

DG-25832

Figure 2-4 Step summary, installing AOS and system software, and configuring system software, on a blank hard disk (concluded)

What Next?

If you want to stop for a while, fine. Later, you can bring the system back up as described in the next chapter.

If this is your first system, you may want to check startup/shutdown steps, described in the next chapter; or you may want to try the CLI session described in Chapter 4; or get some system background (Chapter 5); or try CEO (Chapter 6).

If you are rebuilding/restoring a system, the next steps are to restore information from backup diskettes (Chapter 7).

Starting Up and Shutting Down 3

Read this chapter when

- you want to start up your computer or AOS system;
- you want to shut down your AOS system or computer.

This chapter gives the details on system startup and normal shutdown. (For abnormal shutdown, see Chapter 14, the error chapter.) The major sections are

- The Break Sequence
- Startup
- Normal Shutdown
- What About Abnormal Shutdown?
- What Next?

The Break Sequence

The break sequence stops the computer, freezing all system activity, and gives control of the system console to a loader program (*!* prompt.) Generally, you will want to avoid typing the break sequence unless AOS is deadlocked (frozen, with no response at any terminal).

The break sequence is the CMD and BREAK/ESC keys. If someone types the break sequence at the system console, the computer halts and displays a *!* prompt. AOS and all its processes — if running at the break — are suspended. To have the computer proceed, type `P`. The P command tells the computer hardware to proceed, and system activity resumes.

Accidental break sequences can be disconcerting and annoying. But they are not disastrous if you know how to handle them.

Startup

Startup includes turning power on (if off), starting up an AOS operating system, and bringing up the multiuser environment. It assumes that AOS was shut down normally, not by a fatal error ***FATAL AOS ERROR*** or power failure. If a fatal error or power failure occurs, see Chapter 14, the error chapter.

On Model 10/SP computers, when you turn power on, the system console has limited capability. For example, it can display only uppercase letters and the DEL key does not delete characters from the screen. (DEL does remove the characters — but it displays an underscore for each character removed instead of erasing it.) To give the system console full capability, AOS loads an emulator program from the hard disk when AOS starts up. This means that, on a Model 10/SP computer, the system console cannot do certain things until AOS is up. If the system console remains limited — for example, it can't produce lowercase letters — after AOS is up, then the needed emulator program file may be missing from the hard disk. The program pathname (for English language system consoles) is :MD200.TX for monochrome (standard) monitors and :CD200.TX for color monitors. You can check for this file from AOS by typing `F/AS□:MD200.TX□:CD200.TX↵`. (Each box represents a space). If the appropriate filename does not show up, you must install the emulator program on the hard disk as described in Chapter 2, step 25.

To start up AOS and the multiuser environment, follow these steps:

1. If the system console is off, turn it on. Use the switch behind the screen, on the right.

   With a Model 20 or 30, the ON LINE status light, above the keyboard, should glow. If it doesn't glow, press the CMD key, hold it down, and press the ON LINE key. This puts the terminal on line; the light should glow. The system console should show a self-test message and may beep.

   Turn on power to the cartridge tape module (if present) and the second hard disk unit (if present).

2. Turn on power to the computer unit. Use the main power switch on the upper right of the computer module. On a Model 10/SP, the system console may display a *SELECT LOAD DEVICE...* message. Then — on any system — the system console should display the *!* prompt. Wait 15 seconds for the hard disk to become ready.

   (If the screen displays nothing, check the brightness (contrast) control under the right front corner of the screen unit; set this control in a central position. If this doesn't help, make sure that the unit is plugged in and that connections are secure. Turn power off and on and wait 15 seconds. If this doesn't help, consult Chapter 14, the error chapter.)

3. Type **26H** next to the *!* prompt:

   ```
   !   26H
   ```

   If you are using a European keyboard and you get a question mark next to the prompt (*?!*), use the numeric keypad on the right to type the numbers.

4. This loads a *bootstrap* program into the computer. The bootstrap program displays

   ```
   WELCOME TO THE DG DESKTOP GENERATION

   PROGRAM NAME?
   ```

   (If nothing happens, you may have made a typing error. Type the break sequence — CMD and BREAK/ESC keys — and return to step 3.)

   Whenever it asks *PROGRAM NAME?* , you can specify AOS by typing **AOS↵**. Do it:

```
AOS ↲
```

(If you see an *ABORT* error message, don't worry. You may have made a mistake typing the name AOS. Return to step 3 and try again — type **AOS** and press ↲.)

5. Depending on the last program you ran (for example, AOS or FIXUP) loading the system takes about 10 seconds, or more than a minute. Be patient. Soon, you'll see

```
AOS REV n

DATE (MM/DD/YY)?
```

It's very important to enter the correct date (and time). Every file that you (and other users) create while the system is up will have a creation date/time based on the date and time you enter. A file's creation date is a useful and significant piece of information.

If you type the wrong date or time here, continue until you see the CLI) prompt. Then correct the date or time using the DATE or TIME command, described in Chapter 5. The system date or time can be set only from the master CLI process, PID 2 (the first CLI you see).

Type the date as numbers for month, day, and year. Insert a space or slash before the day and year. For example, for December 14, 1984, type

```
12  14  84 ↲

TIME (HH:MM:SS)?
```

6. Type the time, based on a 24-hour clock, in hours, minutes, and seconds. (Minutes and seconds are optional. If you omit them, the system uses 00:00.) Insert a space or colon before the minute and second numbers. For example, for 2:30 p.m., type

```
14  30 ↲

DO YOU WANT TO INSTALL AOS SYSTEM SOFTWARE [N] ?
```

7. Installing AOS software is described in Chapters 2 and 8. For normal startup, you don't want to install it. So say no by pressing

)

   A pause occurs here; then

   ```
   AOS CLI  REV n  date  time
   )
   ```

   The master CLI, process ID (PID) 2, is running. You might want to adjust screen brightness for comfort here, using the control under the panel.

8. If your system has a printer, make sure it is on and on line, with paper aligned. If the on-line light is not lit, press the on-line key.

9. If your system has one or more user terminals, and people will be logging on to them, make sure each one is turned on and is on line (use the CMD and ON LINE keys).

10. The next step is to bring the multiuser environment up via the UP macro. UP starts needed printer queues; and it also tries to execute macros that bring up other software, like XODIAC, INFOS II, and CEO. Type

```
) UP ↵
PID: 3
.
```

11. Now, if your system has a communications line configured to operate over a modem, the XODIAC UP macro tells you that it's time to dial your central host. (If there is no line configured over a modem, it skips this step.)

```
To use the modem line for communications, you must dial up now.
If you don't dial now, software that needs the line (like CEO) cannot
use it. Dial up or not. When ready to continue, press NEW LINE.
```

Generally, your system needs the modem connection. To make it, dial the remote system number and connect the modem. If you don't dial up, there will be no connection to the remote system. You will not be able to log on to the remote system (Chapter 12). And, if you have CEO with Mail and Calendar on the remote system, the Post Office and Calendar will not be available.

In any case, after you've dialed the remote system or decided not to dial, press ↵. (If you make a mistake dialing, there will be no connection; and CEO will display a *Post Office is Closed* message when someone executes it. See the message Chapter 14 to recover.)

12. The AOS UP macro tries to execute INFOS and CEO UP macros:

    ...(messages from INFOS or CEO processes, if any)...

    ...(other software messages)...

    ```
    ↵           (press ↵)
    )
    ```

    The CLI prompt means that EXEC and other programs that support the multiuser environment are running.

    You can use the printer and people can log on to user terminals (if there are any). All CLI commands and supplied macros are available to you: for example, `WHOS↵` to display the processes that are running, and `HELP↵` to get help.

13. The CLI running on the system console is the master CLI, PID 2. This CLI has super powers: it can shut the system down, turn Superuser on and read anyone's files, and so on. This CLI also has the username OP — which means that you may not be able to run CEO from it.

    So, for security and to be able to use CEO, run the LOGON macro, format `LOGON username`, where `username` is your username (as given to the PROFILE macro). The LOGON macro runs a standard CLI, with your username, under the master CLI. For you to communicate with the remote system, a profile with your desktop username and password must also exist on the remote system. Generally, the username cannot be OP (for several reasons, one of which is security on the central host). For example, if your username is SALLY, type

    ```
    ) LOGON  SALLY ↵

    AOS CLI  REV n  date  time
    )
    ```

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 ALPHA LOCK once and type Q again. If the screen still displays Q, the terminal emultor isn't working; read the paragraph that precedes the numbered steps above.

Figure 3-1, next, summarizes startup.

1. If power to the system console is off, turn it on. The ON LINE status light should glow. Also turn on power to the cartridge tape module (if present) and the second hard disk (if present).

2. If computer power is off, turn it on with the main power switch. Wait 15 seconds for the disk to become ready.

3. `! 26H` (Type 26H)

   WELCOME TO THE DG DESKTOP GENERATION

4. *PROGRAM NAME?* `AOS ↵`

   (Depending on the last program run, wait either 10 seconds, or a minute or so.)

   AOS REV n

5. *DATE (MM/DD/YY)* `12 14 84 ↵` (Type the current date.)

6. *TIME (HH:MM:SS)* `14 30 ↵` (Type current time, 24-hour clock.)

7. *DO YOU WANT TO INSTALL AOS SOFTWARE [N] ?* ↵ (Say no by pressing ↵.)

   ```
   AOS  CLI  REV  n  date  time
   )
   ```

8. If your system has a printer, make sure it is turned on and is on line.

9. If your system has multiple terminals, make sure each one is turned on and is on line.

10. `) UP ↵`

    PID: 3

    .

11. With a communications line configured to run via a modem, it asks

    To use the modem line for communications, you must dial up now.
    If you don't dial now, software that needs the line (like CEO) cannot use it. Dial up or not. When ready to continue, press NEW LINE.

    Dial up and press ↵.

DG-25835

Figure 3-1 System startup summary (continues)

12. The AOS UP macro tries to execute other other UP macros:

    ...(messages from XODIAC, INFOS or CEO proceses, if any)...

    ```
    ...

    ↲
    )
    ```

13. `) LOGON username ↲` (for example, `LOGON SALLY↲`.)

    ```
    AOS CLI REV n date time
    )
    ```

DG-25835

Figure 3-1 System startup summary (concluded)

Normal Shutdown

Normal shutdown means orderly shutdown from an active AOS system to the *!* prompt; and, optionally, to turning off computer power.

These shutdown steps assume that the multiuser environment is running and that other people *may* be logged on to the system. If you know that no one else is logged on, start with step 3.

WARNING *It's very important to shut AOS down normally. Never turn computer power off when AOS is running. Turning power off while AOS is up produces an abnormal (incomplete) shutdown. System users may lose a lot of work. And you will have to run a disk fixing program named FIXUP before you can restart AOS.*

1. Work from the system console. Check the multiuser environment with the WHOS macro, to see if any users are logged on when you bring the system down. If anyone is logged on, (especially in :UTIL:CEO_DIR:CEO_CP.PR, :UTIL:SED.PR or :UTIL:MBASIC.PR), broadcast a warning with the BROADCAST macro to notify them. Any user may be annoyed if the system comes down without warning. For example,

   ```
   ) WHOS ↲
   ```

```
Processes on the system are:

PID: 1     PMGR      PMGR      :PMGR.PR
PID: 2     OP        OP        :CLI.PR
PID: 3     OP        EXEC      :UTIL:EXEC.PR
  .
  .
PID: 9  JACK    009    :UTIL:CEO_DIR:CEO_CP.PR
PID: 10 ALICE   010    :UTIL:CEO_DIR:CEO_CP.PR
PID: 14 SALLY   014    :CLI.PR

) BROADCAST  System coming down soon. Please log off. ↓

FROM PID 14 (SALLY) : System coming down. Please log off.
```

2. Repeat the WHOS and BROADCAST sequence until the PIDs (Process IDs) of every user who might lose work has vanished — indicating that the user has left the program. For example,

```
) WHOS ↓


Processes on the system are:

PID: 1  PMGR    PMGR   :PMGR.PR
PID: 2  OP      OP     :CLI.PR
PID: 3  OP      EXEC   :UTIL:EXEC.PR
  .
PID: 10 ALICE   010    :UTIL:CEO_DIR:CEO_CP.PR
PID: 14 SALLY   014    :CLI.PR

) BROADCAST Alice -- Please leave CEO. Shutdown soon. ↓
.

) WHOS ↓

Processes on the system are:

PID: 1  PMGR    PMGR   :PMGR.PR
```

```
PID: 2  OP     OP    :CLI.PR
PID: 3  OP     EXEC  :UTIL:EXEC.PR
  .
PID: 14 SALLY  014   :CLI.PR
                 .
```

Here, ALICE's CEO_CP process has vanished — indicating that she has left CEO.

3. When you know that users will not lose work, continue with shutdown. You must get back to PID 2, the master CLI. Type

```
) BYE ↵

AOS  CLI TERMINATING  date  time

You are now
PID: 2  OP     OP    :CLI.PR
)
```

When you type BYE↵ to the standard CLI (on the system console), the PID of the master (2) is displayed. This is a convenience.

4. Run the DOWN macro to bring the multiuser environment down. DOWN also tries to execute product-specific shutdown macros to bring CEO, INFOS, and XODIAC down in an orderly way. Type

```
) DOWN ↵
.
. (Messages about CEO, INFOS, XODIAC messages, if these
. products are on the system)

DOWN processing complete...
↵
)
```

5. After DOWN runs (without a *server process* message) there should be only two processes left: the peripheral manager (PMGR) and the master CLI. For example,

```
) WHOS ↵

PID: 1  PMGR    PMGR  :PMGR.PR
PID: 2  OP      OP    :CLI.PR
```

If you see a *server process* message, this means that INFOS or CEO could not be shut down normally. Find the message in Chapter 15.

If there are any other processes (like your own site's program), terminate them.

When ready, start shutdown by typing

```
) BYE ↵

DO YOU REALLY WANT TO SHUT THE SYSTEM DOWN?
```

6. 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

SYSTEM SHUTDOWN

nnnnnn
!
```

AOS has shut down and the computer has halted. (If the last message is *ABNORMAL SHUTDOWN* , see Chapter 15, the error chapter.)

If your system is connected to another via a modem, disconnect the modem and hang up.

Note that PID 2, the master CLI, can always shut the system down directly via BYE↵. (When a process shuts down, all its sons shut down. The master CLI is the father of all processes except PID 1.) So, if you type BYE↵ to the master CLI when there are processes other than 1 or 2, it gives you a chance to reconsider. It asks

```
YOU HAVE SONS. DO YOU WISH TO TERMINATE?
```

If you type Y↵, all processes under PID 2 will be killed; the system will ask for confirmation again, then shut down. This is a fast way to get down, but it can damage the integrity of CEO and other databases, which should be shut down normally, via the DOWN macro.

After shutdown, you can leave everything as is, you can cut power to the computer, or you can bring up AOS again. If you decide to cut power, remove any inserted diskette from its slot before you do so (turning power off while diskettes are inserted may affect data on the diskette).

Figure 3-2 summarizes normal shutdown.

Abnormal Shutdown

An abnormal shutdown can result from a panic *(FATAL AOS ERROR)* , hardware failure (which may cause a panic), power failure (which may cause a panic), or deadlock.

A *panic* (sometimes called a *crash*) results from an error condition that the operating system recognizes, but cannot fix. The error may involve hardware or software. On a panic, the system console displays the message

```
FATAL AOS ERROR:  nnnnnn
nnnnnn nnnnnn nnnnnn nnnnnn

AOS SYSTEM DUMP
LOAD DISKETTE FOR DUMPING
STRIKE ANY KEY WHEN READY
```

For panic recovery, see *FATAL AOS ERROR* in Chapter 15, the error chapter.

In a *deadlock*, AOS is frozen. There is no response to commands you type on the system console. And CTRL-C does not show (echo). (Just to make sure display isn't suspended by CTRL-S, type CTRL-Q and see if the system comes alive.) You will need to force a shutdown. See "(nothing)" in Chapter 15, the error chapter.

If no one is logged on to user terminals, or if the system console is the only terminal, start with step 3.)

1.
```
) WHOS ↵
.
. (Check for user processes.)

) BROADCAST System coming down -- please log off. ↵
.
```

2. Use the WHOS and BROADCAST macros until all :UTIL:CEO_CP.PR, :UTIL:SED.PR, and :UTIL:MBASIC:MBASIC.PR users are out of their programs (all such PIDs are gone).

3.
```
) BYE ↵

AOS  CLI TERMINATING  date  time

You are now
PID 2     OP     :CLI.PR            (Back in the master CLI.)
)
```

4.
```
) DOWN ↵
.
. (Other product messages...)
.
↵
```

5. Check for processes other than 1 and 2 with WHOS ↵. If any of your own site's programs are still running, terminate them. Then,

```
) BYE ↵
```

6.
```
DO YOU REALLY WANT TO SHUT THE SYSTEM DOWN?   Y ↵
.
SYSTEM SHUTDOWN
.
!
```

If connected on a modem line, disconnect the modem and hang up.

Turn off computer power if you want.

DG-25836

Figure 3-2 System shutdown summary

What Next?

This chapter described the break sequence, startup, and normal shutdown. Now that you know how to start up and shut down your AOS system, you may want to learn more about the CLI. Try the CLI session in the next chapter. Or, learn more about AOS structure and commands in Chapter 5. Chapter 6 discusses CEO.

Using the CLI: A Session with AOS

4

Read this chapter when

- you want to learn how to log on to a user terminal (in a multiterminal system);
- you want to learn to use the CLI.

You don't need this whole chapter to use CEO or a language like MP/BASIC. It's here to show you how to log on to a user terminal, and to give you a sense of the CLI's commands, power, and versatility.

This chapter leads you through a sample session with the CLI. It's for the beginner and assumes no experience with AOS — so if you have built an AOS system or know AOS, you'll see terms that you already know defined. The major sections are

- What Terminal Should I Use?
- In the CLI
- Using Files and Directories
- Filenames and Pathnames
- Printing Files
- Taking a Break
- System Control Sequences
- File Access Control
- Making Files Permanent
- Backup for Your Files
- Help from the CLI
- Ending the Session
- Changing a Password
- What Next?

Just follow the steps described. The session will take between 30 minutes and an hour, but don't hurry.

AOS is a forgiving system, with good error handling and explicit error messages. If you see an error message that isn't mentioned in the text of this chapter, and the message text doesn't explain the problem, you can look up the message in Chapter 15, the error chapter.

What Terminal Should I Use?

Work with a user terminal — if you can — for the user's view of the system. If you can work with a user terminal, skip to the next section.

If there is no user terminal available, use the system console. If AOS is not up, bring it up (Chapter 3). Make sure you're working as a normal user, in a standard CLI — by typing **LOGON username)** as described in Chapter 3. For realistic examples, this chapter shows a username of **ADRIAN** — but your username will probably be different. Next, skip to the section "In the CLI".

User Terminals

A terminal (also known as a VDT or CRT display), looks like a small television screen above a typewriter keyboard. Look for one whose screen displays a message like this:

```
**** AOS REV n / TYPE NEW-LINE TO BEGIN LOGGING ON ****
```

If you can't find a terminal whose screen shows this message, find a terminal attached to the system and turn it on. The ON/OFF switch is behind the screen on the right. After you turn it on, the terminal may print a *Self test* message and beep twice.

If — when you turn it on — the terminal beeps, but displays nothing, the screen brightness may be off. The brightness control is under the right side of the screen unit, toward the front. Slide the control to the center.

If adjusting the brightness doesn't help, the terminal may not be plugged in. Plug it into a wall outlet. Other appliances (like lamps or typewriters) shouldn't share an outlet with a terminal.

After the terminal prints *Self test* , press the) (NEW LINE key); the terminal should ask for *USERNAME* : If not, maybe it isn't on line.

*

Logging on to a User Terminal

Having examined the keyboard, you're ready to log on.

You must have a username and password before you can log on to a user terminal. These prevent unauthorized people from reading or changing authorized people's files.

If you don't know your username and password, find someone in authority (the person who acts as *system manager*) and have him or her set you up with a *user profile*, as described in Chapter 2.

For realistic examples, this chapter shows a username and password of ADRIAN and ADR — but your username and password will probably differ.

To start logging on, press

(NEW LINE key). The terminal will display

```
AOS n / EXEC n date time terminal-name
USERNAME:
```

It wants a username. Type in your username and press ⤶. Take your time; you have 30 seconds to respond. For example, type

```
[illegible]
PASSWORD:
```

Now it wants your password, so type in your password and ⤶; for example, ADR⤶. The system doesn't echo (show) the password, so you can't tell if you make a mistake. But it's okay if you do. When you make a mistake, or your username and password are not what you typed, the system will say

```
INVALID USERNAME-PASSWORD PAIR
USERNAME:
```

If this *INVALID* message appears, type your username and password again. After you've typed the correct username and password, the terminal will display

```
LAST PREVIOUS LOGON  date  time

AOS CLI REV n  date  time
)
```

(If the terminal displays *Main Menu* instead of *AOS CLI* , this means that your initial (startup) program is the CEO Electronic Office or Word Processor. You can exit from CEO by pressing the eleventh function key (CANCEL/EXIT on the CEO template). Otherwise, to learn more about CEO, see one of the CEO books described in Chapter 6.

In the CLI

The) prompt means that you're in the CLI. The CLI is the Command Line Interpreter, the program that gives you access to all other programs. You are logged on and can start using the system.

To sign off the CLI, you can type the CLI command BYE↲ at any time. On a user terminal, this logs you off.

Now, if you are interested *only* in using the CEO Electronic Office, skip to Chapter 6. To use the SED text editor, skip to Chapter 9. For other DG software, go to the pertinent chapter. For a hands-on session that will teach you about using the CLI, read on.

Now that you're in the CLI, you can examine the terminal in the light of *using* it.

The *cursor* shows your position on the screen. It is a box that shows where the next character you type will appear on the screen. A number of key sequences exist just to move the cursor around.

You can control cursor brightness, along with the brightness of other screen characters, with the brightness control under the right corner of the screen unit.

The Keyboard

At the top, in a line, there are status lights. The ON LINE light glows if the terminal is on line; the ALPHA LOCK light glows if ALPHA LOCK is in effect.

Below the status lights, there are four groups of unmarked keys. These *function keys* represent shorthand commands. The commands vary with the software you use, and are identified by plastic *templates* shipped with the software product.

Below the function keys is the main keypad, which resembles the one on a typewriter. Numbers 1 through 0 are at the top, letters below, and space bar at the bottom. There are some extra characters and some extra keys. Some of the important extra keys are ON LINE and CTRL, on the upper left, NEW LINE, on the middle right; and CMD, below NEW LINE. The NEW LINE key (shown in this book as ↲) tells the computer to take action; it's the key you'll use most often. (The CR key next to NEW LINE usually — but not always — works the same way as NEW LINE.)

The SHIFT keys work the same way they do on a typewriter. The ALPHA LOCK key instructs the terminal to print all letters in UPPERCASE. Note that zero and uppercase O (0 and O), and number 1 and lowercase L (1 and l), are *different characters* although they look somewhat alike in this typeset text.

To the right of the alphanumeric keypad, you'll see two other keypads. The numeric keys on the far right are simply that — numeric keys. The pad between the main and numeric keypads is for cursor control; it includes the ERASE PAGE key and may include a TAB key. You'll be using both of these later on.

Upper- or Lowercase?

With the CLI and most other programs, you can use either UPPERCASE or lowercase characters. For example, you can use either `f` or `F` for the FILESTATUS command. The system translates lowercase to uppercase internally, so f is really F.

The ALPHA LOCK status light should be off. If this light is on, all letters you type will be uppercase. If the light is off, all letters will be lowercase unless you press SHIFT. The latter produces more readable type. So, if the ALPHA LOCK light is lit, press the ALPHA LOCK key. This turns the light off.

Using Files and Directories

As a user, you have your *own user directory*, bearing the username you logged on with. Your user directory "belongs" to your username. Type

```
) DIRECTORY ↲
:UDD:ADRIAN

) DIR ↲
:UDD:ADRIAN

) D ↲
ERROR: COMMAND ABBREVIATION NOT UNIQUE
)
```

DIRECTORY is a CLI command; DIR is an abbreviation of the command; and D is *too short* an abbreviation of the command (the CLI cannot tell which command you mean). The DIRECTORY command describes the *working directory* name. The working directory is simply the directory where you are. When you log on, or type `LOGON username↲` on the system console, the working directory becomes your initial user directory. (As always in this chapter, your own username will appear instead of ADRIAN.)

:UDD:ADRIAN describes the *path* to the working directory, ADRIAN. It's a *pathname*. ADRIAN is at the end of a path through other directories. From the top, the path looks like Figure 4-1.

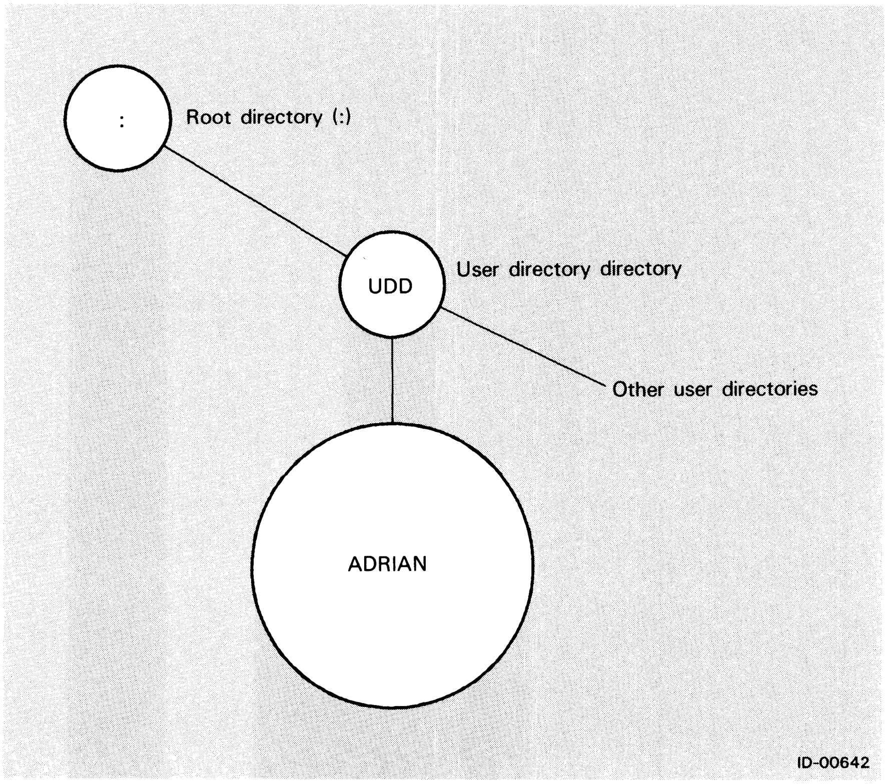


Figure 4-1 Path to a user directory

But directories are practically useless without files. So, type

```
) CREATE  MYFILE ↵
) FILESTATUS  MYFILE ↵
DIRECTORY :UDD:ADRIAN

MYFILE
)
```

Like DIRECTORY, CREATE and FILESTATUS are commands. CREATE creates a file, and FILESTATUS (one of the most useful commands), reports working directory and file status. You *can* use the single letter F as an abbreviation for FILESTATUS.

In these commands, MYFILE is an *argument*. Arguments give the system added information, as (here) the filename to create and check. To separate a command and its arguments, insert at least one space. You can use more than one space, one or more tabs, or combinations as desired.

Type

```
) FILES/AS  myfile ↲
DIRECTORY :UDD:ADRIAN

MYFILE  TXT  14-DEC-84  9:46:50  0
)
```

The /AS appended to FILES is a *switch*. Switches change the meaning of a command, or the act performed on the arguments. The /AS switch tells the CLI to display an assortment of information about the file.

The /AS assortment includes

- the type of file (TXT);
- the date and time the file was created (14-DEC-84);
- the length of the file in bytes (0).

A *byte* is 8 bits or one character; in this case, MYFILE is empty. Obviously, the date and time will differ for your own MYFILE.

Instead of typing the /AS switch, or other FILESTATUS switches, you can use the FSTAT macro. This is an ease of use macro provided by DG. To try it, type

```
) FSTAT ↲


   FILENAME        TYPE  CREATED     LAST MODIFICATION     SIZE
   --------        ----  -------     -----------------     ----

   DIRECTORY  :UDD:ADRIAN

   MYFILE           TXT   14-DEC-84   14-DEC-84 9:46:50      0
```

The FSTAT macro does pretty much the same thing as F/AS, but adds column headers and is easier to use. Also, it sorts the filenames alphabetically, which you cannot see with only one file but will be evident later. CLI macros are one of the great features of AOS.

Type

```
) DELETE/V  MYFILE ↲
DELETED MYFILE
) FSTAT  MYFILE ↲


    FILENAME        TYPE  CREATED     LAST MODIFICATION     SIZE
    --------        ----  -------     -----------------     ----
)
```

DELETE does pretty much what you'd expect: removes the file. The /V switch tells the CLI to verify the deletion, which it did in the example by saying *DELETED MYFILE*. The FSTAT macro displays no name, which means that the file was, in fact, deleted. Type

```
) CREATETEXT MYFILE ↲
When you have typed all the text you want in this file,
press NEWLINE, and type right parenthesis NEW-LINE to
terminate entry.

)) Greetings from MYFILE. ↲            (Type this text and ↲.)
)) )↲
           you type this parenthesis and ↲.
)
```

CREATETEXT is another ease of use macro. It tells the CLI to create a file, take the text you type, and insert it in the file. During the insertion, the CLI displays two prompts [*))*] instead of one. To end the insertion, type one right parenthesis and ↲, as above. Type

```
) FSTAT   MYFILE ↲

    FILENAME        TYPE  CREATED     LAST MODIFICATION     SIZE
    --------        ----  -------     -----------------     ----

    MYFILE          TXT   14-DEC-84   14-DEC-84 9:52:50     24
```

```
) TYPE  MYFILE ↵
Greetings from MYFILE.
)
```

The new MYFILE is type TXT and has 24 bytes (characters). Your MYFILE might have more or less, depending on the number of spaces you inserted. The TYPE command (which you can abbreviate TY) displays the contents of a file.

```
) CREATETEXT  MACRO.CLI ↵
When you have typed all the text you want in this file,
.
.
)) WRITE  MYFILE contains ↵
)) TYPE  MYFILE ↵
)) )↵
            you type this parenthesis and ↵.
)
```

One of the great features of the CLI is its custom command groups called CLI *macros*. If a filename consists of name and .CLI, when you type the name↵, the CLI will try to execute all commands in the file. Try it:

```
) MACRO ↵
MYFILE contains
Greetings from MYFILE.
)
```

You created MACRO.CLI to write "MYFILE contains", and then type MYFILE; and it does so. The WRITE command simply writes the characters that follow it; it's useful to see how things will look on the terminal.

Now that you have created two files, check them:

```
) FSTA ↵
ERROR: NOT A COMMAND OR MACRO, FSTA
FSTA
```

Unfortunately, you cannot abbreviate macro names. And, you need to type them accurately. Try it again:

```
) FSTAT ↲


   FILENAME        TYPE  CREATED     LAST MODIFICATION     SIZE
   --------        ----  -------     -----------------     ----
   .

   DIRECTORY  :UDD:ADRIAN

   MACRO.CLI        TXT  14-DEC-84    14-DEC-84 9:54:22      34
   MYFILE           TXT  14-DEC-84    14-DEC-84 9:52:50      24
   .

)
```

FSTAT (or the FILESTATUS command) *without* an argument describes all files in the working directory. There might be files other than MACRO.CLI and MYFILE. Any such files were supplied by someone else to set certain conditions for you as a user.

A New Directory

Now you can create a new directory for the two little files. The primary reason for directories is to help keep track of different files.

```
) CREATEDIR LEARNING ↲

) FSTAT ↲


   FILENAME        TYPE  CREATED     LAST MODIFICATION     SIZE
   --------        ----  -------     -----------------     ----


   DIRECTORY  :UDD:ADRIAN

   LEARNING         CPD  14-DEC-84   14-DEC-84 9:56:44        0
   MACRO.CLI        TXT  14-DEC-84   14-DEC-84 9:54:22       34
   MYFILE           TXT  14-DEC-84   14-DEC-84 9:52:50       24
```

```
) DIR LEARNING ↵
) DIR ↵
:UDD:ADRIAN:LEARNING
)
```

Creating directories is easy: you just use the CREATEDIR macro and specify the new directory name as an argument. To limit the directory size, you can specify a number after the name. This sets a maximum size for the new directory in 512-byte blocks. For example: CREATEDIR RECORDS 50 creates a directory with a maximum size of 50 blocks. The CREATEDIR macro creates all directory files as type CPD. You can list only directories by typing `FSTAT/DIR↵` — here in LEARNING, there are none.

To change the working directory, use the DIRECTORY command with a directory-name argument.

```
) DIR ^ ↵
) DIR ↵
:UDD:ADRIAN
)
```

A caret (press the SHIFT and 6 keys on the alphanumeric keypad) is one of the CLI shorthand characters. Used with DIR, it tells the CLI to go *up* one directory. As you can see, it worked; the working directory is now the original user directory instead of directory LEARNING.

```
) MOVE/V  LEARNING  macro.cli  myfile ↵
MACRO.CLI
MYFILE
)
```

MOVE, another handy command, copies files into another directory. The /V switch told the CLI to verify the move, which it did. The first argument to MOVE must be the destination directory name, here LEARNING. The following arguments are the filename(s) you want moved.

The original files are still in your initial user directory; copies are in directory LEARNING.

Screen Editing

Now for a little screen editing. Press the CTRL key and hold it down. While you're holding CTRL down, press the A key. This is called CTRL-A. Release both keys. On your screen, you will see the last command line repeated, just as you originally typed it:

```
) MOVE/V  LEARNING  macro.cli  myfile
```

Look at the screen cursor (your position). It is at the end of the line, after myfile. Press ↵:

```
↵
WARNING: FILE ALREADY EXISTS, FILE LEARNING:MACRO.CLI
WARNING: FILE ALREADY EXISTS, FILE LEARNING:MYFILE
)
```

By pressing ↵, you re-entered the MOVE command, which provoked an error message because the files already existed in directory LEARNING. The error is okay, it was deliberate, no harm done. Type CTRL-A again, and the MOVE command will appear once more:

```
) MOVE/V  LEARNING  macro.cli  myfile
```

Now let's edit the command line on the screen. The cursor is at the end of the line. Press the HOME key to the right of the main keyboard. HOME moves the cursor to the beginning of the line, after the prompt.

Now, type DELETE/V to overwrite MOVE/V. The command should now look like this:

```
) DELETE/VEARNING  macro.cli  myfile
```

The cursor will be one space ahead of the last character typed, in this case on the E after the V. Press the space bar until you've wiped out the rest of EARNING. The command line should look like this:

```
) DELETE/V        macro.cli  myfile
```

The cursor should be before macro.cli. Now type CTRL-F. CTRL-F moves the cursor forward to the next word. Keep typing CTRL-F until the cursor is at the end of the command line.

With the cursor at the end of the line, press ↵. This tells the CLI to execute the DELETE command:

```
↵
DELETED macro.cli
DELETED myfile
)
```

You just used screen editing to change the MOVE command to a DELETE command. This DELETE just eliminated duplication, deleting the files from your user directory. Copies of the files remain in directory LEARNING.

For practice, try it again. Type CTRL-A to redisplay the DELETE line. Then press HOME to move the cursor to the beginning of the line. Finally, press the CR key to kill the entire command line.

The major screen editing characters follow, in Table 4-1.

Table 4-1 Major screen editing characters

Character	What it Does
CTRL-A	redisplays the last command typed.
CTRL-F	moves the cursor forward to the next word.
CTRL-B	moves the cursor backward to the previous word.
→ (key)	moves the cursor forward one character.
← (key)	moves the cursor backward one character.
CR or ERASE EOL (key)	deletes all characters to the right of cursor. CR also tells the CLI to execute the command.
CTRL-E	inserts new text or terminates an insert.
DEL (key)	deletes the character to the left of the cursor.

Although describing it requires many steps, screen editing is really very easy and fast. It can save a lot of aggravation, when you make a typing mistake or want to re-enter a long command line. (The CTRL key by itself does nothing, but in conjunction with other characters — like A, E, or F — it can do a lot.)

You may be using screen editing often — so don't be afraid to experiment with the screen editing characters.

Filenames and Pathnames

At this point, you're in your user directory, having just moved your files into directory LEARNING. Try to type one of the files:

```
) TY  MYFILE ↲
WARNING: FILE DOES NOT EXIST, FILE MYFILE
)
```

Try a *pathname* that includes directory LEARNING:

```
) TY  LEARNING:MYFILE ↲
Greetings from MYFILE.
)
```

A pathname specifies a path to the file and always includes the file's name. Generally, if a file is not in the working directory, the pathname must include one or more directory names.

If the file is in a *subordinate* (lower) directory, you need specify only the subordinate directory name(s) in the pathname — as above.

If the file is in the working directory, the filename can be the pathname:

```
) DIR LEARNING ↲
) TYPE  MYFILE ↲
Greetings from MYFILE.
)
```

If the file is in a *superior* (higher) directory, you must specify a pathname that includes the directory.

A file's full pathname always works, but can be laborious to type — so use the ^ character (SHIFT and 6 keys) as needed. In most cases, as you work with the CLI, you'll find it easier to DIR into the directory you want instead of using long pathnames.

Legal Filenames and Pathnames

A filename can contain any of the alpha characters A-Z, numbers 0-9, underscore (_), period (.), question mark (?) and dollar sign ($). It can be up to 31 characters long. This allows you to use very explicit filenames. For example,

```
TRANS_$INCOMES_RECEIVED?MAY.10
```

is a valid, if cumbersome, filename.

Each pathname consists of one or more legal filenames. Each directory name must be followed by a colon (since only directories can contain other files). A space is a delimiter and cannot be part of a filename or pathname.

Keeping Track of Filenames

Eventually, there may be many files in your user directory. Subordinate directories can help classify these, but even within a directory there can be hundreds of files.

For an alphabetical list of filenames, use the FSTAT macro (or FILESTATUS command with the /SORT switch). Type

```
) FSTAT ↲


   .
   DIRECTORY     :UDD:ADRIAN:LEARNING
   .
   MACRO.CLI     TXT    14-DEC-84    9:54:22       34
   MYFILE        TXT    14-DEC-84    9:52:50       24


)
```

Directory LEARNING doesn't provide much of a challenge for alphabetic sorts, but the example makes the point.

Templates to Find Filenames

People often forget the full names of files. Filename *templates* allow them to access a file without typing the full name. A template specifies a set of filenames. The most common template characters are

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 subordinate directories.* Without this template, the search occurs only in the working (or specified) directory.

To see all filenames that are six characters long, and that don't contain a period, type

```
) FSTAT  ****** ↵
.
DIRECTORY  :UDD:ADRIAN:LEARNING

MYFILE ... ... .... .....
)
```

To see *all* filenames that don't contain a period, type

```
) FSTAT  - ↵
.
DIRECTORY  :UDD:ADRIAN:LEARNING

MYFILE ... ... .... .....
)
```

Or to see all filenames whose names end in .CLI, type

```
) FSTAT  -.CLI ↵
.
DIRECTORY  :UDD:ADRIAN:LEARNING

MACRO.CLI ... ... .... .....
)
```

To see all filenames, including periods, that begin with M, type

```
) FSTAT  M+ ↵
.
DIRECTORY  :UDD:ADRIAN:LEARNING

MACRO.CLI ... ... .... .....
MYFILE ... ... .... .....
)
```

To see all filenames that do *not* end in .CLI, type

```
) FSTAT  +\+.CLI ↵
DIRECTORY  :UDD:ADRIAN:LEARNING
.
MYFILE ... ... .... .....
)
```

The # character allows you to search multiple directories. For example, type

```
) FSTAT  :UDD:ADRIAN:# ↵

DIRECTORY  :UDD
ADRIAN
.
DIRECTORY  :UDD:ADRIAN
LEARNING
.
DIRECTORY  :UDD:ADRIAN:LEARNING
.
MACRO.CLI ... ... .... .....
MYFILE ... ... .... .....
)
```

To search *all* your directories for a file named FOO, type

```
) FSTAT  :UDD:ADRIAN:#:FOO ↵
```

. (pause for search)
)

It found nothing because you have no file named FOO.

For a hypothetical example, assume that you have a lot of CLI macro files, which conventionally end in .CLI. You want to see and sort all their names. You'd type

```
) FSTAT  +.CLI ↲
```

The resulting display might be

```
BIORHYTHM.CLI    UDF   12-JAN-85     9:01:56     278
BOOMER.CLI       UDF   28-DEC-84    14:48:01    1665
FSTAT.CLI        UDF   11-JUN-83    11:01:33     110
SETUP.CLI        UDF   10-JAN-85    10:46:50     194
```

Templates can also help you to organize your files by category in different directories, back them up selectively, or delete them selectively.

The CLI commands and macros that allow template characters include FILESTATUS, FSTAT.CLI, DELETE, FULL_BACKUP, INC_BACKUP, MOVE, PERMANENCE, and TYPE (the backup macros are described in Chapters 7 and 5). If you ever want to use a template and are not sure that your command accepts it, just try the template. But be careful with the DELETE command and templates: if you accidentally insert a space after a template character (for example, you type DELETE□+□.XX↲), you could delete all files in the directory.

Printing Files

If your system has a printer, you'll want to print files. To print one or more files, use the QPRINT command, which can be abbreviated QPR. Type

```
) QPR  MACRO.CLI ↲
QUEUED, SEQ=1, QPRI=127
)
```

QPRINT sends the text of a file to the printer queue. A queue is needed to manage the printer while it prints — allowing you to continue work at the terminal, instead of waiting for printing to end.

Handling the Printer

At the printer, see if there is anyone (like an operator) to work the printer controls. If not, do it yourself:

1. Wait for the printer to finish printing your file (here, the file is tiny, so you won't have to wait).
2. If the paper is in separate 8.5 x 11 inch sheets, simply pick up the printed file; you're done.
3. If the paper is in a pile of connected sheets, press the printer ON LINE switch. The ON LINE light should go out. (The letter-quality printer has a LOCAL switch that you use in the same way as an ON LINE switch.)
4. Press the FORM FEED switch. This tells the printer to feed a form. Tear off the paper after the blank form.
5. *Important*: press the ON LINE switch again. The ON LINE light should glow. If you forget this step, the printer can't print; all print requests that people make will wait in the queue until someone puts it back on line.

Now you can examine the printed text of MACRO.CLI. Not exactly *War and Peace* — but printing it did show you how to work the printer. (To insert paper, correct printer faults, and so on, see the printer description in the hardware book that accompanies this one.)

In many CLI commands — including FSTAT and FILESTATUS — you can send text to the printer instead of to your terminal screen. To do this, use the /L=@LPT switch. This switch is very handy for things like big file sorts. For example:

```
) FSTAT/L=@LPT ↵          (Sends directory filestatus to printer.)
```

Taking a Break

At this point, you've learned about logging on; about the DIRECTORY, command, the CREATETEXT and CREATEDIR macros, FILESTATUS and FSTAT, the MOVE, DELETE, TYPE, WRITE, and QPRINT commands; and about switches.

You've created a directory, moved files into it, tried screen editing, learned something about filenames, pathnames, and templates, and printed a file.

Your directory and your file structure look like Figure 4-2.

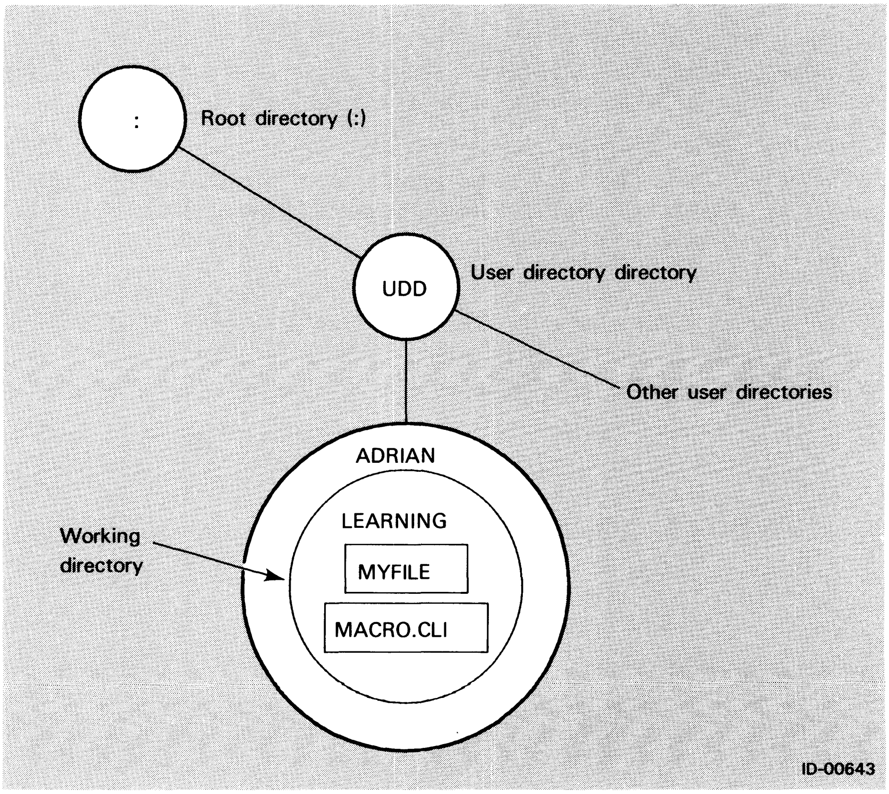


Figure 4-2 In working directory LEARNING

You may want to stop for a while. If so, and you're on a user terminal, log off by typing BYE↵. (You can't log off if you're working on the system console.) Later, start up as before.

To proceed, read on.

Running Other Programs

The CLI is a versatile and powerful program. But it's most useful as a companion to other programs — as a file maintenance aid and interface to programs like word processors, text editors, compilers, and other operating systems (like CP/M-86).

You can execute another program from the CLI via either the XEQ or or EXECUTE command (they do the same thing), the PROCESS command, or a macro that contains one of these commands.

The new program runs *under* the CLI. When the new program terminates, or you abort it with CTRL-C CTRL-B, the original CLI continues on your terminal.

For example, we'll have you try a program called FCU.PR. FCU (Forms Control Utility) creates form specifications for file printing; it is not an often-used program, but we chose it because it is supplied with all AOS systems and any user can execute it. Type

```
) XEQ  FCU ↲

AOS Forms Control Utility Revision n ....

.
```

You've executed the FCU program and it's running on your terminal. To terminate it, type

```
) BYE ↲

FCU terminating  date  time

)
```

You terminated FCU and the CLI returned. This is generally the procedure you'll follow: you'll run the new program, work with it as desired, then terminate it.

The DG-supplied program file(s) are stored in a directory other than your user directory (in :, :UTIL, or below :UTIL). The CLI finds these programs by a feature called a *searchlist*. This has been made automatic in AOS for desktop systems; if you want, you can learn more about searchlists by reading the CLI manual.

For more information on the programs in the root (:) and :UTIL directories, see Table 1-1 in Chapter 1.

System Control Sequences

There are several system CTRL sequences that you will find very useful (aside from the screen editing CTRL characters shown earlier). These CTRL sequences do such things as suspend and resume display, clear the screen, interrupt execution of a CLI command, and abort the current process. All CTRL sequences are described in Chapter 1. For convenience, we'll repeat the most common ones here. They are

CTRL Sequence	**What it Does**
`CTRL-S`	Suspends display. To continue display, type CTRL-Q.
`CTRL-Q`	Resumes display suspended by CTRL-S.
`CTRL-C CTRL-A`	Interrupts execution of current command (in CLI, BASIC, and some other programs).
`CTRL-C CTRL-B`	Aborts current program, returns to its parent process. If entered from a user CLI, CTRL-C CTRL-B logs you off the system. From master CLI (PID 2), CTRL-C CTRL-B is ignored; this prevents people from bringing the whole system down by typing two characters.

Let's try some of these sequences, using the files in :UTIL. Type

```
) DIR :UTIL ↵     (This makes :UTIL the working directory.)
) FSTAT ↵         (This displays status of all files.)
.
.
CTRL-S            (Type this immediately after FSTAT.)
```

As you see, display has stopped. To continue from where it stopped, type

```
CTRL-Q
```

Display continues. (If the system finishes displaying filenames, type `FSTAT ↵` again.)

```
CTRL-S
CTRL-Q
```

CTRL-S and CTRL-Q can make a long file readable by breaking it into segments. If *ever* the system appears frozen, try typing CTRL-Q; the problem may just be that display was suspended by CTRL-S.

To *interrupt* a display or file, use CTRL-C CTRL-A. To try this, press CTRL-A and ↵ to repeat the FSTAT line:

```
) FSTAT ↵
.
. (names)

CTRL-C CTRL-A

ERROR: CONSOLE INTERRUPT
)
```

A *CONSOLE INTERRUPT* isn't really an error, but the CLI had to say something about it. (By *CONSOLE* , it means terminal.) Try all CTRLs again: type CTRL-A to redisplay the FSTAT line, press ↵, and type CTRL-S and CTRL-Q to freeze and thaw the display. Repeat the these CTRLs until you're comfortable with them.

Now, all that remains is CTRL-C CTRL-B. Don't type this now; it would abort your CLI. To try it, fire up FCU again:

```
) XEQ FCU ↵
.
.
CTRL-C CTRL-B

*ABORT*
CONSOLE INTERRUPT
ERROR: FROM PROGRAM
XEQ,FCU

)
```

As you can see, CTRL-C CTRL-B has a mighty effect. It's most useful when you want to stop a program before it finishes normally. Otherwise, avoid CTRL-C CTRL-B.

File Access Control

In a multiuser system, people need access to, and privacy for, their own files. User and system files need protection from accidental (or malicious) deletion. AOS file access control provides this. For an example, type

```
) FSTAT  :UPD:+ ↲
WARNING: FILE ACCESS DENIED, FILE :UPD
)
```

This FSTAT :+↲ tells the system to list all the filenames in the :UPD directory. The *FILE ACCESS DENIED* error message shows that you are denied access (even to see the file names) in the directory. This *DENIED* message appears — generally — whenever you try to access files in system or other user directories.

The system allows only the person who owns a user directory to read, modify, or delete the files in it. Everyone can read files in a public directory, like :UTIL. This is generally a good arrangement; it gives each user exclusive access to his own files and to public files.

But, there may be times when you want to share a file — allow other users to access it. To do this, use the UNPROTECT macro. UNPROTECT gives all users owner access to one or more of your own files. For example, assume you have a series of files whose names end in REPORT, and you want to give everyone access to them. You could type

```
) UNPROTECT  +REPORT ↲

The following file(s) can now
be accessed by any system user:

MAY_REPORT              (filenames unprotected)
APR_REPORT
JUNE_REPORT

)
```

Now, any user can read (or change or delete) any of these files. All he or she needs to know is the file pathname, which, for user directories, always begins with :UDD:username. For example, if user AKW knew that the pathname was :UDD:JACK:MAY_REPORT, he could type

```
) TYPE :UDD:JACK:MAY_REPORT ↵
     or
) QPRINT :UDD:JACK:MAY_REPORT ↵
```

And the Jack's file would be printed on AKW's terminal or the printer. If Jack didn't unprotect the file, AKW would see

```
WARNING: FILE ACCESS DENIED, FILE :UDD:JACK:MAY_REPORT
```

instead of the file text. After unprotecting one or more files, any user can protect them again with the PROTECT macro. For example

```
) PROTECT  +REPORT ↵
.
. (verification)
.
```

You can unprotect, or protect, any and all files in and below your user directory.

One way to sidestep all system access controls is with the SUPERUSER command. SUPERUSER is needed for the person who runs the system, does backup copies of disk files, and so on. Only the master CLI, PID 2, can turn on SUPERUSER — it is further described in Chapter 5.

Making Files Permanent

If you are ever careless with a template, you might accidentally delete a lot of cherished files, or even directories. For example, type

```
) CREATEDIR  FOO ↵
) DIR  FOO ↵
) CREATETEXT  FOOFILE ↵
.
)) Hi ↵
)) )↵
) DIR  ^↵
) FSTAT  FOO ↵


   DIRECTORY  :UDD:ADRIAN:LEARNING
   FOO      CPD      14-DEC-84      10:12:48      4608
```

```
) DEL/V  FOO ↵
DELETED FOO
)
```

It's as easy as DEL/V to delete a directory or file — but not if you turn PERMANENCE on for it. PERMANENCE makes the file or directory permanent; no one can delete it until you turn PERMANENCE off. Try it:

```
) CREATEDIR  FOO ↵
) PERMANENCE  FOO ON ↵
) PERM/V  FOO ↵
FOO ON

) DEL/V  FOO ↵
WARNING:CANNOT DELETE PERMANENT FILE, FILE FOO

) PERM  FOO  OFF ↵
) DEL/V  FOO ↵
DELETED  FOO )
```

As you see, PERMANENCE is an easy way to protect your files from deletion. The command accepts templates, so you can set permanence easily for all your files; for example, by `PERM + ON↵`. When you make a file permanent, you may also want to make its parent directory permanent (since PERMANENCE will not save a file if its parent directory is nonpermanent and you delete the directory).

Backup for Your Files

No matter how small a computer system, it should have some procedure for file backup (copy for safekeeping). Then, if files are inadvertantly deleted or otherwise lost, they can be restored from the backup medium. The AOS files you received from DG do not absolutely *need* to be copied, since they can be reloaded from the DG diskettes. But things like user directories and CEO file structure are often irreplacable, and should be backed up.

This section shows you how to back up the files you created during this session. All backup procedures are detailed in Chapter 7. But, since this backup is easy to do — you might want to try it now.

Doing a Full Backup of Your User Directory

Get a diskette, remove it from its paper envelope, and make sure the write-enable notch is open (covering this notch prevents the system from writing to the diskette, and you want to write material to diskette).

If there is no label on the diskette, apply one as described in Chapter 7, section "Handling and Storing Diskettes".

Insert the diskette in the unit, as shown in Chapter 2, Figure 2-1. If you have two diskette slots, use the rightmost one (this is the primary unit).

DG has supplied several macros for backup and recovery: FULL_BACKUP.CLI, INC_BACKUP.CLI, and RESTORE.CLI. The macro FULL_BACKUP copies *all* files in and below the working directory to diskette. INC_BACKUP copies only those created since the last backup (it uses a file created by one of the backup macros the *last* time files were backed up).

The easiest and most efficient way to handle backups is to do a full backup only occasionally, and do an incremental backup more often (perhaps each day, or on alternate days). All details on backup are covered in Chapter 7. For now, let's settle for a FULL_BACKUP from the user directory, just to see how it works.

First, get to your initial user directory (for example, :UDD:ADRIAN, or :UDD:OP) using the DIR command. Then type

```
) FULL_BACKUP/L==LIST_OF_FILES ↵
```

This command tells the system to dump all files in and beneath the working directory to the primary diskette. The second equal sign (=) tells the system to stay within the working directory. So, it creates a file named LIST_OF_FILES in the working directory and writes each filename backed up to this file. Thus, file LIST_OF_FILES is a record of all files copied. From this session, there won't be many files.

The macro dialog is as follows

```
    ** Full backup from directory :UDD:ADRIAN at 15:57 on 28-Dec-83 **

    Please insert the first diskette to receive backup material in the
primary -- rightmost -- unit. The device name is @DPM0. This
diskette and others used for backup will be overwritten -- so
don't use diskettes that have material you want to keep.

Please number the envelope of each diskette as it is filled so
that -- if needed -- the diskette can be restored in correct
order.

Press NEW-LINE when ready to begin backup.
```

You've already inserted the diskette, so press ↵.

```
Beginning file backup - a list of files dumped to diskette follows.
.
. (it writes names of files to the disk file specified with /L)
.
File backup complete at time.

)
```

The backup is done. Remove the diskette from its unit. Then write the date, type of backup (FULL_BACKUP), and diskette number (#1) on the diskette label, and replace the diskette in its envelope. The diskette number is not essential for a single-diskette backup, but it *is critically important* for multiple-diskette backups. This is so because — to restore the files — the diskettes must be restored in the same order that they were backed up. You could easily lose track of the order in a dump that included many diskettes.

You're done! If somehow these files were deleted from the hard disk, you could restore them via the RESTORE macro. In the original directory, you'd insert the diskette in the primary unit, then type RESTORE↵.

Help from the CLI

If you forget a command name, you can get help by typing

```
) HELP ⤶

TOPICS ARE:


Change *COMMANDS
       *PSEUDO-MACROS

to

*1_SWITCH       *2_SWITCH        *AFTER_SWITCH     *CLI_INPUT *COMMANDS
*CONDITIONALS *CONTROL_CHARS  *CURSOR_CONTROL  *DISPLAY   *ENVIRONMENT
*EXCEPTIONS    *FILENAMES       *GENERAL_FILES    *I_SWITCH  *LFE
*LINE           *LINKS            *L_SWITCH          *MACROS    *M-SWITCH
*NEWLINE        *PATHNAMES        *PSEUDO_MACROS     *P_SWITCH  *QUEUES
*Q_SWIT         *SED              *SWAT              *SWITCHES  *TEMPLATES
*TOPICS

FOR MORE HELP ABOUT ANY TIME ABOVE TYPE 'HELP *ITEM'
```

This is a list of CLI help *topics.* Many of the topics involve concepts that you will not use in routine processing, so they are not explained in this book. If you want further information on any of these topics, see the *Command Line Interpreter (CLI) User's Manual* and *AOS and AOS/VS User's Handbook.*

You can get more help on any topic by typing `HELP *topic`. For a list of CLI commands, type

```
) HELP *COMMANDS ⤶
```

The CLI then lists all its commands:

```
CLI COMMANDS ARE:

ACL             ASSIGN          BIAS            BLOCK           BYE
CHAIN           CHARACTERISTICS CHECKTERMS      CLASS1          CLASS2
COMMENT         CONNECT         CONTROL         COPY            CREATE
CURRENT         DATAFILE        DATE            DEASSIGN        DEBUG
DEFACL          DELETE          DIRECTORY       DISCONNECT      DISMOUNT
DUMP            ENQUEUE         EXECUTE         FILESTATUS      HELP
HOST            INITIALIZE      LEVEL           LISTFILE        LOAD
LOGEVENT        LOGFILE         MESSAGE         MOUNT           MOVE
PATHNAME        PAUSE           PERFORMANCE     PERMANENCE      POP
PREFIX          PREVIOUS        PRIORITY        PROCESS         PROMPT
PRTYPE          PUSH            QBATCH          QCANCEL         QDISPLAY
QFTA            QHOLD           QPLOT           QPRINT          QPUNCH
QSNA            QSUBMIT         QUNHOLD         RELEASE         RENAME
REVISION        REWIND          RUNTIME         SCREENEDIT      SEARCHLIST
SEND            SPACE           SQUEEZE         STRING          SUPERPROCESS
SUPERUSER       SYSID           SYSINFO         SYSLOG          TERMINATE
TIME            TRACE           TREE            TYPE            UNBLOCK
VAR0            VAR1            VAR2            VAR3            VAR4
VAR5            VAR6            VAR7            VAR8            VAR9
WHO             WRITE           XEQ

FOR MORE HELP ABOUT ANY ITEM ABOVE, TYPE 'HELP ITEM'
```

You can learn the purpose and switches of any command by typing HELP/V command↲. For example, try FILESTATUS:

```
) HELP/V  F ↲


FILESTATUS  DISPLAY INFORMATION ABOUT ONE OR MORE FILES

FORMAT:     FILESTATUS <TEMPLATE> <TEMPLATE> ...
                (IF NO ARGUMENT IS SPECIFIED, "+" IS USED)

COMMAND SWITCHES:  /1= /2= /L(=) /Q
        /ASSORTMENT  DISPLAY FILE TYPE, CREATION TIME,
                     CREATION DATE, AND FILE LENGTH
        .
        .
        /SORT        SORT FILENAMES ALPHABETICALLY
```

Some common CLI commands, their switches, and all the desktop system macros are described in Chapter 5. *All* CLI commands and switches (but not the macros) are given in the *Command Line Interpreter (CLI) User's Manual (AOS and AOS/VS).*

Ending the Session

You're done with the CLI session. If you're working on the system console, you can simply leave the system console, or you can shut down AOS as described in Chapter 3.

If you're on a user terminal, log off by typing

```
) BYE ↓

AOS CLI TERMINATING  date  time

PROCESS n TERMINATED
CONNECT TIME  hours:minutes:seconds
USER 'username' LOGGED OFF @CONn  date  time

**** AOS REV n / TYPE NEW-LINE TO BEGIN LOGGING ON ****
```

The BYE command terminates the current CLI process. Typed on a user terminal, it logs you off and frees the terminal for anyone with a valid username and password to log on. If the terminal will not be used for a long time (like overnight or over the weekend), turn it off using the switch at the right, back of the screen.

To log on again, press ⤶ (turning the terminal on first, if needed), then type username⤶ and password⤶, as described at the beginning of the chapter.

Changing Your Password

You can always change a password by deleting the user profile and recreating it with the new password. Use the PROFILE macro shown in Chapter 5.

Another way — if your system has multiple terminals — is to change your password when you log on. For this to work, the multiuser environment must be up.

CAUTION *To use CEO in conjunction with a remote DG host system, you must have an AOS profile with the same username and password on both systems. If you change the password on the desktop system, you won't be able to use the remote CEO Mail or Calendar. If this situation pertains, arrange to have your password changed to the new password on the remote host (or log on to the host as described in Chapter 12, then change the password yourself as described here). Then change your password on the desktop system.*

To change your password, log on as usual, with username and password; but, after you type the password, press the **ERASE PAGE** key instead of ⤶. The system will then ask for the new password, and you'll type it. The password must be 3 through 15 characters long, and can include all valid filename characters: A-Z, 0-9, ?, $, _ (underscore), or . (period).

For example, take the following dialog, in which a user whose username and password are both JOAN changes her password to JKN, then logs off and logs on again with her new password. When you choose the new password, remember that you'll have to type it each time you log on to a user terminal, so keep it short and easy to remember.

```
*** AOS REV n / TYPE NEW-LINE TO BEGIN LOGGING ON ***

↲                               (Press ↲ to start logging on.)

AOS EXEC REV n ..............

USERNAME: JOAN ↲                (Type your username.)

PASSWORD: JOAN (ERASE PAGE)     (Type your old password and press
                                the ERASE PAGE key.)

ENTER YOUR NEW PASSWORD: JKN↲   (Type the new password and ↲; the
                                password doesn't echo.)

--NEW PASSWORD IN EFFECT --     (The system confirms.)
...LAST PREVIOUS LOGON...       (The system logs the user on.)

) BYE ↲                         (The user logs off.)

.... TYPE NEW-LINE TO BEGIN LOGGING ON ****

↲                               (She logs on again to check
AOS EXEC n date ....            the new password.)

USERNAME: JOAN ↲                (Username.)

PASSWORD: JKN ↲                 (New password doesn't show.)

...LAST PREVIOUS LOGON...       (The new password works.)
)
```

The new password remains in effect until you change it again. If you change your password, choose one that you can remember easily — because, if you forget it, you won't be able to log on. (If you ever *do* forget your password and/or username, you will be unable to log on to a user terminal until your profile is deleted and recreated — as described in Chapter 5, PROFILE macro.)

If privacy and security are very important to you, don't choose obvious names or initials for your password. Good choices include a mixture of alphabetic and numeric characters, and/or special characters; for example

```
CALL_ME_ISHMAEL
```

What Next?

You have finished the entire CLI chapter. You've logged on; created and deleted files; created a directory; moved files into it; tried screen editing, learned something about filenames, pathnames, and templates; and printed a file. You've used the control characters; executed a program; backed up some files; and logged off. Lastly, you learned how to change your password.

Congratulations. This session included most of the concepts and many of the commands you'll ever need with the CLI.

For a description of system file structure, and CLI commands and macros, continue to Chapter 5.

To start working with the CEO Electronic Office, skip to Chapter 6.

For details on file backup, skip to Chapter 7; or to build programs in DG computer languages, skip to Chapter 10.

Understanding AOS File Structure and Commands

5

Read this chapter when

- you want to know how parts of AOS interrelate;
- you want details on AOS and other software structure;
- you want to know about an often-used CLI command, macro, or program supplied with AOS.

This chapter explains AOS file structure. Then, it describes the CLI, often-used CLI commands and macros (like the ones in the Chapter 4 session). You need *not* read this chapter to start up CEO or use BASIC.

Not all CLI and EXEC commands are described here. All CLI commands are explained in the *Command Line Interpreter CLI User's Manual*; and EXEC commands are described in *How to Generate and Run Your AOS System,* available on order.

The major sections are

- AOS File Structure
- Commands, Macros, and Programs
- What Next?

AOS File Structure

This section explains AOS file structure — briefly — in the context of desktop systems. (It's also described in the CLI manual.)

If you installed this AOS system (Chapter 2) or tried the CLI session (Chapter 4), you already have some sense of the directory structure.

In a running AOS system, with XODIAC networking and one computer language (MP/BASIC) installed, but without the CEO Electronic Office, the file structure looks like Figure 5-1. The AOS programs in each directory are described in the glossary.

From the top, left to right, the files in Figure 5-1 are as follows:

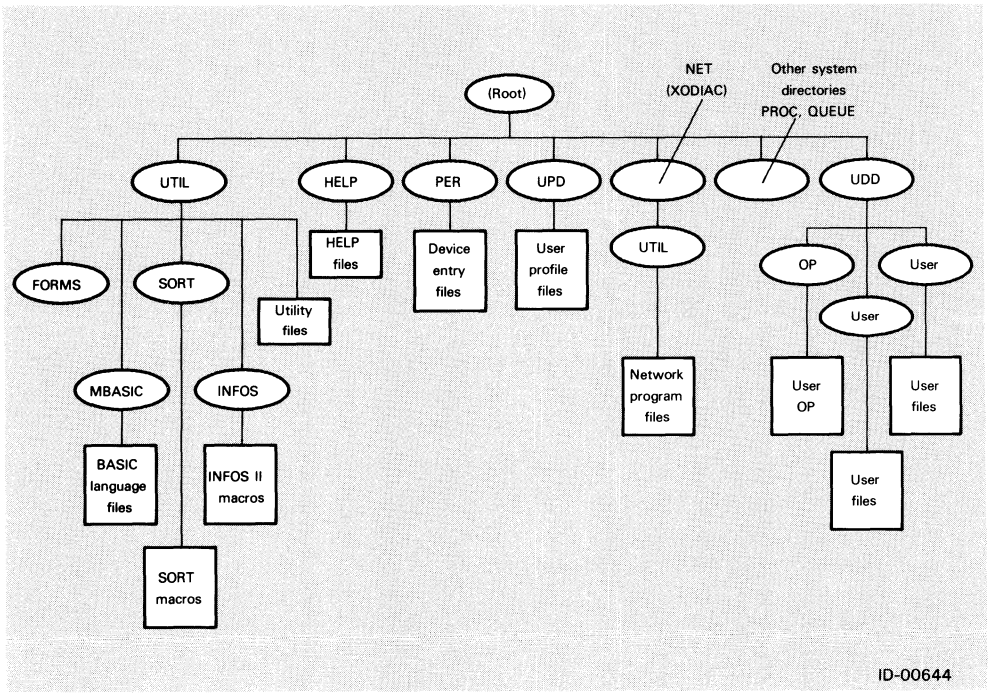


Figure 5-1 AOS and related software file structure

Filename (pathname) **Description**

: (:)

This is the root directory, which contains all other directories. It was shipped with AOS and created on the hard disk by the AOS Disk Formatter. Its nondirectory files include the CLI and peripheral manager (PMGR).

UTIL (:UTIL)

This is the utilities directory, shipped with AOS. UTIL contains most of the AOS utility program files: EXEC, the Disk Formatter for diskettes, and so on. Some INFOS II and Sort/Merge files are here. Also, :UTIL contains most *General Language Development Package (GLDP)* products, if these were ordered with AOS and installed. These are outlined in Chapter 1, Table 1-2.

FORMS (:UTIL:FORMS)

This is the forms directory, used for special forms printing and by CEO. It was created when the AOS was installed on the hard disk.

MBASIC (:UTIL:MBASIC)

This is the directory for MP/BASIC, created when MP/BASIC was installed on the hard disk. Generally, BASIC users will keep their programs in their own user directories, not in this directory.

Like MP/BASIC, other DG products have their own directories *under* :UTIL. The language directory is created when the language product is installed on the hard disk. All the languages available are named in Chapter 1, Table 1-3 and in Chapter 10.

INFOS (:UTIL:INFOS)

This directory has macros for the INFOS II file management system. INFOS II is needed by the CEO Electronic Office (full product), and by the AOS COBOL language. INFOS II is included with AOS systems. This INFOS directory is created when INFOS II is installed on the hard disk. Some INFOS files are installed in other directories (: and :UTIL).

Filename (pathname)	Description
SORT (:UTIL:SORT)	This is the directory for Sort/Merge macros. Sort is needed by CEO. It also allows you to sort and edit records — but requires some computer experience to run. Sort is included with AOS software. This directory is created when Sort/Merge software is installed on the hard disk. Most Sort *files* are installed in :UTIL (their names begin with "SORT").
HELP (:HELP)	This is the Help directory, which contains the text you see when you type `HELP/V command)`. It was created when AOS was installed on the hard disk.
PER (:PER or @)	This is the peripherals directory, created by AOS each time the system comes up. PER has an entry file for each device. CON is a communications line, printer, plotter, or terminal. DPM and DPN are diskettes and disks. The @ is convenient shorthand for PER; for example, @DPM0 is easier to type than :PER:DPM0.
UPD (:UPD)	This is the user profile directory, shipped with AOS. UPD contains a profile file for each user.
NET (:NET and :NET:UTIL)	This is the network directory, created by AOS. If your system has XODIAC networking, its X.25 and agent (VTA and RMA) program files are copied here when XODIAC is installed on the hard disk.
Other system directories (PROC, QUEUE)	The PROC directory — created and managed by AOS — helps AOS keep track of processes. Directory QUEUE, created by EXEC, contains the queue file. Don't store files in either :PROC or :QUEUE.
UDD (:UDD)	This is the user directory directory, created when AOS was installed, with directory OP. It also contains a directory for each user who has a profile created with the PROFILE macro. The user directory (:UDD:username) becomes the user's working directory when the user logs on. Within it, the user can create files (including directories).

CEO Electronic Office File Structure

A system running the CEO Electronic Office (full product) has the CEO files shown in Figure 5-2, in addition to the files shown in Figure 5-1. CEO Word Processor - Independent files are described in the *Managing CEO® Word Processing-Independent* manual.

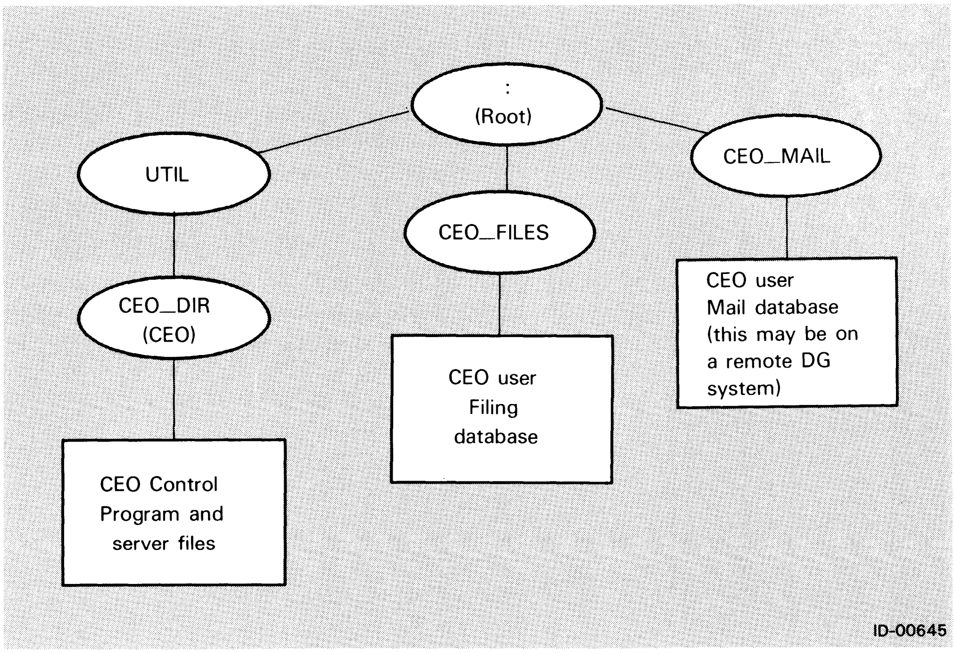


Figure 5-2 CEO Electronic Office files

CEO Directory (pathname)	**Description**
CEO_DIR (:UTIL:CEO_DIR)	This directory contains all the CEO program files, CLI macros, and other CEO support software. This directory is created when CEO is installed on the hard disk.
CEO_FILES (:CEO_FILES)	This directory exists only for the full CEO Electronic Office — not for CEO Word Processing - Independent. It holds all user CEO file system databases and profiles. CEO creates it during installation. The CEO user file system profiles are created by someone using a CEO profile creation menu.
CEO_MAIL (:CEO_MAIL)	This directory — which exists only for the full CEO, not the CEO Word Processor - Independent, serves the same functions for CEO Mail as CEO_FILES does for Filing. CEO creates it during installation. If you specify a remote configuration when you configure CEO, it tries to use the :CEO_MAIL directory on the remote system, not the :CEO_MAIL on your desktop system.

Directory Hierarchy and Pathnames

The file structure of AOS allows up to eight directory levels. Via directory pathname, any file is accessible from any other. The root (:) is always the highest directory. Usually, the most commonly accessed directories on the next level are :UTIL (for utility and other programs) and :UDD (for user directories). A common third-level directory is :UDD:username. A directory structure six levels deep is shown in Figure 5-3.

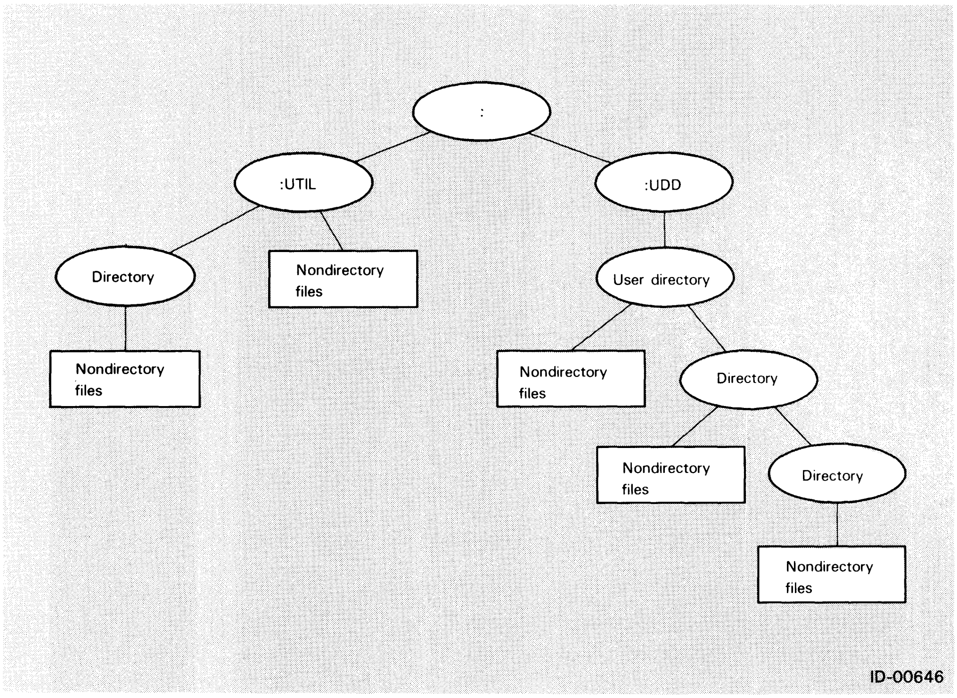


Figure 5-3 An AOS directory hierarchy

For CEO users, directories have other names. In the full CEO product, each user has a personal *cabinet*. In either CEO, the user can create *drawers* — and within drawers, *folders*. The folders can contain *documents*.

After AOS and other DG software are installed, people can use the system with practically *no* knowledge of file structure. The ability to create and use directories is handy for CLI users (shown in action in Chapter 4).

But some understanding of file structure is important for the person who installs AOS and other DG products on the hard disk, and/or acts as system manager or operator. That is why we explain file structure here.

Filenames and Templates

Filenames can be up to 31 letters, numbers, or special characters: ?, $, _ (underscore), and . (period).

You can use a template character (+, -, or *) to match part of a filename. Templates are explained in Chapter 4.

File Access Control

In a multiuser system, users need free access to, and protection for, their own files. System and user files need protection from accidental or malicious deletion. AOS user directories — and the UNPROTECT and PROTECT macros — help control access to files.

Each user "owns" the files in his or her user directory and subordinate directories; and no one else can read them, change them, or even list their names. The files are *protected* from access by other users.

Any user can unprotect any or all his or her files at will, using the UNPROTECT macro. This is useful for sharing text or other files. For example, assume user Joan creates a text file named MONTHLY_REPORT, in her user directory (:UDD:JOAN). User Fred (and other users) can't read this file. For example:

```
) WHO ↲
PID: n  FRED  n  :CLI.PR

) TYPE  :UDD:JOAN:MONTHLY_REPORT ↲
WARNING: FILE ACCESS DENIED, FILE :UDD:JOAN:MONTHLY_REPORT
)
```

If Joan wants other people to have access, she can use the UNPROTECT macro. For example, she could type

```
) UNPROTECT  MONTHLY_REPORT ↲
```

And Fred (and other users) could then read it:

```
) WHO ↲
PID: n  FRED  n  :CLI.PR
```

```
) TYPE :UDD:JOAN:MONTHLY_REPORT ↲
.
. (text of Joan's file)
```

The UNPROTECT macro gives all other users ownership access to the file, which means they can read *or* delete it. To restore the original ownership access, in which no one can access the file, use the PROTECT macro.

A user can unprotect *only* files he or she owns — normally, files in and below the user's directory. (Fred can't unprotect Joan's files.) This doesn't apply to a Superuser CLI. The Superuser privilege, activated by the command `SUPERUSER ON↲`, bypasses all these controls. The master CLI, with Superuser on, can protect or unprotect any file in the system. Actually, it can read, execute, or delete any file on the system regardless of the file's protect status.

DG ships most files protected, and owned by user OP. This gives the master CLI, when it is running at the system console, the power to use the system files without turning Superuser on. Public files, like :UTIL and its contents, are shipped with read-only access. Any user can read, or execute them, but not delete them.

Commands, Macros, and Programs

This section describes common CLI commands, and macros and programs supplied with AOS for desktop systems.

About CLI Commands and Macros

The CLI has over 100 commands built into it — and it is shipped with more than 20 macros (disk files that contain one or more CLI commands). You can do a lot of work using only a few commands and macros.

You can get a display of all CLI commands (but not macros) by typing `HELP *COMMANDS↲`. For details on any command, type `HELP/V commandname↲`. The CLI lets you abbreviate any CLI command to the fewest characters that identify it; for example, TY for TYPE. You can't abbreviate *macro names* (but you can omit the .CLI), or program names (but you can omit the .PR).

CLI Errors and Error Messages

When the CLI cannot execute a command, it displays an explanatory message called an *error message*. Then it stops and waits for another command. Generally, CLI errors are okay; everyone makes them, and usually all you need do next is retype the command correctly.

CLI error messages begin with the word ***WARNING***, ***ERROR***, or ***ABORT***. The text that follows this word describes the error; for example

```
WARNING: FILE DOES NOT EXIST, FILE filename
```

If the explanatory text allows you to understand what went wrong, you can correct it and continue with your desired operation. If you *cannot* understand what went wrong, look up the error text in Chapter 14, the error chapter. This chapter tells you how to cope with the most common, important errors — not only with CLI errors, but errors that occur in other programs too.

Multiple Commands and Long Command Lines

To stack more than one CLI command or macro on a line, separate the items with a semicolon; for example:

```
) TYPE  MYFILE;  WRITE  MYFILE ↲
```

To continue a command onto another line, type an ampersand and press ↲ (&↲). The CLI will then display an ampersand-prompt on the next line and you can continue typing. When you want the CLI to execute the command, press ↲ without typing an ampersand. For example:

```
) TY&↲
&) PE&↲
&) □ MYFILE↲
```

Extra spaces or tabs within CLI command lines have no effect. But neither a space nor a tab is allowed before a switch (for example, DELETE/V MYFILE).

About EXEC

The Executive program (EXEC) enables the system to serve multiple users in an orderly way. EXEC and a companion program named XLPT help manage the printer and printer queue. And, in systems with more than one terminal, EXEC manages user logon and logoff. EXEC also handles other things, like batch processing.

EXEC is started by the UP macro and terminated by the DOWN macro. It runs automatically, and you rarely need to pay attention to it.

Sometimes, though, you will need to type control commands to EXEC; for example, to stop the printer. In desktop systems, there are CLI macros to control EXEC. Each macro name begins with "PRINTER". For example, to stop the printer, type `PRINTER_STOP)`. EXEC commands themselves are described in *How to Generate and Run AOS.*

If EXEC cannot obey a command, it will display an error message. EXEC messages begin with the phrase *FROM PID n :(EXEC)*. The text that follows this phrase describes the error; for example

```
FROM PID 3 (EXEC): FILE DOES NOT EXIST
```

As with the CLI, if the message text allows you to understand what went wrong, correct it and continue with your desired operation. If you *cannot* understand what went wrong, look up the error text in Chapter 14, the error chapter.

Command, Macro, and Program Summary

This section lists the common commands, macros (macro filenames end in .CLI), and programs (program filenames end in .PR). Each description explains what the item does, the format for using it, the reasons why you might use it, and then shows an example. Table 5-1 summarizes the commands, macros, and programs. The more commonly used switches are listed with each command. Switches allow you to build and individualize each command. For a full list of switches, refer to each command in the *Command Line Interpreter (CLI) User's Manual.*

Table 5-1 AOS commands, macros, and programs for DESKTOP GENERATION systems (continues)

Command, Macro, or Program Name	**What It Does**
BROADCAST.CLI	Sends a message to all system users
BYE	Logs you off a user terminal or shuts down the system
CLOSE.CLI	Closes an OPENed diskette before you remove it from its slot
CONFIGURE.CLI	Identifies system devices
COPY	Copies to a file one or more files
CREATEDIR.CLI	Creates a directory
CREATETEXT.CLI	Creates a file and inserts text in it
DATE	Displays or changes the system date
DELETE	Deletes one or more files
DFMTR.PR	Formats a diskette for use as an AOS directory
DIRECTORY	Displays or changes the working directory
DISABLE7BIT.CLI	For model 10/SP, enables the 8-bit character set
DOWN.CLI	Brings down the multiuser environment
ENABLE7BIT.CLI	For model 10/SP, enables the 7-bit character set
EXECUTE	Executes another program
FILESTATUS	Describes filenames and statistics
FIXUP.PR	Closes a diskette that was open at abnormal shutdown
FSTAT.CLI	Describes filenames and statistics, alphabetically
FULL_BACKUP.CLI	Copies files to diskette for backup
HELP	Gives help information on topics and CLI commands
INC_BACKUP.CLI	Copies files to diskette for backup — based on date

Table 5-1 AOS commands, macros, and programs for desktop systems (continued)

Command, Macro, or Program Name	What It Does
INSTALL.CLI	Starts installing a DG software product
LOGON.CLI	Runs a standard CLI process on the system console
MMOVE.PR	Copies one or more files to diskette for safekeeping
MOVE	Copies one or more files to a different directory
OPEN.CLI	Opens a formatted diskette as a directory
PERMANENCE	Displays or sets permanence for one or more files
PRINTER_ALIGN.CLI	Stops a printer so you can align paper, then restarts it
PRINTER_CONTINUE.CLI	Continues printing after a PRINTER_STOP
PRINTER_REDEFINE.CLI	Changes number of characters per line and lines per page
PRINTER_STOP.CLI	Stops a printer immediately
PROFILE.CLI	Creates, Deletes, or Renames a User Profile
PROTECT.CLI	Allows only you to access files that you own
QCANCEL	Cancels or aborts a printing job or batch job
QDISPLAY	Describes jobs in the batch and print queues
QPRINT	Places one or more files on the printer queue
RENAME	Renames a file

Table 5-1 AOS commands, macros, and programs for desktop systems (concluded)

Command, Macro, or Program Name	What It Does
RESTORE.CLI	Restores files from backup media to the hard disk
RUNTIME	Describes a process's life span and overhead
SETUP.CLI	Sets up the environment for each user at logon
SPACE	Displays the amount of disk space used and remaining
SUPERUSER	Turns on Superuser to bypass file access controls
TERMINATE	Kills a process
TIME	Displays or changes the system time
TYPE	Types one or more files on the terminal screen
UNPROTECT.CLI	Allows any user access to a file that you own
UP.CLI	Starts up printers, other DG products, and the multiuser environment
WHO	Describes a process
WHOS.CLI	Describes all processes on the system
WRITE	Displays arguments on the terminal or writes them to a file
XEQ	Executes another program

BROADCAST.CLI

Sends a message to all system users

BROADCAST message

BROADCAST.CLI is a macro provided by DG to simplify communication with system users. It uses the SEND command to send a message to all terminals.

Why Use It?

If there is more than one user on your system, you may want to send them all a message — for example, when you want all users to log off so that you can shut down.

Example

```
) WHOS ↵              (Check all users.)

PID: 2  OP   OP    :CLI.PR
PID: 3  OP   EXEC  :UTIL:EXEC.PR
.
.
.
PID: 13 SAM   013  :CLI.PR
PID: 15 JOAN  015  :UTIL:CEO_DIR:CEO_CP.PR
PID: 16 ALLEN  016  :CLI.PR

) BROADCAST System coming down in 2 minutes. Please log off. ↵

From PID 13: (SAM) System coming down in 2 minutes. Please log off.
```

... a minute passes...

BROADCAST.CLI (continued)

```
) WHOS ↲            (Check all users again.)

PID: 2   OP   OP      :CLI.PR
PID: 3   OP   EXEC    :UTIL:EXEC.PR
.
PID 13:   SAM   013   :CLI.PR

) BYE ↲             (Sign off your CLI.)
AOS CLI TERMINATING
```

You are now

```
PID: 2   OP   OP   CLI.PR

) DOWN ↲            (Start shutdown.)
```

In this example, Sam, the person using the system console, wants to shut down. He uses the WHOS macro to check all users, finds two others (JOAN and ALLEN), then uses BROADCAST to warn them about shutdown. Then he waits a minute or so for them to log off, checks again with WHOS, finds that they have logged off, and starts shutdown.

BYE

Logs you off a user terminal or shuts down the system

BYE

The BYE command — when you type it from the CLI on a user terminal — terminates the process that the system created when you logged on. This logs you off the system.

Typed from a CLI process on the system console, BYE↲ terminates that CLI. If the CLI is the master CLI (PID 2), BYE↲ starts the shutdown sequence.

Why Use It?

BYE is the preferred way to log off a user terminal; and it is the *only* way to shut down the system normally.

Examples

```
) WHO ↲
PID: 8   OP      008     :CLI.PR

) BYE ↲
AOS CLI TERMINATING ...

You are now
PID: 2   OP      OP     :CLI.PR

) BYE ↲
DO YOU REALLY WANT TO SHUT THE SYSTEM DOWN? Y↲
SYSTEM SHUTDOWN
```

The first example shows BYE signing off a user CLI process. The second example shows system shutdown (BYE↲ typed to PID 2, the master CLI.)

CLOSE.CLI
Closes an open diskette before you remove it from its slot

```
CLOSE     [diskette-directory-name]
```

The CLOSE macro makes it easy to close a diskette for access. You need to do this *only* for a diskette that was opened with the OPEN macro.

If you omit an argument, CLOSE closes the diskette you opened without a directory name. The diskette must be in the slot where it was opened.

However, if you opened a diskette using a directory name, you must use the *same* directory name as an argument to the CLOSE command.

If you have forgotten the directory name used to open the diskette, you can find it. To learn the directory name and close, type

`) DIR/I ↵`	(Initial user directory.)
`) FILES/TYPE=LNK ↵`	(Display link filenames.)
linkname	(System displays a linkname; for example, ..FLOPPY.SAL.DISK1 — where DISK1 is the ***diskette-directory-name***.)
`) CLOSE directory name ↵`	(CLOSE by displayed name; for example, DISK1)

These commands display the link name (and the directory name within it) and close the diskette. If nothing is displayed after you type the FILES command, this means you do not have a diskette open. Another user might have one open, but *you* do not. Anyone can check for an open diskette by typing DIR/I, then FILES/TYPE=LNK as above, and close (if desired) by typing CLOSE directory name as above.

Always move out of the diskette directory before you use the CLOSE macro. You need not be in the same directory where you opened the diskette.

NOTE *You'll get an ...ENVIRONMENT... error message if you close a diskette when it is the working directory. The system will close the diskette and print the error message; the working directory will then (usually) be your user directory, :UDD:username. To avoid this error condition, don't close a diskette when it is or contains the working directory (for example, type DIR/I ↵ before closing).*

diskettes before invoking the DOWN macro. If you forget to do so, the message FILE ALREADY EXISTS appears the next time you try to open the diskette. When this message appears, refer to the recovery procedure listed under "FILE ALREADY EXISTS" in Chapter 15.

You need not use CLOSE for backup diskettes accessed during FULL_BACKUP, INC_BACKUP, MMOVE, or RESTORE operations.

Why Use It?

After a diskette has been opened (OPEN macro) it must be closed (CLOSE macro) to maintain file integrity. If you remove an open diskette from its slot without closing it, you must run FIXUP on the diskette to make it usable again and you must bring the system down, run FIXUP on the hard disk, and bring the system up again. You can avoid running FIXUP by typing `CLOSE ↲` to a diskette before removing it from its slot.

Example

```
DIR ↲
:UDD:ALLEN;

) OPEN ↲
Diskette has been OPENed as directory :UDD:ALLEN:FLOPPY3

.
. (operations proceed)

) DIR/I ↲
) CLOSE ↲
)
```

Allen can remove the diskette from the primary unit.

CONFIGURE.CLI
Identifies system devices

```
CONFIGURE
```

CONFIGURE is a macro supplied with AOS to help you identify multiple communications lines, diskette units, a printer, a plotter, or modems connected to a USAM multiplexor device.

The AOS macro asks questions and uses the answers to build a file later used to bring the system up. Then, if your system includes XODIAC networking and/or CEO, it passes control to the XODIAC and/or CEO configure macros. If CEO Mail and Calendar are configured to run remotely, the CEO configuration macro brings up CEO to allow you to create CEO profiles. Ultimately, files are built for the UP macro to use as it starts the multiuser environment.

You can rerun CONFIGURE to change the configuration of

- AOS (for example, to add diskette unit or fix a mistake);
- XODIAC (for example, to change your local hostname or fix a mistake); or
- CEO (for example, to add a CEO user profile if CEO runs remotely).

You can skip AOS, XODIAC, or CEO configuration questions, if desired, to change another product's configuration. You can change *some* printer specifications with the PRINTER_REDEFINE macro.

After you run CONFIGURE, run the UP macro, which finishes building or rebuilding the multiuser environment, if needed, and brings everything up.

Only the master CLI, PID 2, can run CONFIGURE. The multiuser environment must be down (DOWN macro) before CONFIGURE can run.

Details on CONFIGURE appear in Chapter 2.

Why Use It?

Desktop computers can run a communications line, printer, plotter, and user terminals; they can also run CEO either remotely or locally. CONFIGURE is needed to tell the system, other software products, and the UP macro about devices connected to each line.

Examples

```
) COPY/V  MYFILE_ALL  MYFILE  MYFILE1  XXX:MYFILE2 ↵
MYFILE
MYFILE1
XXX:MYFILE2
```

This command creates MYFILE_ALL, then copies several other versions of MYFILE to it.

Put the source diskette into the primary unit. Do not open this diskette.

```
)COPY/V DFILE @DPMO ↵
@DPMO
```

Replace the source diskette with a destination diskette. Do not open this diskette either.

```
)COPY/V @PDMO DFILE ↵
DFILE
```

With these two commands, you can copy the contents of one diskette to another when you have only one diskette drive unit. The first command copies the contents of the source diskette to a disk file; the second command copies the contents of the disk file to the destination diskette. COPY copies a physical diskette with or without a file structure. You cannot type, read, or execute the "file" you copied from the diskette onto the hard disk. To make copies of individual files in their normally accessible form, open the diskette, make it into your working directory with the DIRECTORY command, and then move the desired file(s) to a directory on the hard disk.

Example

```
) CONFIGURE ↵

In the following questions the values in braces are the
valid choices for the response. The value in the
quotation marks is the default response.

Please type your system model number?
{10SP 20 30}  "10SP"    ↵

How many diskette units do you have?
{1 2} "1"    2 ↵

Do you want to reserve line 0 for communications?
{Y N} "N"    Y ↵

Is line 0 connected via a modem?
{Y N} "N"    ↵

What is the baud rate of line 0?
{300 1200 240 4800 9600} "4800"    ↵

How many printers do you have?
{0 1} "0"    1 ↵

What is the maximum number of characters you want printed per line?
{16 - 255} "80"    160 ↵

What is the maximum number of lines that can be printed per page?
{6 - 144} "66" ↵

How many plotters do you have?
{0 1} "0"    ↵

How many user terminals do you have?
{0 - 3} "0"    ↵

. (Here, if XODIAC networking is installed, XODIAC's configure macro
. asks one or more questions. And if CEO is installed,
. CEO's configure macro asks other questions.

System definition is complete...

(Person creates AOS and CEO profiles, if needed.)

) UP ↵

.
```

Here, the person specifies a Model 10/SP system configuration — with two diskette units, a communications line and a printer.

COPY

Copies to a file one or more files

`COPY destination-pathname source-pathname` *[source-pathname] [...]*

The COPY command copies the contents of the file(s) named in **source-pathname** to the file named in **destination-pathname**. If you specify two or more files, they are copied to the destination file in order you specify.

The copy commands creates the destination file if one doesn't exist. If the file named in **destination-pathname** already exists, use the /A or /D switch.

Template characters don't work with COPY pathnames: you must specify the entire name.

Why Use It?

COPY allows you to add text to a file or to build one large file from smaller ones. It also allows you to copy the contents of a diskette onto another diskette, or onto a hard disk; or copy the contents of a file on a hard disk onto a diskette. And, COPY can duplicate a file in the same directory. It copies the *contents* of a file, without the name, creation date, and so on. Therefore, it's preferable to the MOVE command when you want to copy a file under a new name.

Switches

/A Appends the **source-pathname(s)** to the end of existing file **destination-pathname**. Use this switch if you want to keep the old file and add text to it.

/D Deletes the **destination-pathname** and create a new one before doing the copy.

/V Verifies the filename(s) copied; useful when you copy many files into one large file.

CREATEDIR.CLI

Creates a directory

`CREATEDIR   directory-pathname` *[maxsize]*

The CREATEDIR.CLI macro creates the directory you specify in `directory-pathname`. To limit the directory's capacity, include a ***maxsize*** number; this limits the amount of information that the new directory will hold to ***maxsize*** disk blocks. A disk block holds 512 bytes (characters). If you omit ***maxsize*** , the directory size is limited only by total disk space. The new directory is created as a *control point* directory, which means that you can use the SPACE command to see how much space is being used in it.

(For people who want to learn more about the CLI, the CREATEDIR macro uses the CLI command CREATE.)

Why Use It?

Directories are a useful feature of AOS — and CREATEDIR allows you to create a directory.

Examples

```
) CREATEDIR  NEWDIR ↵
) CREATEDIR  TEST  50 ↵
```

Each CREATEDIR creates a new directory in the working directory. The first creates directory NEWDIR without a specific space limit; the second creates directory TEST with a capacity of 50 disk blocks. Since each block holds 512 bytes (characters), directory TEST can hold files that total more than 20,000 bytes.

CREATETEXT.CLI

Creates a file and inserts text in it

`CREATETEXT pathname`

The CREATETEXT macro creates the file named in `pathname` and inserts the text you type. The file cannot already exist — if you want to change an existing text file, you must either delete and recreate it with CREATETEXT, or use COPY/A to append another file to it.

After you type `CREATETEXT pathname↵`, the CLI displays two right parentheses *,))* , as a prompt. It writes each line you type into the file. To end each line, press ↵. To end the entire insert, press ↵ and type `)↵` next to the CLI *))* prompt. (The macro gives instructions.)

Why Use It?

CREATEXT allows you to create a file containing text — useful for macros and other shortcuts. If you did not acquire a text editor (CEO or SED) with your system, this is the *only* easy way to create a file of text. You can use the WRITE command to put one line of text at a time into a file.

Example

```
) CREATETEXT TODAY.CLI ↵
.
)) WRITE Today's files are ↵
)) F/AS/S/AFTER/TLM=[!DATE]  :UDD:[!USERNAME]:# ↵
)) ) ↵
)
```

This example shows creation of a macro: TODAY.CLI, which can be executed via `TODAY↵`. TODAY.CLI displays *`Today's files are`* followed by an alphabetical list of the filenames created or modified today, in and below any user's directory. `[!DATE]` belongs to a family of CLI constructs called *pseudo-macros.* Pseudo-macros are designed for CLI macros. Used within a command, they expand to a specific value. [!DATE] expands to the system date.

DATE

Displays or changes the system date

```
DATE
DATE  [dd-mon-yy]
```

The DATE command alone displays the current system date. The command with an argument changes the system date. The system date can be set only from the master CLI, PID 2. For dd, use the date digit (1 through 31), for mon, use the first three letters of the month (JAN, FEB,...), for yy, use the last two digits of the year; for example, 14-DEC-84.

Why Use It?

The DATE command is handy when you forget the date. And, it can be useful when you bring up the AOS system with the wrong date. It's very important that the system date be correct; you can use the DATE command to set it from the master CLI.

Examples

```
) DATE ↓
15-DEC-84
```

The system displays the date as December 15, 1984.

```
) DATE  14-DEC-84 ↓
ERROR: CALLER NOT PRIVILEGED FOR THIS ACTION

) WHO ↓
PID: 14  SAM  :CLI.PR

) BYE ↓              (To return to master CLI.)
AOS CLI TERMINATING ...

You are now
PID: 2  OP  OP  :CLI.PR

) DATE  14-DEC-84 ↓
)
```

This sequence shows someone trying to set the system date, getting an error message, and retrying from the master CLI.

DELETE

Deletes one or more files

```
DELETE  pathname [pathname] [...]
```

The DELETE command can delete both nondirectory and directory files. The system will not delete a directory that has subordinate directories, unless you use a pathname template character.

You can use template characters with DELETE.

To protect files from deletion, use the PERMANENCE command.

If you try to delete a directory that has subordinate directories, the system may reject the command with a *DIRECTORY DELETE ERROR* message. To delete a directory that has subordinate directories, use the # template; for example, DELETE/V #↵. This form also allows you to delete a user directory that has subdirectories: just get into :UDD, then type DELETE/V username:#↵.

CAUTION *be very careful with DELETE and the # template.*

Why Use It?

As people use a computer system, they create files, which accumulate and consume valuable disk space. Some of these files become obsolete or redundant. The DELETE command allows users to remove files from the disk. It also allows someone who is running the master CLI to delete unwanted user directories (see above). Periodically, someone should use DELETE to remove files that are no longer needed. In a multiuser system, individual users, who know their files, should be the ones to do this.

For you to delete a file that is not in or under your user directory, the file must be unprotected (UNPROTECT macro), or you must be running the master CLI (PID 2) with Superuser on.

Switches

/C Confirms. AOS will display each filename and wait. If you type Y↵, it will delete the file. If you type any other character, it will not delete the file.

/V Verifies. AOS displays the name of each deleted file.

Examples

```
) DELETE/V  FOO ↲
DELETED  FOO
) DEL/V  SUBDIR:FOO ↲
DELETED  SUBDIR:FOO
```

These commands deleted file FOO in the working directory and in directory SUBDIR.

```
) DEL/V/C  MY+ ↲
=MYFILE.1? Y ↲
DELETED MYFILE.1

=MYFILE? ↲
FILE NOT DELETED

=MYFILE.2? Y ↲
DELETED MYFILE.2
)
```

Here, the /C switch told the CLI to confirm before deleting (and /V verified as usual). The person chose to delete file MYFILE.1 (by typing Y↲), chose not to delete MYFILE (by typing a character other than Y↲) and chose to delete MYFILE.2 (by typing Y↲).

DFMTR.PR

Formats or renames a diskette for use as an AOS directory

```
XEQ DFMTR
```

DFMTR.PR is the Disk Formatter program that software formats diskettes.

The Disk Formatter software formats a diskette, making the diskette into an AOS directory. Once you software format the diskette, you should open the diskette using the OPEN macro. You can then access the diskette as you do other AOS directories. You can create files within it; you can move or copy files to and from it; or you can execute programs from it. (But before you remove the diskette from the unit, you must close it using the CLOSE macro.)

The Disk Formatter assumes that the diskette is in your primary unit (the right unit). It cannot format diskettes in the secondary unit (the left unit). However, after you format a diskette in the primary unit, you can move it to the secondary unit and open it there.

After you have formatted a diskette, you should label it (if it is not already labeled). You should include relevant information on the label such as the diskette directory name, your name, and the date the diskette was formatted. (Use a felt-tipped pen, to avoid scoring the surface.) Otherwise, it's easy to forget what's on the diskette.

If the Disk Formatter finds too many bad blocks on the diskette, the message *TOO MANY BAD BLOCKS* appears. Discard the diskette and try another.

If the Formatter says *HARD ERROR, DEVICE 000 020, ERROR:PHYSICAL UNIT FAILURE,* the diskette has not been hardware formatted. You should hardware format it as shown in Chapter 14, "Formatting and Copying Diskettes."

If the Formatter says *DISK AND FILE SYSTEM REVISION NUMBERS DON'T MATCH,* the diskette has not been software formatted. Rerun the Formatter (DFMTR.PR) and choose an ERASE format.

You need not run the Disk Formatter on any diskette to be used for backup (using the FULL_BACKUP or INC_BACKUP macros).

Disk Formatter Dialog

After the Disk Formatter starts up, it asks if you want to save information, the name you want to call the diskette, and if you want the diskette to be protected. The questions are as follows:

```
AOS DISKETTE FORMATTER (DFMTR) REV x.xx

DO YOU WANT TO SAVE ANY INFORMATION ON YOUR DISKETTE?
TYPE Y (YES) TO SAVE INFORMATION, OR
   N (NO) TO ERASE YOUR DISKETTE.
```

If the diskette has information on it that you want (for example, if it has already been software formatted), type `Y ↓`. The diskette surface check for the SAVE operation takes about 50 seconds.

If the diskette is new or if you'd like to clean it off and start again, type `N ↓`. You must type `N ↓` to format a new diskette as an AOS directory. The ERASE surface check takes about 3 minutes. If you type `N ↓`, the Formatter asks for confirmation:

```
THIS WILL ERASE ALL FILES FROM YOUR DISKETTE.
TYPE Y (YES) IF YOU ARE SURE YOU WANT TO DO THIS, OR
TYPE ANYTHING ELSE TO START OVER.
```

It does this to make sure that you don't accidentally erase the diskette. To confirm, type `Y ↓`. To start again, type `N ↓`.

```
MOUNT DISKETTE TO FORMAT, AND TYPE NEW LINE WHEN READY.
```

If the diskette is not in the primary unit, insert it. Press ↓.

```
DISKETTE NAME (1 TO 31 CHARS) [xxx]
```

This question allows you to name (or rename) the diskette. The name you enter will be the directory name after the diskette is opened. You can use up to 31 characters: A through Z, 0 through 9, ? (question mark), $ (dollar sign), _ (underscore), and . (period).

The item displayed in brackets ([ ]) is the old name of the diskette. In an ERASE format there is no previous name, and you must type one. If you can, use a name that describes the material that will be on the diskette (for example, `MEMOS ↓`). In a SAVE format, you can either keep the old name by pressing ↓, or type a new name to rename the diskette.

```
DO YOU WANT THIS DISKETTE TO BE PROTECTED?
TYPE Y FOR YES, OR N FOR NO.
```

You can answer Y or N to this question. A protected diskette allows only a person with your username to access its files (when OPEN). Anyone can access unprotected files. Since you can always take the protection off individual files, a good answer to this question is Y ↵.

In a SAVE format, you also have the option to leave it as it was (*...NEW LINE TO LEAVE IT UNCHANGED*). To keep the old protect status, press ↵.

The Formatter now checks the disk surface for bad blocks — disk areas that won't hold information.

```
ANALYZING YOUR DISK FOR BAD BLOCKS
NUMBER OF OPERATIONS REMAINING: n
```

(*n* starts at 89 in an ERASE format, or 29 in a SAVE format.)

When *n* reaches 0, the Formatter says

```
TOTAL BAD BLOCKS: n

DONE!
nnnnnn
!
```

If the Disk Formatter displays

```
A BAD BLOCK WAS ALLOCATED, MUST RUN FIXUP
```

you must run FIXUP. After the Formatter terminates, type XEQ FIXUP ↵. FIXUP will try and find a path around the bad block(s). You may lose one or more files in the process. Make note of any *FILES MAY BE MISSING* messages from FIXUP, and check the diskette for missing files after opening it.

Why Use It?

You may want to use a diskette as a directory, for storage, or to transport files to another DG system. You may want to run programs from a diskette. Before you can use diskettes in any of the ways mentioned, you must software format them with the Disk Formatter.

Again, using the Save rather than the Erase option, you can use the Disk Formatter to rename a diskette or to protect it.

If you receive a *HARD ERROR* message about a diskette, your diskette could have developed a new bad block. You can salvage at least part (maybe all) of the files on the diskette by running a SAVE format on the diskette (type `Y ↵` to the first question). Then, if the Formatter says *MUST RUN FIXUP*, run FIXUP (`XEQ FIXUP.PR ↵`). FIXUP isolates the bad block and tries to work around it. You may then be able to open your diskette as usual. The bad block may mean the diskette is wearing out, however. To protect yourself from this possibility, you should copy it to another diskette (described in Chapter 14, "Formatting and Copying Diskettes"). Then discard the old diskette.

You can remove or store the diskette; you can use it to transfer files to some other system; or you can leave it in the primary unit to use as an AOS directory. After a diskette is open, it remains an accessible directory until it is closed or the system shuts down.

You do not need the Disk Formatter to prepare diskettes for backup using FULL_BACKUP, INC_BACKUP, the MMOVE program, or for diskettes that you use with an operating system other than AOS (such as CP/M®-86 or MS™-DOS, for example).

Example

The following example shows a diskette used to store a specific directory. First, the user (Sally) checks the directory to see if it will fit on a 368-Mbyte diskette; it will. Then, she runs the Disk Formatter on a diskette. She names the diskette JUNE83.FILES.SAVE. When she finishes formatting the diskette, she opens it, moves all the files from her disk directory (JUNE.83.FILES) to the diskette directory (JUNE83.FILES.SAVE), and sends a listing of files to the printer.

Sally uses FSTAT to check that the files were moved, after which she closes the diskette and deletes the disk directory she copied. If she needs any deleted files later, she can move them back from the diskette. In this way, she frees up space on her hard disk.

```
) SPACE JUNE84 ↵                     (From the CLI, check space
                                     used by directory)

MAX 1000   CUR 660   REM 380         (660 blocks; this will fit on
                                     the diskette. A 368-Kbyte
                                     diskette holds approxi-
                                     mately 700 blocks.)
```

Sally puts a diskette in the right (primary) unit. She will copy the directory onto this diskette.

```
) X DFMTR ↲

AOS DISKETTE FORMATTER (DFMTR) REV x.xx

DO YOU WANT TO SAVE ANY INFORMATION ON YOUR DISKETTE
TYPE Y (YES) TO SAVE INFORMATION, OR
   N (NO) TO ERASE YOUR DISKETTE.     N ↲

THIS WILL ERASE ALL FILES FROM YOUR DISKETTE.
TYPE Y (YES) IF YOU ARE SURE YOU WANT TO DO THIS, OR
TYPE ANYTHING ELSE TO START OVER.     Y ↲

MOUNT DISKETTE TO FORMAT, AND TYPE NEW LINE WHEN READY. ↲

DISKETTE NAME (1 TO 31 CHARS) [ ]    JUNE83.FILES.SAVE ↲

DO YOU WANT THIS DISKETTE TO BE PROTECTED?
TYPE Y FOR YES, N FOR NO.    N ↲

ANALYZING YOUR DISK FOR BAD BLOCKS
NUMBER OF OPERATIONS REMAINING: n

   (3 minutes pass)

TOTAL BAD BLOCKS: 0

DONE!

) OPEN ↲
Diskette has been OPENed as
:UDD:SALLY:JUNE83.FILES.SAVE              (Diskette pathname)

) MOVE/V/L=@LPT JUNE83.FILES.SAVE JUNE.83.FILES:# ↲   (Move the directory)
   .
   . (Listing of files to printer)

) SPACE JUNE83.FILES.SAVE ↲          (Check space remaining on the diskette)
MAX 689, CURR 668, REM 21

) FSTAT JUNE83.FILES.SAVE:# ↲
   .
   . (Verify the files moved to the diskette)
   .

) CLOSE ↲

) DELETE/V JUNE.83.FILES:# ↲   (Delete the copied directory from
                                the hard disk)
```

DIRECTORY

Displays or changes the working directory

DIRECTORY *[directory-pathname]*

Without switches and arguments, the DIRECTORY command displays the working directory pathname.

If you include ***directory-pathname*** , DIRECTORY makes this the working directory. If ***directory-pathname*** is beneath the working directory, the pathname can start in the working directory. But if ***directory-pathname*** is above the working directory, you must use the ^ specifier (press SHIFT and 6 keys) or use a full pathname from the root (:).

Why Use It?

Generally, it's easier to work within the directory that holds the files of interest than to type long pathnames to these files. The DIRECTORY command allows this.

Switches

/I Sets the working directory to the initial user directory, or to the ***directory-pathname*** below the initial directory.

Examples

```
) DIRECTORY ↵
:UDD:OP
) DIR  REPORTS ↵
) DIR ↵
:UDD:OP:REPORTS
) DIR/I ↵
) DIR ↵
UDD:OP
```

These DIRECTORY commands display the working directory name, make a subordinate directory (REPORTS) the working directory, and finally make the initial (user) directory the working directory (DIR/I).

DISABLE7BIT.CLI

Disables 7-bit mode, restoring 8-bit mode (Model 10/SP systems only)

```
DISABLE7BIT
```

The DISABLE7BIT macro disables 7-bit mode and the restores the standard 8-bit character handling. If the emulator is already in 8-bit mode, the macro does nothing.

For an explanation of 7- and 8-bit mode, see ENABLE7BIT.CLI in this chapter.

Why Use It?

This macro restores normal, 8-bit character handling after you select 7-bit character handling with the ENABLE7BIT macro. 8-bit mode is the standard mode, which should be in effect if you are not running user programs that require the 7-bit character set. You can use the DISABLE7BIT macro anytime you want to make sure the emulator is in 8-bit mode. (AOS starts up with the emulator in 8-bit mode, so macro isn't needed at startup.)

Example

```
) DISABLE7BIT ↲
```

Also see the example under ENABLE7BIT.

DOWN.CLI

Brings down the multiuser environment

DOWN

DOWN.CLI is a DG-supplied CLI macro that eases the task of bringing the multiuser environment down. Among other things, DOWN.CLI terminates EXEC — which will terminate all user processes that are sons of EXEC. Before terminating EXEC, DOWN tries to shut down CEO, INFOS II, and XODIAC (in that order), if your system has them.

Users in CEO, BASIC, or the SED text editor might lose work if terminated without warning. If you think there may be other users on the system, use the WHOS and BROADCAST macro to give everyone a chance to log off. Then return to the master CLI (if not there) and type **DOWN↵**.

Only the master CLI process (PID 2) can run the DOWN macro.

Why Use It?

Typing **DOWN↵** prepares for shutdown by shutting down DG products like XODIAC, INFOS, CEO, and EXEC.

Example

```
) DOWN ↵
Error - Only the master CLI can run DOWN.

) WHO ↵
PID: 13  OP  013  :CLI.PR

) BYE ↵
AOS CLI TERMINATING

You are now
PID: 2  OP  OP  :CLI.PR              (The master identifies itself.)

) DOWN ↵
.

Bringing down CEO...
... (CEO messages)...
```

DOWN.CLI (continued)

```
Bringing down INFOS...
... (INFOS II messages)...

Bringing down XODIAC...
... (XODIAC messages)...

Bringing down EXEC...

DOWN processing complete at  time  on  date

.

↲

) BYE ↲
DO YOU REALLY WANT TO SHUT THE SYSTEM DOWN?    Y ↲

SYSTEM SHUTDOWN
```

Here, the person at the system console tries to run DOWN, but the macro will not run since the PID is not 2. The person then signs off the CLI, returning to the master CLI, PID 2. DOWN then works, and system shutdown follows.

ENABLE7BIT.CLI

Enables 7-bit mode, providing 7-bit character handling (Model 10/SP systems only)

```
ENABLE7BIT
```

The ENABLE7BIT macro disables standard 8-bit mode and provides 7-bit mode (7-bit character handling) on Model 10/SP system consoles that run the optional 7/8-bit D200 terminal emulator. If the emulator is already in 7-bit mode, the macro does nothing.

On non-U.S. keyboards, some codes display different characters in 7-bit mode than they do in 8-bit mode. For example, on a Swedish terminal in 7-bit mode, the code 135 generates an angstrom A (an A with a circle over it). But on the same terminal in 8-bit mode, code 135 generates a bracket.

This means that if

- you have a Model 10/SP system with a non-U.S. keyboard; and
- you have application programs that use any of your language's special character codes; and
- you run the emulator in 8-bit mode;

the code may not display as the character you expect. This is also true if you input the character from the keyboard. You can check this — if you want — by trying the emulator in 8-bit mode. If certain characters don't display the way you want them to, just change emulator mode using the ENABLE7BIT.CLI macro (type **ENABLE7BIT ↵**). You can re-enable 8-bit mode at any time with the DISABLE7BIT.CLI macro (type **DISABLE7BIT ↵**). These macros are supplied with Model 10/SP systems in AOS (in directory :UTIL).

Why Use It?

In some customer sites, user programs employ codes in the 7-bit character set to print special non-U.S. characters. The codes these programs use will not print as expected if your system uses the (default) 8-bit character set. By selecting 7-bit mode, the ENABLE7BIT macro allows such codes to print as usual.

In 7-bit mode, certain characters (like square brackets and commercial at signs) have special meanings. Thus, you should always re-enable 8-bit mode after executing the user program(s) that require the 7-bit character set.

AOS starts up with the emulator in 8-bit mode.

Example

```
! 26H                           (Start up system...)
.
AOS CLI REV n date time
) UP ↵                          (Bring up multiuser environment.)
) ENABLE7BIT ↵                  (Enable 7-bit mode.)
.                               (Run in 7-bit mode.)
) DISABLE7BIT ↵                 (Disable 7-bit mode,) returning to
                                8-bit.)
```

EXECUTE

Executes a program

```
EXECUTE  pathname [argument]
```

The EXECUTE command executes the program file named in **pathname**. (The XEQ command works the same way and is slightly easier to use since it can be abbreviated to one letter.) A program file is a file that has been built into an executable program; its name always ends in .PR. You can omit .PR from the pathname.

The optional argument typically is the name of a file you want the program to find and process. Generally, you supply an argument when you execute system programs like the optional SED text editor (Chapter 9).

Why Use It?

Often, the kind of action you want from the system requires executing another program — which you can do via EXECUTE or XEQ. For many of the programs supplied with AOS, macros effect the command for you, so you need not type the EXECUTE command.

Example

```
) EXECUTE  SED  BOOMER.F77 ↲
Do you want BOOMER.F77 to be created?   Yes ↲
.
* . (SED editor commands)

* BYE ↲
)
```

In this example, you execute a text editor to write a FORTRAN 77 source program.

FILESTATUS

Describes file names and statistics

FILESTATUS *[pathname] [pathname] [...]*

The FILESTATUS command displays information on files in any directory. If you omit a *pathname*, it describes all files in the working directory.

Templates are allowed (and extremely useful) in pathname arguments. For a listing of files outside the working directory, you can specify the directory pathname (for example, :UDD:JACK:+) or you can use the caret (SHIFT and 6 keys) to specify a parent directory (^+).

Why Use It?

FILESTATUS (or the FSTAT macro) is your primary source of information on files in an AOS system. Within the CLI, you'll probably use it more often than any other command.

Switches

/AFTER/TLM=dd-mon-yy

Includes all files created or modified on or after day dd in month mon in year yy. For dd, use a one- or two-digit number; for mon, use the first three letters of the month's name; for yy, use a two-digit number. For example, AFTER/TLM=14-DEC-84.

/ASSORTMENT

Includes an assortment of information: filename, type, date and time created, and size in bytes (characters).

/BEFORE/TLM=dd-mmm-yy

Includes files created or modified before the given day. Has same form as /AFTER.

/L=@LPT
/L=pathname

Writes pathnames to the printer (/@LPT) or the file named in pathname instead of to the terminal.

FILESTATUS (continued)

/S — Sort the filenames alphabetically.

/TYPE=CPD — Includes only control point directory files (which are created by CREATEDIR).

/TYPE=DIR — Includes only "standard" directory files (created by CREATE command).

/TYPE=LDU — Includes only logical disk unit (diskette) names. After you open a diskette, the F/TYPE=LDU command will display the diskette name (in the directory where it was opened.)

Examples

```
) FILES  TEMP+ ↵
```

This command lists on the terminal the name of every file whose name begins with TEMP in the working directory.

```
) F/AS/S  +.FOO  XDIR:+.FOO ↵
```

Within the working directory and subordinate directory XDIR, this displays an assortment of information about all files whose names end in .FOO. It sorts the filenames alphabetically for each directory.

```
) FILES/S/AS/TYPE=CPD ↵
```

This displays names and other assorted information on all directories (the control point type) within the working directory.

```
) F/AS/S/L=ALL_MY_FILES # ↵
```

This command gets names of all files in the working directory and in each subordinate directory, sorted alphabetically in each directory. It writes all the names to file ALL_MY_FILES. For an example with date switches, see CREATETEXT.

FIXUP.PR

Closes a diskette that was not closed normally with CLOSE

```
XEQ  FIXUP
```

When a disk is open at abnormal shutdown, or when it is removed from a unit, it must be closed before it can be accessed again. The FIXUP disk fixer does this. The FIXUP.PR program fixes diskettes; the other program, :FIXUP, fixes the hard disk.

You need run FIXUP.PR only when you see the message *CAN'T INIT LD, RUN FIXUP OVER IT*. This occurs only after a diskette was open (OPEN macro) at the abnormal shutdown, when you tried to OPEN the diskette again.

An abnormal shutdown leaves open both the hard disk and any open diskettes. Therefore, after an abnormal shutdown, you'll run the disk-fixing FIXUP (so you can restart AOS) before you can run the diskette-fixing FIXUP from AOS.

FIXUP works only on the diskette in the primary (rightmost) unit. If a diskette was in the secondary (left) unit (if any) at the abnormal shutdown, transfer it to the rightmost unit, then run FIXUP on it.

If FIXUP says ***This LDU was not in use, fixing is not necessary* , this means that the diskette wasn't open at the shutdown. FIXUP will give you the chance to cancel the request. You can do so by typing Y ↵.

While FIXUP runs, note any *FILES MAY BE MISSING* messages; if you see one or more of these, find the message in Chapter 16 to recover.

Why Use It?

After an abnormal shutdown with an diskette that was open, the diskette stays open. You cannot open it or access it until someone has run FIXUP to close it.

Also, if the Disk Formatter says *MUST RUN FIXUP* , FIXUP is needed to work around a new bad block that has developed on the diskette.

FIXUP.PR (continued)

Example

Power fails, then returns.

```
!   26H

*** WARNING -- DISK IS IN USE. MUST RUN FIXUP ON THE LDU ***
PROGRAM NAME?   FIXUP ↵

REQUEST 1 (DPNO) FIXING LDU 'xxx' NOW ...
.
. (Runs on hard disk)
.
REPAIR IN DIRECTORY :XDIR:ABC.... .
DONE!

!   26H
PROGRAM NAME?   AOS ↵
.
INITIALIZING SYSTEM DATA ON DISK
. (bring up AOS)
.
) UP ↵
) LOGON SALLY ↵          (Person runs LOGON before opening diskette.)
.
) OPEN ↵          (Try to open primary diskette.)

ERROR: CAN'T INIT LD, RUN FIXUP OVER IT.

) XEQ  FIXUP ↵

Request 1 (DPMO) fixing LDU 'MY__FLOPPY' now ...
.
. (FIXUP runs on diskette)
.
DONE!

) OPEN ↵          (Try again to open diskette.)
MY__FLOPPY          (It works.)
)
```

This sequence shows the events after a power fail — which produces an abnormal shutdown that can't be turned into a normal shutdown. While trying to restart AOS, the person is advised to run FIXUP. After FIXUP runs, he or she restarts AOS. Next, trying to open the diskette, the person is advised to run FIXUP again, executes FIXUP, then opens the diskette.

FSTAT.CLI

Describes filenames and statistics, alphabetically

```
FSTAT [pathname] [pathname] [...]
```

FSTAT is a CLI macro, designed for ease of use, that does a FILESTATUS command with an assortment of information and column headers.

Why Use It?

Typing FSTAT is easier than typing FILES with other switches, and FSTAT displays a header. And it is easier than creating your own macro.

Switches

/DIR Displays the names of directory files only (types CPD and DIR).

Examples

```
) FSTAT  +.FOO  XDIR:+.FOO ↲
```

```
) FSTAT/DIR ↲
```

```
) FSTAT/L=ALL_MY_FILES # ↲
```

These command lines do exactly what their counterparts with FILESTATUS do — see the example explanation under FILESTATUS for more information.

FULL_BACKUP.CLI

Copies files to diskette for backup

```
FULL_BACKUP
```

FULL_BACKUP.CLI is a CLI macro designed to help you copy disk-based material for safekeeping. It copies *all* user files in and below the working directory to one or more diskettes.

For an incremental backup, which copies *only* user files created or changed since the last backup, use INC_BACKUP. Macro FULL_BACKUP creates a file called LAST_BACKUP that contains the current date; INC_BACKUP uses this when it does an incremental dump. Since INC_BACKUP needs the file created by by FULL_BACKUP, someone must do a full backup (perhaps a few days after AOS is installed on the hard disk), before INC_BACKUP will work. If you need to *restore* material from diskette, use RESTORE.

The diskette(s) you use for backup need not have been formatted with the Disk Formatter.

Always back up from the same directory. Preferably, this will be the root directory (:), from the master CLI (PID 2). If you back up from the root, all user files (including CEO user files) will be backed up, and can be restored easily and simply. It can be extremely complex to restore using backups started from different directories.

INFOS_II and CEO processes (if you have either product on your system) cannot be running when you back up the whole hard disk. If you see an error message about *server processes*, you'll probably need to bring the multiuser environment down (via the DOWN↵ macro) before the system will let you start the backup.

If the material to back up requires more than one diskette, the terminal will prompt for another diskette after the current one fills up. Later on, if your system needs to restore material, the backup diskettes must be inserted *in the order* that they were originally filled. If the order is wrong, the restoration won't work. Therefore, we suggest that you note the sequence number and date on the diskette label (apply a label if there isn't one). To write on a diskette label, use a felt-tipped pen to avoid scoring the diskette surface.

All the backup macros — FULL_BACKUP, INC_BACKUP, and RESTORE — use a program named MMOVE.PR. This program was supplied to you with the system.

NOTE *The full and incremental dump macros assume that your system doesn't have a tape unit. If you do have a tape unit, it uses 15-Mbyte cartridges — allowing one cartridge to hold the contents of a hard disk. You may prefer tape to diskettes for backup. If so, don't use the backup macros — instead, use the DUMP command. Later, to restore material, use the LOAD command. DUMP and LOAD are described in the CLI manual. The device name of the tape unit is @MTCO or @TAPE.*

Why Use It?

It's prudent to back up your disk-based material periodically — in case someone accidentally deletes valuable files, or in case material is lost for any reason.

Depending on how many files change each day, you might choose a full backup every month or so, and an incremental backup each week or on alternate days. This way — if somehow files are lost — your site will lose relatively little work.

Switches

`/L=@LPT` Lists directory and filenames backed up to the printer
`/L=pathname` (`/L=@LPT`) or to the file `pathname`.

Examples

Prepare for a full backup by inserting a diskette in the primary diskette unit.

```
) DIR : ↵
) FULL_BACKUP ↵

ERROR: CALLER NOT PRIVILEGED FOR THIS ACTION
SUPERUSER,ON

) WHO ↵
PID: 13  OP  OP  :CLI.PR          (Not PID 2.)

) BYE ↵          (Sign off this CLI.)
AOS CLI TERMINATING ...

You are now
PID: 2  OP  OP  :CLI.PR          (PID 2 — the master.)
```

FULL_BACKUP.CLI (continued)

```
) FULL_BACKUP/L==FILES_BACKED_UP ↵
.
** Full backup from directory : at  16:06:54  on  06-JAN-84.**
.
Press NEW-LINE when ready to begin backup. ↵

Beginning file backup - a list of files dumped to diskette follows.
.
Disk 1 is full. Mount next disk and type NEWLINE to continue.
```

(Replace diskette with a fresh one.)

```
.
.
Disk 14 is full. Mount next disk and type NEWLINE to continue.

** Full backup of directory : complete at 16:41:38 **
.
)
```

In this example, the system will not start the backup because the process's PID is not 2. The person signs off the CLI to return to its father CLI, which is PID 2. The full backup then proceeds, and consumes 14 diskettes. A list of the files backed up goes to file FILES_BACKED_UP, where it's available for examination (via printing or the TYPE command) if needed. When it's no longer needed (perhaps at the next FULL_BACKUP), the listing file can be deleted to save space.

In the next example, CHRIS does a full backup from a user directory.

Chris inserts a diskette in the rightmost unit.

```
) DIR ↵
:UDD:CHRIS

) FULL_BACKUP ↵
.
** Full backup from directory :UDD:CHRIS : at  17:15:38  on  06-JAN-84 **
.
Press NEW-LINE when ready to begin backup. ↵

** Full backup of directory :UDD:CHRIS complete at  17:45:02 **
.
)
```

This copies all files in and below the user directory. Since a user has owner access to all these files, any user can do this from his or her user directory.

HELP
Gives help information

```
          [ *topic    ][ *topic    ]
HELP [/V] [ *command  ][ *command  ]
          [ letter(s) ][ letter(s) ]
```

The format you use depends on the kind of help you want, as follows:

- HELP with no arguments displays a list of CLI topics. Some topics (like FILENAMES) may be familiar to you; other topics involve aspects of the CLI that we don't cover in this book. For more information on these topics, see the *AOS and AOS/VS User's Handbook* or the *Command Line Interpreter (CLI) User's Manual.*
- For help on a specific topic, type `HELP *topic` ↵; for example, `HELP *MACROS` ↵.
- If you type `HELP *COMMAND(S)` ↵, AOS displays a list of all CLI commands.
- For help on any individual command, type `HELP command` ↵. For example, `HELP BYE` ↵
- For a list of CLI commands that begin with a certain letter or combination of letters, type `HELP letter(s)` ↵; for example, `HELP H` ↵.

Some help files are longer than one screen and will scroll by quickly unless you freeze the display. Enter `CTRL-S` to freeze the display and `CTRL-Q` to resume scrolling.

Why Use It?

HELP can be extremely useful when you can't remember a command name, what a command does, or what switches a command has.

Switches

/V Gives a more detailed (verbose) help message.

Examples

```
) HELP ↓
.
(list of topics)
) HELP  *COMMANDS ↓
(list of CLI commands)
.
) HE/V  FILES ↓
.
. (FILESTATUS description)
.
```

INC_BACKUP.CLI

Copies files to diskette for backup — based on date

INC_BACKUP

INC_BACKUP.CLI is a CLI macro designed to help you copy disk-based material for safekeeping. It copies material in and below the working directory to one or more diskettes. For an incremental backup, it copies *only* the files that were created or modified since the last backup (either full or incremental). To discover the date of the last backup, it uses file LAST_BACKUP. When done, it updates LAST_BACKUP for the *next* incremental backup. You can type (TYPE command) file LAST_BACKUP to learn the date on which a directory was last backed up.

For a full backup, which copies *all* material in and below the working directory, use FULL_BACKUP. Someone *must* do a full backup at least once from a given directory before an INC_BACKUP will work from that directory. If you need to *restore* material from diskette, use RESTORE.

The diskette(s) you use for backup need not have been formatted with the Disk Formatter.

For an incremental backup of the whole hard disk, the working directory must be : (the root) when you type INC_BACKUP↵. If the working directory is :, the macro will try to turn Superuser on — which will work only from the master CLI, PID 2, on the system console.

INFOS II and CEO processes (if you have them) cannot be running when you back up the whole hard disk. If you see an error message about *server processes*, you'll probably need to bring the multiuser environment down (via the DOWN↵ macro) before the system will let you start the backup.

If the material to copy requires more than one diskette, the terminal will prompt for another diskette after the current one fills up. Later on, if your site needs to restore material, the backup diskettes must be inserted *in the order* that they were originally filled. If the order is wrong, the restoration won't work. Therefore, we suggest that you apply a label (if the diskette doesn't have one), and that you note the sequence number and date on the label. To write on a diskette label, use a felt-tipped pen to avoid scoring the diskette surface.

All the backup macros — INC_BACKUP, FULL_BACKUP, and RESTORE — execute a program named MMOVE.PR. This program was supplied along with the system.

Why Use It?

It's prudent to back up your disk-based material periodically — in case someone accidentally deletes valuable files, or in case material is lost for any reason. And incremental backups can save a lot of time — copying only files that were created or changed since the last backup.

Depending on how many files change each day, you might choose a full backup each month or so, and an incremental backup each week, or on alternate days.

To have incremental backups work properly for any directory, you must start them from the same directory as the full backups were started from (for example, the root [:]). Otherwise, the INC_BACKUP macro cannot determine the last backup date (left by the last backup macro run).

Switches

`/L=@LPT`
`/L=pathname` Lists directory and filenames backed up to the printer (`/L=@LPT`) or to the file `pathname`.

Example

Prepare for backup by inserting a diskette in the primary diskette unit.

```
) DIR  : ↵

) WHO ↵
PID: 2  OP  OP  :CLI.PR              (PID 2 — the master.)

) INC_BACKUP/L==INC_FILES_BACKED_UP ↵
.
** Incremental backup from directory : at  10:52:03  on  16-JAN-84 **
-This backup will dump all files created or modified since  06-JAN-84 - 17:00:38
Please insert the first ...
.
Press NEW-LINE when ready to begin incremental backup.  ↵
```

INC_BACKUP.CLI (continued)

```
Beginning file backup - a list of files dumped to diskette follows.
.
Disk 1 is full. Mount next disk and type NEWLINE to continue.
```

(Person mounts next diskette and presses ↵.)

```
** Incremental backup of directory : complete at  11:20:12 **
.
)
```

For more examples, see Chapter 7.

INSTALL.CLI

Starts installing a DG software product

```
INSTALL  product
```

The INSTALL macro is designed to simplify the installation of software products like INFOS, XODIAC, SORT, BASIC, or CEO. After installing a product, you need not do it again. But you may *choose* to do it again if you receive a new revision of the product.

Some popular product names, arranged alphabetically follow in Table 5-2.

Table 5-2 Names to install DG software by (continues)

Product	Name on Product Diskette Label(s)	Name to Install By
AOS	AOS PREGEN	AOS
BASIC (Business)	AOS BUSINESS BASIC PREGEN	BBASIC
BASIC (Extended)	AOS EXTENDED BASIC PREGEN	XBASIC
BASIC (MP/BASIC)	AOS MP/BASIC HFP	MBASIC
CEO (full product)	AOS CEO	CEO
CEO (Word Processor - Independent	AOS CEO.WP	CEO.WP
CEO Spelling	AOS CEO SPELLING	CEO.SPELLING
CEO COMPUCALC	AOS CEO COMPUCALC	CEO_COMPUCALC
COBOL (AOS)	AOS COBOL	COBOL
COBOL (Interactive)	AOS COBOL	ICOBOL

INSTALL.CLI (continued)

Table 5-2 Names to install DG software by (concluded)

Product	Name on Product Diskette Label(s)	Name to Install By
COMPUCALC	AOS COMPUCALC	COMPUCALC
DG/GATE software	AOS DG/GATE	DG_GATE
DG/XAP	AOS DG/XAP	DG_XAP
FORTRAN IV	AOS FORTRAN IV	F4
FORTRAN 5	AOS FORTRAN 5	F5
FORTRAN 77	AOS FORTRAN 77	F77
General Language Development Package (SED, LINK, SWAT and other programs)	AOS GLDP	GLDP
GKS (Graphical Kernel System)	AOS GKS	GKS
INFOS II	AOS INFOS II RUNTIMES	INFOS
PL/I	AOS PL/1	PL1
PRESENT®	AOS PRESENT	PRESENT
Sort/Merge	SORT/MERGE RUNTIMES	SORT
SP/Pascal	AOS SP/PASCAL	PASCAL
Terminal Emulator (Model 10/SP)	AOS D200 EMULATOR	EMULATOR
TRENDVIEW	AOS TRENDVIEW	TRENDVIEW
XODIAC	AOS XODIAC PREGEN	XODIAC

Only the master CLI, PID 2, can run INSTALL. Before you start INSTALL, be sure to insert the product diskette in your primary diskette unit. If there is more than one product diskette, be sure to insert the *first* one; the system will prompt for the next diskette(s) as needed.

Why Use It?

INSTALL eases the installation of DG software.

Examples

Insert DG-supplied product diskette in the primary unit.

```
) WHO ↵
PID: 2  OP  OP  :CLI.PR

) INSTALL  MBASIC ↵

Please insert the diskette with the desired DG-supplied MBASIC software
into your primary diskette unit - @DPM0.

Press NEW-LINE when you are ready to begin installation  ↵
.
. (time passes as MP/BASIC files are copied to the hard disk)
.
MBASIC Software installation is complete.
Please remove diskette from the primary unit.
)
```

The product software is now installed in its own directory (usually :UTIL:product, except for XODIAC, which is installed in directory :NET). For example, for FORTRAN 77 (name F77), the directory is :UTIL:F77.

LOGON.CLI

Run a standard CLI process on the system console

```
LOGON [username]
```

The LOGON macro starts a standard CLI process with the username of *username*. A profile for the *username* must have been built with the PROFILE macro. If you omit *username* , the macro asks for it.

The CLI process created by LOGON starts in the user directory, :UDD:username.

When the LOGON-created CLI terminates (either by the BYE↵ command or a CTRL-key interrupt), the master CLI will return to the system console.

Only the master CLI — PID 2 — can run LOGON.

Why Use It?

When AOS comes up, it runs the master CLI, PID 2, on the system console. This CLI can turn Superuser on, trespass on other people's files, or bring the system down. The LOGON macro runs a standard CLI, without special privileges, giving the system console the appearance of a user terminal. It also starts in your user directory, which is the appropriate place to work.

Also, any process with the username OP cannot use CEO Mail or Calendar (if these are on a remote central host system). The master CLI has the username OP, which may prevent it from using CEO. The LOGON macro can run a CLI with a username other than OP — allowing remote CEO to work normally.

Example

```
AOS CLI  REV n  date  time
) UP ↵                      (Bring the environment up.)
.
. (EXEC and other product messages.)

) WHO ↵
PID: 2  OP  OP  :CLI.PR     (This is PID 2, the master CLI.)

) LOGON  HARRY ↵            (LOGON readies the system console
                            for user HARRY.)
```

```
AOS CLI  REV n  date  time

) WHO ↵
PID: 13  HARRY  013  :CLI.PR (The new CLI is not PID 2.)

) DIR ↵
:UDD:HARRY                   (The new directory is :UDD:HARRY.)

) CEO ↵                      (Harry starts CEO.)
```

This LOGON sequence readies the system console — after the multiuser environment is brought up — for Harry, who may be the only user. Harry then starts CEO.

MMOVE.PR

Copies one or more files to diskette for safekeeping

```
XEQ  MMOVE/DUMP [/FROM] diskette-unitname [source-pathname] [...]
```

The MMOVE program was supplied as an aid to backup and recovery. The macros FULL_BACKUP, INC_BACKUP, and RESTORE all use it. Since these macros are easier to use than MMOVE, we suggest you use them instead. But since MMOVE is part of the AOS software for desktop systems, we describe it here.

Always use the /DUMP switch with MMOVE and specify the diskette unit name. For example, **MMOVE/DUMP @DPM0** copies all files from the working directory to the diskette at DPM0. If you add the */FROM* switch, the program copies all files from the specified diskette to your working directory with the directory structure intact.

If you omit a *source-pathname* , MMOVE copies *all* files (or all those selected by date switches) in the working directory. This includes directories. If you include *source-pathname(s)*, MMOVE copies only the pathnames specified. Templates characters are allowed in these pathnames.

As with the backup macros, if the material to be copied requires another diskette, MMOVE will prompt for it. If you ever need to restore files, the diskettes must be inserted in the original order, so you should keep track of the diskette sequence.

Why Use It?

MMOVE is needed by the backup and recover macros: FULL_BACKUP, INC_BACKUP, and RESTORE. It is also used by the INSTALL macro.

Switches

/DELETE	Pertains with /FROM only. If a file on the hard disk has the same pathname as a file on diskette, deletes the file on the hard disk and replaces it with the file from diskette (to do this based on date, see /RECENT).
/DUMP	Is an essential switch; always use it.
/FROM	Copies files from diskette into the working directory. If you omit this, it copies from directory to diskette.

/L=@LPT
/L=pathname — Lists pathnames copied to the printer (/LPT) or to the file named in **pathname**. You must include the /V switch with L=pathname.

/RECENT — Pertains with **/FROM** only. If a file on the diskette has a more recent creation date than a file with the same name on the hard disk, it deletes the file on the hard disk and replaces it with the newer one.

/V — Verifies names copied to the terminal (or, with /L, to file **pathname**.)

You can also use the **/AFTER** and **/BEFORE** **date/time** switches described under FILESTATUS.

Examples

```
) XEQ  MMOVE/DUMP/L=@LPT  @DPM0  MYFILE+  NEWDIR:+ ↵
```

This command copies to the diskette in unit DPM0 all the files in the working directory whose names begin with MYFILE, and directory NEWDIR with all its filenames. It lists all filenames to the printer.

```
) XEQ  MMOVE/DUMP/FROM/RECENT/L=@LPT  @DPM0  MYFILE+  NEWDIR:+ ↵
```

This variation of the previous command restores backed-up files to disk. Assuming the working directory is the same for both MMOVEs, the second MMOVE copies back to hard disk the directory structure originally copied. If directory NEWDIR doesn't exist, MMOVE creates and copies all NEWDIR files from diskette. If NEWDIR does exist on the hard disk, MMOVE leaves it as is — except that if any files on the diskette are newer, MMOVE will delete them and replace them. All filenames restored are listed on the printer.

MOVE

Copies one or more files to a different directory

MOVE destination-pathname *[source-pathname] [source-pathname][...]*

The MOVE command moves a copy of one or more files to the directory named in **destination-pathname**. The *source-pathname(s)* must start in or below the working directory. Templates are allowed with *source-pathnames*.

MOVE can copy both nondirectory and directory files. If you omit *source-pathnames*, AOS copies the entire directory structure, from the working directory down, to the destination directory. This may waste a lot of disk space through file duplication; do it only when you really *want* to duplicate the structure.

The destination directory must be in or under your user directory, or it must be unprotected (UNPROTECT macro), or your CLI must have Superuser on (PID 2 only).

You must be *in* the directory you want to move files from. To move a file into the parent (superior) directory, you can either use the caret (^) or type the full pathname. To move a file into an an immediately subordinate directory, use just the directory name. To move a file into a parallel directory, you can either specify the whole pathname or use ^ followed by the directory name. For example,

```
) F MYFILE XDIR ↲
DIRECTORY :UDD:ALAN:MYDIR           (Working directory is
MYFILE XDIR                         :UDD:ALAN:MYDIR.)
) MOVE/V ^MYFILE ↲                  (Move MYFILE into superior
MYFILE                              directory, ALAN)
) MOVE/V XDIR MYFILE ↲              (Move MYFILE into subordi-
                                    nate directory, XDIR)
MYFILE
) MOVE/V ^APRIL MYFILE ↲            (Move MYFILE into a
MYFILE                              parallel directory called APRIL)
```

CAUTION *Do not move and delete from the original directory, any of the files supplied by Data General. AOS and its support programs expect certain files to be in specific directories. The system may not run properly if you delete any of them.*

Why Use It?

From time to time, you may want to move one or more files from one directory to another; for example, to reorganize your files. You could use COPY for this, but MOVE retains the original name, creation date, and so on; thus serves better to identify the file. Also, MOVE allows template characters and COPY doesn't.

To conserve disk space, you might want to delete the files from the original directory after the move (otherwise, you'll have duplicate files in different directories).

To copy files to diskette for safekeeping, we recommend that you use FULL_BACKUP or INC_BACKUP, not MOVE. If you use MOVE, the diskette must have been formatted with the Disk Formatter (DFMTR.PR), in this chapter; you must open it (OPEN macro) before use, and close it it (CLOSE macro) afterwards.

Switches

/AFTER/TLM=dd-mon-yy — Moves all files created or modified on or after the day dd. See /AFTER in FILESTATUS for more.

/BEFORE/TLM=dd-mon-yy — Moves all files created or modified before the day dd. See /BEFORE in FILESTATUS for more.

/L=@LPT
/L=pathname — Lists pathnames moved to the printer (/@LPT) or the file named in pathname; for /L=pathname, you must also use /V.

/RECENT — If a *source-pathname* is more recent than a file with the same name in destination-pathname, it deletes the older file and replaces it with the newer.

/V — Verifies names moved to the terminal (or, with /L, to file pathname).

Examples

```
) MOVE/V  NEWDIR  XFILE+ ↵
. (system verifies)
.

) DELETE/V  XFILE+ ↵
. (system verifies)
```

This sequence moves all files whose names begin with XFILE to directory NEWDIR. The DELETE command cleans up the source directory and eliminates duplication.

OPEN.CLI

Opens a formatted diskette as a directory

OPEN

```
         @DPM0
         @DISKETTE 1
         @RIGHT_DISKETTE
OPEN     @DPM1              [ diskette-directory-name ]
         @DISKETTE 2
         @LEFT_DISKETTE
```

The OPEN macro was supplied to make it easy to open a formatted diskette for access as a directory. You can do this *only* for a diskette that was formatted with the Disk Formatter. It opens the diskette as a directory, with the name given to the diskette via the Disk Formatter.

Without arguments, OPEN opens the diskette in the rightmost unit (the only diskette, if you have just one). The rightmost diskette unit is the primary unit and has the formal devicename @DPMO.

If you attempt to open a diskette that has not been software formatted, the message *DISK AND FILE SYSTEM REVISION NUMBERS DON'T MATCH* appears. If the diskette has not been hardware formatted, the message *HARD ERROR, DEVICE 000 020 ERROR: PHYSICAL UNIT FAILURE* appears. Follow the instructions in Chapter 14, "Formatting and Copying Diskettes", to hardware and/or software format your diskette.

With arguments, OPEN opens either the primary diskette (names *@DPMO, @DISKETTE1, @RIGHT_DISKETTE*) or secondary diskette (names *@DPM1, @DISKETTE2, @LEFT_DISKETTE*). If you include the *diskette-directory-name*, the macro checks this name against the name given to the diskette with the Disk Formatter. The macro warns you if the names don't match. You can use arguments either to open the secondary diskette *or* if you must open a specific diskette, not just any formatted diskette. (for example, to make sure the diskette opened is JAN_DATA).

If you open without *diskette-directory-name* , you can close a diskette in either unit by using CLOSE without an argument.

If you use `diskette-directory-name` to open a diskette, you *must* use this name as an argument when you CLOSE the diskette. If you forget this name, you can find it through the linkname (described in the CLOSE macro).

If one diskette is already open and you want to open a second diskette, you must use `diskette-directory-name`. The system won't let you use OPEN without an argument when one diskette is already open.

If you try to open a second diskette without using a *diskette-directory-name*, you get the following message:

```
ERROR-You have already opened a diskette without a diskette directory name. You
must either close the first diskette or open this one with a diskette directory
name.
```

When this message appears, you can either close the first diskette before you open the second one, or include the diskette directory name of the second diskette with the OPEN command.

If you want two diskettes open at the same time, we suggest using `diskette-directory-name` to open BOTH diskettes. For example,

```
OPEN  @DPM0  AUGUST_REPORTS ↲
OPEN  @DPM1  SEPTEMBER_REPORTS ↲
```

Then, later, to close the diskettes, you'd type

```
CLOSE  AUGUST_REPORTS ↲
CLOSE  SEPTEMBER_REPORTS ↲
```

The formal names of the diskette units are @DPM0 and @DPM1. You can use any of the other names — for example, DISKETTE2 — if the multiuser environment is up. The UP macro creates files that allow you to use the other names.

When you open a diskette, it becomes a directory added to the working directory. It has the filename given to it by the Disk Formatter when it was formatted. While the diskette is open, it's just like any directory: you can DIR into it, list files in it, create directories in it, move files to and from it, and so on. For example, assume a diskette was formatted with the name MY_FLOPPY; and assume you open it from directory :UDD:ALLEN. The diskette directory pathname becomes :UDD:ALLEN:MY_FLOPPY; and you can list all files in it by typing

```
FSTAT  MY_FLOPPY:+ ↲
```

OPEN.CLI (continued)

CAUTION *After you open a diskette, do not remove it from its slot until you have closed it (CLOSE macro). If you shut down the system before you close the diskette, you get the following message the next time you try to access the diskette: ERROR: FILE ALREADY EXISTS. When this message appears, refer to the recovery procedure listed under the FILE ALREADY EXISTS error message in Chapter 15.*

If you move files onto a diskette, you may want to note the filenames on the diskette label (apply the label first, to the seamless side of the inner envelope, if needed). Use only a felt-tipped pen to write on a diskette label, to avoid damaging the diskette surface. Do not write-protect a diskette that you intend to open as a directory.

To open a diskette in any directory, you need owner and write access to the directory. This is a problem only when the working directory is not in your user directory (for example, the root). If you get a *WRITE ACCESS DENIED* message, check the working directory; and DIR back to your user directory if needed.

If you get the message *ERROR: INITIALIZATION PRIVILEGE DENIED* , you do not have write access to your diskette. If you have Superuser privileges, you can turn Superuser on and open the diskette. If you do not have Superuser privileges, you can run the software formatter (DFMTR.PR), and indicate that you want to SAVE your files. In this way, you remove the protection from the diskette.

You don't need to (and can't) open backup diskettes accessed for FULL_BACKUP, INC_BACKUP, MMOVE, or RESTORE operations.

Why Use It?

There may be times when you want to use a diskette as a directory, perhaps to transport files to another DG system. Or, when you want to put programs on diskette and run them from diskette. You need the OPEN macro to allow either of these things (after you have run the Disk Formatter on the diskette).

Examples

```
) DIR ↵
:UDD:SAM:XX

) OPEN ↵
Diskette has been opened as directory :UDD:SAM:XX:MARCH_FILES
```

```
) MOVE/V  MARCH_FILES MYPROG.PR MDATA ↲
MYPROG.PR
MDATA

) DIR  MARCH_FILES ↲
) XEQ  MYPROG ↲
.
. (MYPROG operations proceed on diskette)
.
) DIR/I ↲            (Make sure the diskette isn't the working directory.)
.
) CLOSE ↲
```

Remove the diskette from the primary unit.

The next example shows an OPEN with the wrong diskette name in the secondary unit. Note that you must use exactly the same diskette name as you did to open it — even if the name you used was not correct.

```
) DIR ↲
:UDD:OP

) OPEN  LEFT_DISKETTE  BASIC_PROGRAMS ↲

Warning - diskette name does not match the name you specified.
  Diskette FORTRAN_PROGRAMS has been OPENed as
  directory :UDD:OP:FORTRAN_PROGRAMS - if
  this is not the desired diskette, please
  CLOSE and retry with the proper diskette.

) CLOSE  FORTRAN_PROGRAMS ↲            (Must close before removing.)
```

Person removes diskette and replaces it with the correct one.

```
) OPEN  LEFT_DISKETTE  BASIC_PROGRAMS ↲

Diskette has been OPENed as directory :UDD:OP:BASIC_PROGRAMS
)
```

In this example, the operator (username OP) opens a diskette — that he or she thinks is named BASIC_PROGRAMS — in the secondary unit. The OPEN warns of the name conflict. The operator closes the diskette, replaces it with another, tries again, and the OPEN works without the warning message. All this could have been avoided if the diskette directory name had been written on the paper label — generally, a good idea.

PERMANENCE

Displays or sets permanence for one or more files

```
PERMANENCE  pathname  { ON  }
                      { OFF }
```

File permanence, if on, prevents any directory or nondirectory file from being deleted by any user. (But permanence will not save a file if its parent directory is nonpermanent and someone deletes the parent directory.)

If you omit both *ON* and *OFF* , PERMANENCE displays the permanence setting. If you include *ON* or *OFF* , it sets permanence on or off. You can use template characters with **pathname**.

Why Use It?

It's not hard to delete a file or directory accidentally — especially if you're casual with the DELETE command and template characters. If this happens, and the file hasn't been backed up, it is lost forever. Keeping PERMANENCE on prevents such unpleasant events.

Switches

/V Displays the filename along with the permanence attribute.

Example

```
) CREATEDIR  ZDIR ↲
) PERMANENCE/V  ZDIR ↲
ZDIR        OFF
) PERM  ZDIR  ON ↲
) DEL/V  ZDIR ↲
WARNING: CANNOT DELETE PERMANENT FILE, FILE ZDIR
) PERM  ZDIR  OFF ↲
) DEL/V  ZDIR ↲
DELETED  ZDIR
)
```

This command sequence creates a directory, checks its permanence setting (OFF), then turns permanence on and tries to delete the directory. The DELETE command fails, since the file is permanent; so permanence is turned off and the DELETE command works.

PRINTER_ALIGN.CLI

Stops the printer so you can align paper, then restarts it

```
PRINTER_ALIGN
```

The PRINTER_ALIGN macro tells EXEC to suspend printing — allowing you to align paper in the printer. Printing stops at the first line of the next even-numbered page.

The macro stops the printer (if printing), then asks if you want to reprint pages that have already been printed. If the paper is badly misaligned, or if it jammed and a lot of text overprinted, you may want to specify that the printer restart on a certain page.

The macro next waits for you to correct the problem and ready the printer. When you press NEWLINE, the macro restarts the printer.

Why Use It?

Printer paper may shift out of alignment or jam. PRINTER_ALIGN stops a printer so you can align paper or do other things with it, then resumes printing. If any pages before the current page have printed badly or not at all, you can specify the reprinting of these pages — instead of reprinting the whole file.

Example

...notice that printer-paper alignment is wrong...

```
) PRINTER_ALIGN ↲

FROM PID 3 : (EXEC) time

After you align the printer, do you want it to reprint any pages ...    Y ↲
```

Type the page number where the printer should restart. In the following example the printer will restart on page 4.

```
To reprint more than 32 pages requires extra time to return to the
requested page. Please type the number of pages you want to reprint... 4   ↲

When the printer is ready, press NEWLINE to continue.   ↲

FROM PID 3 : (EXEC) time
```

. (printer continues)
)

PRINTER_CONTINUE.CLI

Continues printing (the next file) after PRINTER_STOP

```
PRINTER_CONTINUE
```

The PRINTER_CONTINUE macro tells EXEC to continue operation of the printer. It will start with the next printing request.

Why Use It?

PRINTER_CONTINUE allows you to continue printer operation after the printer was stopped with PRINTER_STOP.

Examples

```
) PRINTER_CONTINUE ↵

FROM PID 3 : (EXEC)  @PRINTER CONTINUING

.
```

This tells the printer to resume normal operation. It will print the next file submitted.

PRINTER_REDEFINE.CLI

Changes printer columns per line and lines per page

```
PRINTER_REDEFINE
```

The PRINTER_REDEFINE macro tells EXEC to change the maximum number of characters per line and lines per page specified (by the CONFIGURE macro) for a printer.

The macro asks two questions, to which you supply answers:

```
What is the maximum number of characters you want printed per line?
{16 - 255} "80"
```

The answer you give here specifies a maximum number of characters per line. If any line exceeds the maximum, it will be truncated (cut off). Lines less than or equal to the maximum length will print normally (they will not be padded out by AOS). A model 4434 dot-matrix printer can print up to 160 characters in compressed mode, so, generally, for a 4434, specify **160↵**. A model 4518 letter-quality printer can print up to 203 characters (compressed); and a model 4433 printer can print up to 233 characters, compressed. Usually you should specify the maximum for your printer; for example, **160↵**. If this causes problems, try a smaller number like **90↵**.

The system default, if you don't give a number but press ↵, is 80 characters per line.

```
What is the maximum number of lines that can be printed per page?
{16 - 144} "66"
```

This determines the number of lines per page for printing. After this number of lines, the system will output a form feed, and start printing at the top of the next page. Generally, a printer can print 66 lines per page. (Most printers print 6 lines per inch, which, for 11-inch paper, comes to 66 lines per page.)

If your system runs CEO, you *must* specify the maximum number of lines a page can accept; if you specify a different number, users will not be able to print documents from CEO: when they try, the printer will print a *USER SPECIFIED FORM...* error message.

PRINTER_REDEFINE.CLI (continued)

The new answers you give to PRINTER_REDEFINE will not take effect until the next file is printed. The answers you give to PRINTER_REDEFINE are temporary. They remain effective only until you run PRINTER_REDEFINE again or use the DOWN macro. Always, the next time the system is brought up, the answers given to the CONFIGURE macro will be used.

Why Use It?

Sometimes, you might want to change the maximum allowed characters per line or lines per page (without CEO). The PRINTER_REDEFINE macro allows you to do this easily, without bringing the multiuser environment down and rerunning CONFIGURE. The Model 4434 printer has its own firmware program that you can use to change print characteristics. (It is described in your hardware *Operating* manual.)

Example

```
) PRINTER_REDEFINE ↵

What is the maximum number of characters you want printed per line?
{16 - 255} "80"    160 ↵

What is the maximum number of lines that can be printed per page?
{6 - 144} "66"  ↵
```

This dialog sets a limit of 160 columns per line and 66 lines per page for the printer.

PRINTER_STOP.CLI

Stops a printer immediately

PRINTER_STOP

The PRINTER_STOP macro tells EXEC to stop a printer immediately, and discard the rest of the current file (if any).

To restart the printer, use PRINTER_CONTINUE.

The printer will stop printing; if a file is currently being printed, it will end with a *TERMINATED BY OPERATOR* message.

Why Use It?

Sometimes you will want to stop the printer — for example, to

- stop printing a file that you don't want printed; or
- empty queues so you can shut down the system.

PRINTER_STOP is the easiest way to kill an active printing job. (Another way is for the user who QPRINTed the file to type `QCANCEL sequence-number↵`. The command `QDISPLAY↵` displays the current job's sequence number, under the heading of LPT.)

Example

```
) PRINTER_STOP ↵

FROM PID 3 : (EXEC) @PRINTER WILL PAUSE AT END OF CURRENT JOB
FROM PID 3 : (EXEC) @PRINTER FLUSHING CURRENT JOB
```

PROFILE.CLI

Creates, Deletes, or Renames a User Profile

PROFILE

The PROFILE.CLI macro was supplied to ease the process of creating AOS user profiles. The macro asks questions, then submits your answers to the profile editor, PREDITOR; then PREDITOR takes the appropriate action.

The PROFILE macro can be executed only from the master CLI, PID 2, on the system console. When you start it, it asks which act you prefer: create (C↵,) delete (D↵), or rename (R↵).

To create a new profile, type C↵; the macro then asks the following questions:

Please type the username -- 1 to 15 characters -- to create:

Answer with the name by which the system will know the user: 1 to 15 filename characters, no spaces allowed. The username you specify cannot already exist. If this user will use CEO, the AOS username and CEO User ID must be the same.

Please enter xxxx's password -- 3 to 15 characters.

Answer with the private password for the user: 3 to 15 filename characters, no spaces. If this person will use CEO with Mail and Calendar databases on a remote DG system, the passwords must be the same on both systems. If there are no CEO restrictions on the password, an easy method is to repeat the username, then have the user change the password as desired when he or she logs on. Changing a password is described near the end of Chapter 4.

The standard initial macro is SETUP.CLI.
Do you want to specify a different macro? {Y N} "N"

Your answer selects the macro to be executed for the user after logon. If you want the user to come up in CEO, type Y↵. Then, to have the user come up in the full CEO Electronic Office, type the CEO-supplied macro name, :UTIL:CEO.STARTUP.CLI↵. To have the user come up in the CEO Word Processor - Independent, type :UTIL:CEO.WP.STARTUP.CLI↵. Press ↵ to select the AOS-supplied macro name, SETUP.CLI. Any file you specify must exist; if not, SETUP.CLI will be used. When a user logs on, the standard SETUP.CLI will execute a SETUP.CLI in each user's directory if there is one. This allows each user to create his or her own setup procedure.

```
The standard initial program executed at logon is CLI.PR.
Do you want to specify a different program?  {Y N} "N"
```

This specifies the program to be executed for the user after logon. Generally, press ↵ for the CLI. The CLI also works for CEO users.

After you answer this question, the macro tells the profile editor to build the profile. The user (or you, to test things) can then log on. The username and password are required for a user terminal only. The system console is open to anyone unless someone has run the LOGON macro on it (see LOGON.CLI for more). There is more detail on the PROFILE Create questions in Chapter 2.

To delete a profile, type `PROFILE↵`, then `D↵`. The macro asks for a username to delete

```
Please type username to delete:
```

Type the username; for example, `JOAN↵`. The macro then deletes the profile:

```
Deleting JOAN's profile.....
The profile for JOAN has been deleted.

The user directory -- :UDD:JOAN -- has not been deleted.
You can delete it via the DELETE command if you so desire.
```

As the macro says, the user directory (and all its files) remain on the disk. To delete it, you must return to the master CLI (PID 2) on the system console; type `SUPERUSER ON↵`, then type `DELETE/V□:UDD:username:#↵`. Don't do this on impulse; be certain before you do it.

Renaming a user profile — PROFILE rename — removes access by the old username — so *don't do it while the user is logged on*, or the user will lose access to all his or her files. Also, if the user has a CEO profile and you rename the system profile, you must recreate the CEO profile, or the user won't be able to use CEO. (CEO profiles cannot be renamed.) If CEO runs remotely, this user will lose access to Mail and Calendar until the remote CEO and AOS profiles are changed to match the local one (this is a lot of work — it's better not to rename under these circumstances). Do not rename the OP profile: the operator profile must always have the username OP.

PROFILE.CLI (continued)

To rename a user profile, type PROFILE↵, then R↵. The macro prompts for the old and new names:

```
Please type the old username:
```

Answer with the old username; for example, GILLIAN↵.

The macro then checks for the existence of the old profile, then asks

```
Please type the new username:
```

Type the new username, 1 to 15 filename characters, no spaces; for example, JILL↵. The macro then confirms the rename:

```
Renaming user to JILL .....
Protecting all user JILL's files....

The user's old password still works. All files for JILL
are protected. JILL must unprotect the files for which
protection is not desired.
```

The user directory and profile have been renamed. The user can log on under the new name and old password. All files are protected (described under PROTECT.CLI).

Why Use It?

The AOS system is shipped with a profile and user directory that have the username OP. This allows one person to use the printer, and log onto a terminal other than the system console.

But the username OP cannot use CEO Mail and Calendar (if these are on a remote system). Thus, every system user should have a profile with a username other than OP. For remote access to work with CEO, a user must have a profile with the same username and password on both systems. Lastly, a profile gives each user a user directory, with access and protection for his or her own files. These are reasons to use PROFILE Create.

You might use PROFILE Delete when a user has left your organization, or if you want to remove his or her right to use the system.

You might use PROFILE Rename if someone were unhappy with his or her username. Before doing it, note the cautions above.

Example

```
) PROFILE ↲

            User Profile Options

          C Create a new user profile
          D Delete a user profile
          R Rename a user profile

      Type C or D or R -- based on your preference:     C ↲

Please type the username -- 1 to 15 characters -- to create:     JUSTINE ↲
Please enter JUSTINE's password (3 to 15 characters):     MYS ↲

The standard initial macro is SETUP.CLI.
Do you want to specify a different macro? {Y N} "N"  ↲

The standard initial program executed at logon is CLI.PR.
Do you want to specify a different program? {Y N} "N"  ↲

Starting to create the profile...

JUSTINE's profile has been created.
```

PROTECT.CLI

Allows only you access to a file that you own

```
PROTECT  pathname
```

Normally, all files in and below each user directory are protected. But if you choose to unprotect a file (perhaps so that another person can read it), this allows *all* users owner access: they can read, change, or even delete the file. The PROTECT macro restores the original protected status to files.

You can use template characters (for example, +) in the PROTECT macro.

You can protect only files that you own. Normally, these are files in and beneath your user directory.

Directory :UTIL and all its files are shipped with a limited universal access list, which allows all users to execute and read (but not change or delete) its files.

Why Use It?

Sometimes, after unprotecting a file, you may want to restore its original protected status. Use the PROTECT macro in such situations.

Examples

```
) PROTECT  MY+ ↓
MYMACRO.CLI
MYFILE
)
```

The PROTECT gives this user exclusive rights to MYMACRO.CLI and MYFILE.

QCANCEL

Cancels or aborts a printing job or batch job

QCANCEL sequence-number

The QCANCEL command cancels a printing or batch job that *you* submitted. It doesn't work with a job that *any other user* submitted. The job can be active (for example, the printer can be printing your file), or the job can be inactive (waiting in the queue).

To cancel a print job, include its sequence-number. This number is displayed when you type the QPRINT command. To check all current sequence-numbers, type QD↵ (QDISPLAY command).

Why Use It?

There will be times when you want to cancel a printing job — perhaps because it involves the wrong file. The QCANCEL command is the best way to do this. (The PRINTER_STOP command at the system console also works, but it's extreme.) And, QCANCEL is the only easy way to cancel a batch job.

Example

```
) QPRINT/COPIES=5  CHAPTER4 ↵
QUEUED, SEQ= 34, QPRI=127
.
```

...person decides to cancel, but doesn't remember the number (SEQ)...

```
QD ↵

BATCH_INPUT  BATCH  OPEN
.
.
LPT        PRINT  OPEN
  34     BYRON      :UDD:BYRON:CHAPTER4

) QCANCEL  34 ↵
)
```

In this example, a person starts printing a file, decides to cancel, checks the printing sequence number with QDISPLAY, and cancels.

QDISPLAY

Describes jobs in the batch and print queues

```
QDISPLAY
```

QDISPLAY describes queue usage, including the batch queues and — more useful to desktop system users — the printer queue.

Why Use It?

Often, you may want to know whether the printer is free, or how soon it will be free. Or, you may want to know the sequence number of a printing job so that you can cancel it (QCANCEL). QDISPLAY (QD) tells you these things.

Switches

/V Includes the estimated number of pages the file will produce. This number appears under the head LIMIT. The system considers 1,000 characters as a page and adds four pages for headers and trailers. Thus, it would estimate a file with 14,500 bytes as 14 pages and add 4 pages for a total of 18.

Example

```
) QD/V ↵

BATCH_INPUT    BATCH  OPEN
BATCH_OUTPUT    PRINT  OPEN
BATCH_LIST    PRINT  OPEN
LPT    PRINT  OPEN
SEQ# PRI  TIME  LIMIT FLGS USERNAME FORMS .
* 98  127 16:59:06    34     OP        .

FLAGS EXPLANATION:
* = ACTIVE
```

QPRINT

Places one or more files on the printer queue

```
QPRINT  pathname [pathname] [...]
```

QPRINT copies to the printer queue the file you specify in pathname.

For QPRINT to work, the printer must have been started on a queue and continued on a queue named LPT. These things are done by the UP macro that starts the multiuser environment. (In other words, QPRINT won't work unless the multiuser environment is running.)

Why Use It?

QPRINT (QPR) is the best way to print a file. It is the *only* way to print without tying up your terminal until the file is printed.

Switches

/COPIES=n	print n copies of the file.
/NOTIFY	notifies you (via a beep and message on your terminal) when the file has been printed.

Examples

```
) QPRI  FILE1 ↵
QUEUED, SEQ=92, QPRI=127
```

This command prints the contents of FILE1 on the printer.

RENAME

Renames a file

```
RENAME  pathname  new-filename
```

The RENAME command renames the file you specify in **pathname**, giving it the name **new-filename**.

CAUTION *Some DG products expect others to have specific names. If you want your system to function properly, do not rename any of the DG-supplied directories or files, unless the DG documentation or Release Notice advises you specifically to do so.*

Why Use It?

RENAME is handy when you want to give a file or directory a new, more descriptive name. (Often, as you work with your own files and directories, you'll think of better names for them.)

Examples

```
) RENAME  XXDIR  ARCHIVE_DIRECTORY ↲
```

This command changes the name of directory XXDIR in the working directory to ARCHIVE_DIRECTORY.

RESTORE.CLI
Restores files from backup media to the hard disk

```
RESTORE [pathname]
```

RESTORE is a macro that restores files that were copied to diskette for backup. The files must have been copied with the FULL_BACKUP or INC_BACKUP macros, or with the MMOVE program.

For the original file structure to be recreated correctly, the working directory for RESTORE *must be the same as the working directory for the backups* — both full and incremental. Often this will be the root directory (:). Only the master CLI, PID 2, can start RESTORE from the root directory.

If the original backup required more than one diskette (as is likely), the *first* diskette must be in the primary diskette unit. If you start with a wrong diskette, it will say *ILLEGAL DUMP FORMAT... Disk 1 must be mounted first;* find and insert the first diskette and retry RESTORE.

If, after the first diskette, you insert a diskette out of order, the terminal will display

```
Wrong disk. Please mount disk n and type NEWLINE to continue.
```

From the original group of diskettes, find the correct diskette (n) insert it, and press ⤶.

If your backup diskettes include both full and incremental backups, restore the last (most recent) *incremental backup* first. Then proceed backwards with the incremental backups until you reach the last full backup; then restore the last full backup. Only the most recent versions of all files will be copied — so that ultimately your hard disk will be restored to its state at the time of the last incremental backup. (The hard disk will include *all* files that existed during the backup period — this may include some files you don't want: for example, if a directory was cleaned up during this period, all the old, useless files will be restored, and must be deleted again.)

RESTORE.CLI (continued)

To restore a specific *file*, which can be a directory, include the *pathname*. The pathname you specify must be the full pathname as used for the backup. For example, if the original working directory was :, and you want to restore all files in user JACK's directory, you'd specify a pathname of `UDD:JACK:#` Or, to restore nondirectory files whose names start with FOO in JACK, you'd specify a pathname of `UDD:JACK:FOO+`. The system will read the diskettes sequentially, and display the name of each file restored. To restore an individual nondirectory file, try to specify the whole name (without a template), so that the system will stop after it finds and restores the file. (If you use a template character, it will want to run through all the diskettes, even after restoring the file you want. In this case, you can abort the restore with CTRL-C CTRL-B.)

All the backup macros — INC_BACKUP, FULL_BACKUP, and RESTORE — execute a program named MMOVE.PR. This program was supplied along with the system.

Note that when you restore a file, its time-last-modified is changed to the time at which you restore it. This means — if you restore many files, or a whole hard disk — that the next backup you do must be a full backup. An incremental backup won't work, since the original time-last-modified of many files has been changed.

RESTORE is shown in action in Chapter 7.

Why Use It?

RESTORE is the easiest way to restore one or more files from backup diskettes to the hard disk. It is also the easiest way to recreate your entire system file structure on a new hard disk.

Switches

`/L=@LPT`   Lists directory and filenames restored to the
`/L=pathname`   printer (`/L=@LPT`) or to the file pathname.

Example

The person mounts the first diskette of the last incremental backup in the primary unit.

```
) WHO ↓
PID: 2  OP  :CLI.PR            (Okay — the master CLI.)
```

```
) DIR  : ↲

) RESTORE ↲

.
** Restoration of directory : at  10:16:03  on  06-FEB-84 **

.
Press NEW-LINE when ready to begin restoration.   ↲

Beginning file restoration - a list of files restored from diskette follows.

.
Disk 1 is exhausted. Mount disk 2 and type NEWLINE to continue.
```

(Replace diskette with a the next one and press ↲.)

```
.
Disk 2 is exhausted. Mount disk 3 and type NEWLINE to continue.
```

(Replace diskette with the next one and press ↲.)

```
** Restoration of directory : complete at  10:22:31 **
)
```

The person now mounts the first diskette of the last full backup in the primary unit.

```
) RESTORE ↲

** Restoration of directory : at  11:00:00  on  06-FEB-84 **

.
Press NEW-LINE when ready to begin restoration.  ↲

Beginning file restoration -- a list of files restored from diskette follows.
Disk 1 is exhausted. Mount disk 2 and type NEWLINE to continue.
```

(Replace diskette with the next one and press ↲.)

```
.

.
Disk 14 is exhausted. Mount disk 15 and type NEWLINE to continue.
```

(Replace diskette with the next one and press ↲.)

```
** Restoration of directory : complete at  11:35:20 **
)
```

This example shows a restoration that involves one incremental backup and one full backup. The person makes sure the master CLI, PID 2, is running, then starts the restoration. The restoration proceeds through two incremental backup diskettes, then through 15 full backup diskettes.

RUNTIME

Describes a process's life span and overhead

RUNTIME *[pid]*

The RUNTIME command tells you how long a process has run *(ELAPSED)*, how much actual computing time it has used *(CPU)* , and how much information it has transferred into and out of memory, in 512-byte (character) blocks *(I/O BLOCKS)*. You can ignore *PAGE-MSECS* (an indicator of memory page usage per millisecond).

If you omit the *pid* (process identification) number, it displays statistics for your own CLI process.

Why Use It?

RUNTIME can tell you how much system overhead a process is using. Use the WHOS.CLI macro to list all PIDs, then type individual RUNTIME commands for each PID you care about. A process that is malfunctioning will use a lot of CPU time or I/O time. If, for every 10 seconds ELAPSED, a process uses 4 seconds or more CPU time, it may be malfunctioning. If system performance is poor, you may want to terminate this process and restart it. If the process involves a DG program, you might want to consult your DG support organization. If the process involves your own site's user program, you can terminate it: return to the master CLI (father of all processes) and type

TERMINATE n↵, where n is the PID of the offending process.

You can tell how long the system has been up by typing RUNTIME 1↵, (the PMGR's PID). This works for *any* user or system process.

Example

```
) RUN  1↵
ELAPSED 450:45:30, CPU  4:52:59.329, I/O BLOCKS 0, PAGE MSECS 416549615

) RUN↵
ELAPSED 9:25:44, CPU  0:00:02.030, I/O BLOCKS 30, PAGE MSECS 48605
```

SETUP.CLI

Sets up the environment for each user at logon

```
SETUP
```

The SETUP.CLI macro is supplied by DG to help desktop users access needed files. It is the default file (given in PROFILE) to execute for each user as he or she logs on.

Unless you want a user to start up in CEO, we suggest that you keep this as the default file to execute at logon. (The CEO file name is shown in the PROFILE macro.) If you specify a file other than SETUP.CLI or the CEO file, it must set the correct access control list and search list (described in the CLI manual).

If a file named SETUP.CLI exists in any user's initial directory (:UDD:username), this file will also be executed when the user logs on. This can be useful for greetings, general guidelines, and so on. The user's SETUP.CLI can be created either with the CREATETEXT macro or a text editor.

Why Use It?

Generally, people don't execute SETUP directly. The system executes it for them when they log on. The LOGON.CLI macro also executes it.

Example

Try typing SETUP (`TYPE SETUP.CLI↵`) for yourself.

SPACE

Displays the amount of disk space used and remaining

```
SPACE [directory-pathname]
```

The SPACE command tells you how much disk space you have (MAX), how much is occupied by files (CUR) and how much remains free (REM). In desktop systems, the MAX and REM figures are meaningful *only for the root directory (:)* and directories created with a maxsize figure. For other directories, MAX (and therefore REM) is a hypothetical figure (see examples).

If you get a *FILE IS NOT A CONTROL POINT DIRECTORY* error message from SPACE, this means that the directory was not created by the CREATEDIR macro (which creates all directories as control point directories). SPACE will not work with directories that are not control point directories.

If you omit arguments, SPACE describes the space usage in the working directory.

Each 15-Mbyte disk contains approximately 30,400 disk blocks, while each 38.6-Mbyte disk contains approximately 75,000 disk blocks. Each disk block can store 512 bytes (characters).

Why Use It?

If the hard disk becomes saturated with files, system performance will suffer. Eventually, if the disk fills up, all productive work will stop until someone frees some space by deleting files.

The SPACE command tells you how much space is left on the disk, and how much space each user is consuming. You can then use this information to decide when to clean up (by copying files to diskette, if needed, and deleting them). The FILESTATUS command or FSTAT macro can tell you the size of each file.

Examples

```
) SPACE  :↲
MAX 30442   CUR 10133   REM 20309

) SPACE  :UDD:OP ↲
MAX 2147483647   CUR 147   REM 2147483500

) CREATEDIR  FOO  100 ↲

) SPACE FOO ↲
MAX 100   CUR 0   REM 100
```

The first SPACE command describes the space remaining on the disk. The second shows how many blocks are currently used in :UDD:OP (the MAX and REM figures are meaningless). After the CREATEDIR macro creates a directory with a specific maximum size, the last SPACE command shows that the directory is empty.

SUPERUSER

Turns on Superuser to bypass file access controls

```
                       { ON  }
SUPERUSER              {     }
                       { OFF }
```

SUPERUSER turns on (and off) CLI super powers — the ability to read, create, modify, or delete a file anywhere on the system. It also allows the CLI to unprotect or protect any file on the system.

Normally, a user has access rights only to files created in and under his or her user directory, and to files in :UTIL, which is a public directory. Superuser bypasses all these controls.

Typing `SUPERUSER ON↵` turns Superuser on; typing `SUPERUSER↵` without an argument describes the Superuser status; and `SUPERUSER OFF↵` turns Superuser off. When Superuser is on, the CLI prompt is *).

Superuser can be turned on only from the master CLI process, PID 2, on the system console.

Why Use It?

Superuser is needed by macros to install products, and back up and restore files. You may need it to examine user directories and delete obsolete files that user OP does not own.

We suggest that you keep Superuser on only as needed, and turn it off afterwards. Then, you should type `LOGON username↵`; for example, `LOGON SAM↵`, on the system console. This turns off Superuser (if on) and executes a standard CLI, which cannot turn on Superuser.

```
) FILES/AS/S :UDD:JOE:+ ↵
WARNING: FILE ACCESS DENIED, FILE :UDD:

) SUPERUSER ON ↵
ERROR: CALLER NOT PRIVILEGED FOR THIS ACTION
SUPERUSER,ON

) BYE ↵
AOS CLI TERMINATING ...

You are now
PID: 2 OP   OP   :CLI.PR
```

```
) SUPERUSER ON ↲
*) FILES/AS/S :UDD:JOE:+ ↲

DIRECTORY :UDD:JOE

BIORHYTHM.CLI   UDF     9-JUN-82      14:57:52 20480
BOOMER.PR       PRG     9-JUN-82      14:57:52  65536
.

*) LOGON JACK ↲
.
)
```

Here, a CLI without Superuser on tries to list the files in Joe's directory (:UDD:JOE). The system refuses and displays *FILE ACCESS DENIED.* The user (Jack) tries to turn on Superuser, but — since this CLI is not the master — he gets another error message. Finally, he terminates the CLI with BYE↲, returning to the master CLI, turns Superuser on, and is able to list the files in Joe's directory. For security, he then executes the LOGON macro, bringing up a user CLI on the system console. (All dialog occurs on the system console: PID 2 runs on the system console only.)

TERMINATE

Kills a process

```
TERMINATE pid
```

The TERMINATE command immediately aborts the process specified by the `pid` (process identification) number. If the process is a user process, it logs the user off the system.

TERMINATE works only with a son of your own process. Generally, if you need to use it, use it from the master CLI, PID 2. PID 2 is the father of all processes on the system, thus it can terminate any process on the system.

You can use the WHOS command for a list of processes.

Why Use It?

There may come a time when a process is monopolizing the system by consuming unreasonable amounts of computer time. If the system is sluggish, use the WHOS macro and RUNTIME command to check individual processes. If you determine that one process is probably guilty, and you don't want to shut the system down, you can use the TERMINATE command to kill the process.

CAUTION *Don't use TERMINATE lightly — and don't use it at all for a CEO or INFOS II process (this could harm system databases). If either CEO or INFOS seems to be malfunctioning, shut the multiuser environment down (DOWN macro) and bring it up again.*

Example

```
) RUNTIME  10 ↲
ELAPSED  0:20:02,  CPU  0:08:43.036,  I/O BLOCKS 4556,....
```

This process is using too much CPU time.

```
) SEND  10 Please log off ...↲
```

Wait for PID 10 to log off...

```
) RUN  10 ↲
ELAPSED  0:22:02,  CPU  0:10:03.036,  I/O BLOCKS 5444,....
```

The process is still running — kill it.

```
) WHO ↲
PID: 2  OP  OP  :CLI.PR

) TERMINATE  10 ↲

) RUN  10 ↲
ERROR: ATTEMPT TO ACCESS PROCESS NOT IN HIERARCHY          (PID 10 is gone.)
)
```

TIME

Displays or changes the system time

```
TIME
TIME [hh [mm] [ss]]
```

The TIME command displays the current system date. The TIME command with an argument changes the system date; the time can be set only from the master CLI, PID 2. For hh, use the hour in a 24-hour clock (1 through 24), for minutes *mm* and seconds *ss* , use the correct number (1 through 59); for example, 13 22 40↲ is 1:22:40 p.m.

Why Use It?

TIME is similar to DATE, and it should be as accurate as possible. Use the first form to check the time. Use the second form if the system clock is wrong (the system has been brought up with the wrong time).

Examples

```
) TIME ↲
14:17:46
```

The system time is 2:17:46 p.m.

```
) TIME  13 18 ↲
ERROR: CALLER NOT PRIVILEGED FOR THIS ACTION
```

(At system console, type BYE↲.)

```
You are now
PID: 2  OP  OP  :CLI.PR
```

```
) TIME  13 18 ↲
) LOGON  SARAH ↲            (Run a standard CLI after setting system time.)
```

TYPE

Types one or more files on the terminal screen

TYPE pathname *[pathname] [...]*

The TYPE command copies the file named in pathname to your terminal screen.

TYPE works properly only for text (ASCII) files; for example, those whose names end in .CLI. If you try to TYPE a file whose name ends in .PR or .OL, you'll get strange results.

You can type only those files that you own, unless Superuser is on (master CLI, PID 2 only). To enable other users to type one of your files, use the UNPROTECT macro.

When there is too much text to fit on the screen, use CTRL-S to suspend display and CTRL-Q to resume it. To interrupt a command (as always), type CTRL-C CTRL-A.

Why Use It?

TYPE is the fastest and easiest way to see what's in a file. You can use it on CLI macros and other text files.

Examples

```
) TYPE  FSTAT.CLI ↵
.
. (displays text of FSTAT.CLI)
.

) TY  MYDIR:MYFILE ↵
.
. (displays text of MYDIR:MYFILE)
.
```

These commands type the contents of two files on the terminal screen.

UNPROTECT.CLI

Allow any user access to a file that you own

```
UNPROTECT  pathname
```

Normally, all files in and below each user directory are protected, which means that only the user owns them (can read, modify, or delete them). For system files, the owning user is OP.

After a user unprotects a file, everyone on the system can read, modify, execute, or delete it. Only a user who owns a file can unprotect it.

You can use template characters (for example, +) in the UNPROTECT macro.

Directory :UTIL and all its files are shipped with a limited universal access list, which allows all users to execute and read (but not change or delete) its files.

File access restrictions do not apply to the master CLI, PID 2, when this CLI has Superuser on.

Why Use It?

Sometimes, you may want another user to be able to read or execute a file that you own. Or, you may want to be able to read or execute a file that someone else owns. The UNPROTECT macro is a good way to open access to such a file.

Examples

```
) UNPROTECT  MYMACRO.CLI ↵
MYMACRO.CLI'

) UNPROTECT  +.REPORT ↵
JUNE.REPORT
APR.REPORT
MAY.REPORT
)
```

The first UNPROTECT gives all users owner rights to MYMACRO.CLI; the system verifies. The second UNPROTECT gives all users owner access to all files whose names end in .REPORT in the working directory.

UP.CLI

Starts up the AOS multiuser environment and other DG products

UP

The UP.CLI macro is provided to ease the process of bringing up system software after starting AOS. It starts the EXEC program, XODIAC, INFOS II, and CEO (if your system has them).

UP also executes a file that allows you to use friendly names for diskette units and printers. After you run UP, you can address your devices by name as follows:

Device	**Names**
Diskette unit (primary, rightmost)	@DPM0 (formal name), @DISKETTE, @DISKETTE1, or @RIGHT_DISKETTE
Diskette unit (secondary, leftmost)	@DPM1 (formal name), @DISKETTE2 or @LEFT_DISKETTE
Plotter	@PLOTTER
Printer (Model 10/SP)	@LPT (queue name), @PRINTER, and @CON1 (formal device name).
Printer (Model 20 and 30)	@LPT (queue name); @PRINTER and @CON2 or @CON3 (formal device name.
Tape unit	@MTC0 (formal name) and @TAPE

Only the master CLI, PID 2, can run the UP macro.

Later, to shut down all the software started by UP, use macro DOWN.CLI. Then you can shut down AOS.

The AOS UP macro executes a XODIAC, INFOS, and CEO UP macro, in sequence, if your system has these products. If your system has a communications line that relies on a modem, the XODIAC macro prompts you to dial up the host. Do it. If you don't, products that need the communications line won't work properly.

UP.CLI (continued)

Why Use It?

There may be many software components to a desktop system — and all must be brought up a certain way, in a certain order. The UP macro makes this easy.

Example

```
AOS CLI  rev n  date  time
) UP ↓

Creating the EXEC process...

PID: 3
FROM PID 3 :(EXEC) READY
FROM PID 3 :(EXEC) 8:52

Creating the XODIAC processes...

... (XODIAC messages)...

Creating the INFOS process...

... (INFOS II messages)...

Creating the CEO processes...

... (CEO processes)...
.
↓
)
```

This example shows all processes in the multiuser environment coming up.

WHO

Describes a process.

```
WHO [pid]
```

WHOS.CLI

Describes all processes on the system

```
WHOS
```

The WHO command describes the username of a process ID. If you omit the number *pid* , it displays your own process ID and username.

WHOS.CLI is a macro, supplied to help you keep track of all processes running on your system.

Why Use It?

WHO is useful when, at the system console, you don't know what CLI is running and you want to get to PID 2 to do some privileged operation.

WHOS is useful to see what kind of processes are running. Also, before you shut down a system that has multiple terminals, you should make sure that no users will lose work. The easiest way to do this is with the WHOS and BROADCAST macros. Any user running :UTIL:CEO_DIR:CEO_CP.PR, :UTIL:MBASIC:MBASIC.PR, or :UTIL:SED.PR should be warned until he or she leaves this program. Then, his or her process is no longer displayed by WHOS.

WHO and WHOS.CLI (continued)

Examples

```
) WHO ↲

PID: 12  OP  012  :CLI.PR
```

The PID of this CLI — owned by user OP — is 12.

```
) BROADCAST  System coming down. Please log off. ↲

FROM PID 12 (OP): System coming down. Please log off.
```

(Wait a minute or so for users to log off.)

```
) WHOS ↲

PID: 1  PMGR  PMGR  :PMGR.PR
PID: 2  OP  OP  :CLI.PR
.
.
PID: 14  ALICE  014  :UTIL:CEO_DIR:CEO_CP

) BROADCAST  Alice, please log off -- system coming down. ↲
.
```

You might repeat the WHOS and BROADCAST sequence until PID 14 (Alice) no longer appeared — indicating that she had logged off CEO.

WRITE

Displays arguments on the terminal or writes them to a file

`WRITE` *[argument] [...]*

If you omit switches, the WRITE command displays each ***argument*** on the screen. For an argument, you can use either a text string and/or CLI pseudo-macros (see example).

If you omit arguments, WRITE displays a blank line. Generally, avoid using CLI punctuation — like parentheses or semicolons — *within* WRITE arguments because the CLI will interpret them (described in the CLI manual, under coding aids).

Why Use It?

WRITE can help you document macros, and write text to disk files. Unlike CREATETEXT, WRITE will *append* text to a disk file if the file already exists.

Switches

/L=pathname Writes ***argument*** to the file named in **pathname** instead of the terminal screen.

Examples

```
) WRITE  Hello ↵
Hello
```

This example shows WRITE displaying text.

```
) WRITE  Call me [!USERNAME] on [!PID] at [!TIME] on [!DATE] ↵
Call me CHRIS on 008 at  10:45:34  on  14-DEC-84
```

This example shows WRITE displaying some text with some pseudo-macros.

WRITE (continued)

```
) CREATETEXT  TODAY.CLI ↵

)) WRITE Today's files are ↵
)) FILES/AS/S/AFTER/TLM=[!DATE]  :UDD:[!USERNAME]:# ↵
)) ) ↵
)
```

This example shows WRITE in a TODAY macro. (This macro was shown earlier under the CREATETEXT macro; typing TODAY↵ executes it).

```
) WRITE/L=TODAY.CLI Write The time is [!Time]. ↵
```

This example shows you how to append text to an existing file with the /L switch. The macro, TODAY.CLI, now provides two kinds of information: a user's most recent files and the current time of day.

XEQ

Executes a program

XEQ pathname *[argument]*

The XEQ command executes the program file named in **pathname**. It works exactly the same way as the EXECUTE command, but it's slightly easier to use since it can be abbreviated to one letter. A program is a file that has been built into an executable program; its name ends in .PR. You can omit the .PR from the **pathname**. For many of the programs supplied with AOS, macros do the XEQ command for you, so you need type only the macro name.

The optional ***argument*** typically is the name of a file you want the program to find and process. Generally, you supply an argument when you execute system programs.

```
) XEQ  SED  BOOMER.F77 ↵

Do you want BOOMER.F77 to be created?    Y ↵

.
* . (SED editor commands)

* BYE ↵
)
```

In this example, someone executes a text editor program to write a FORTRAN 77 source program. Next, after compilation and linking, the person can execute the program by typing

```
) X  BOOMER ↵

.
. (program BOOMER.PR executes)
.
```

What Next?

This chapter is really a reference chapter. By section and command, it contains most of the information you need to run and use AOS.

You may want to proceed to CEO (next chapter), a computer language (Chapter 10), or another pertinent chapter.

Starting CEO 6

Read this chapter when

- you want to start the CEO (Comprehensive Electronic Office) on your terminal, and use it.
- you want to start the CEO Word Processor - Independent on your terminal, and use it.

The sections in this chapter are

- What is CEO?
- Using the CEO Electronic Office
- Using the CEO Word Processor - Independent
- What Next?

What is CEO?

Data General's CEO (Comprehensive Electronic Office) system uses your desktop computer to make office work easier.

There are two versions of CEO:

CEO Electronic Office, with Electronic Mail, Word Processor, Electronic Filing, and Electronic Calendar; and

CEO Word Processor - Independent, which includes the Word Processor only.

Using the CEO Electronic Office

The CEO Electronic Office can make your life easier by

- allowing you to create and send mail and memos electronically;
- helping you produce, edit, and print documents;
- filing documents automatically, and finding them by different categories;
- maintaining a calendar of important appointments and events.

The main benefits of the CEO system are improved communication, reduced paperwork and typing, and easier, better filing. CEO saves steps and repetition; and it can streamline your organization by improving access to information. CEO Mail and Calendar may be stored on a large central host system to centralize information. If so, you can exchange mail and documents, and schedule meetings, with CEO users on the remote system — and with users on other desktop systems connected to the central host.

CEO — like other desktop software products — is shipped on diskettes. It must be installed on a hard disk (only once) before people can use it. Chapter 2 explains installing CEO.

Starting and Stopping CEO (full product)

Before you can start CEO, its servers must be running. These are run automatically when the multiuser environment is brought up (via the UP macro, described in Chapter 3).

CEO can run on the system console or a user terminal. To start it on the system console, type

```
) CEO ↵
```

With a user terminal, log on as usual with your username and password (described in Chapter 4).

The CEO *Main Menu* may now appear on your screen. If so, you're all set — you can begin to use CEO immediately.

If the *AOS CLI* appears on your terminal at logon, start up CEO by typing

```
) CEO ↵
```

and wait for the CEO Main Menu to appear.

With the CEO Main Menu on your terminal screen, you can start using CEO.

(If you get a *FILE DOES NOT EXIST* message, CEO may not be installed. Verify that someone has installed it as described in Chapter 2. If you see the message *You do not have a CEO profile...* , this means there is no CEO profile for your username. Someone with office manager privileges (for example, user OP) must create one. If CEO is running remotely, this means running the CONFIGURE macro, from PID 2 on the system console, described in Chapter 2. If CEO is all local, have someone with Manager privileges start CEO from PID 2 on the system console. In either case, to create the profile, the person must select "Utilities" and "Office Manager" functions, then fill in a profile menu, giving your username as "User ID", and choosing other CEO options as explained in *Managing Your CEO System.* After all this is done, try CEO ↵ again.)

To run CEO effectively, you'll need a CEO function key template. This is a cutout that fits over the function keys at the top of the terminal and labels the keys. These templates are shipped with CEO. Find yours and place it on the keyboard. The template name and number are

CEO System, 093-000346

For a sample session with CEO, you can read

Getting Started with CEO, number 069-000036

supplied with the CEO software.

For details on the CEO Word Processor, read

Using CEO Word Processing, number 093-000285

In the word processing book, skip the logon section, since you have already logged on.

From any CEO menu, you can use CANCEL/EXIT function key (indicated on the template) — perhaps pressing it several times — to return to the Main Menu.

To stop CEO, return to the Main Menu, press CANCEL/EXIT, and confirm with Y↵. CEO will stop running and you will be logged off:

```
PROCESS n TERMINATED
USER username LOGGED OFF AT date time
.
*** AOS REV n / TYPE NEW-LINE TO START LOGGING ON ***
```

or you will see the CLI prompt, which is a right parenthesis ,). If so, type BYE↵ next to the prompt:

```
) BYE ↵

PROCESS n TERMINATED
USER username LOGGED OFF AT date time
.
*** AOS REV n / TYPE NEW-LINE TO START LOGGING ON ***
```

If the terminal will be unused for a while, like over the weekend, you might want to shut it off using the switch on the right back of the terminal. Remember to use the DOWN macro to shut down EXEC, and CEO before turning off the terminal.

Using the CEO Word Processor - Independent

The CEO Word Processor - Independent can make your life easier by

- helping you to write and revise documents; and
- helping you to format and print documents.

The CEO Word Processor — like other desktop software products — is shipped on diskettes. It must be installed on a hard disk (only once) before people can use it. Chapter 2 explains installing it.

Starting and Stopping the CEO Word Processor - Independent

Before you can start CEO, its server must be running. This is run automatically when the multiuser environment is brought up (via the UP macro, described in Chapter 3).

The Word Processor can run on the system console or a user terminal. To start it on the system console, type

```
) CEO.WP ⤶
```

With a user terminal, log on as usual with your username and password (described in Chapter 4).

The Word Processor *Main Menu* may now appear on your screen. If so, you're all set — you can begin to use it immediately.

If the *AOS CLI* appears on your terminal at logon, start up CEO by typing

```
) CEO.WP ⤶
```

and wait for the CEO Main Menu.

With the CEO Main Menu on your screen, you can start using CEO.

(If you get a *FILE DOES NOT EXIST* message, CEO may not be installed. Verify that someone has installed it as described in Chapter 2. If you see the message *You do not have a CEO profile...* , this means there is no CEO profile for your username. Someone with office manager privileges (for example, user OP) must create one. Have someone with Manager privileges start CEO from PID 2 on the system console. In either case, to create the profile, the person must select "Utilities" and "Office Manager" functions, then fill in a profile menu, giving your username as "AOS username", specifying something like UDD:username:CEO_DRAWER, as the "path to personal drawer", and choosing other CEO options as explained in *Managing Your CEO System*. After all this is done, try CEO.WP↲ again.)

To run CEO effectively, you'll need a CEO function key template. This is a cutout that fits over the function keys at the top of the terminal and labels the keys. These templates are shipped with CEO. Find yours and place it on the keyboard. The template name and number are

CEO System, 093-000346

For a sample session with CEO, you can read

Using CEO Word Processing - Independent, number 093-000220

In that book, skip the logon section, since you have already logged on.

From any menu, you can use CANCEL/EXIT function key (shown by the template) — perhaps pressing it several times — to return to the Main Menu.

To stop CEO, return the Main Menu, press CANCEL/EXIT, and confirm with Y↵. The Word Processor will stop running and you will be logged off:

```
PROCESS n TERMINATED
USER username LOGGED OFF AT date time
.
** AOS REV n / TYPE NEW-LINE TO START LOGGING ON ***
```

or you will see the CLI prompt, which is a right parenthesis [)]. If so, type BYE↵ next to the prompt:

```
) BYE ↵

PROCESS n TERMINATED
USER username LOGGED OFF AT date time
.
** AOS REV n / TYPE NEW-LINE TO START LOGGING ON ***
```

If the terminal will be unused for a while, like over the weekend, you might want to shut it off using the switch on the right back of the terminal.

What Next?

This chapter told you how to start — and how to learn more about — the CEO Electronic Office.

The next chapter tells how to back up your hard disk: copy material from it to another medium for safekeeping.

Backing Up and Restoring Files 7

Read this chapter when

- you want to plan backup procedures;
- you want to back up your files for safekeeping;
- you want some pointers on handling diskettes; or
- you want to restore some or all material from the backup diskettes.

This chapter explains the whys and wherefores of disk backup — sometimes called archiving. The major sections are

- Why Do It?
- Scheduling Your Backups
- Handling and Storing Diskettes
- Using the FULL_BACKUP and INC_BACKUP Macros
- Restoring Material from Diskette
- What Next?

Why Do It?

A lot of information — critically important, irreplaceable information — can be stored in a computer system. The magnetic disk that stores it is stable and reliable, but not invulnerable. Accidental (or malicious) deletions, or mechanical failure could destroy information on it.

Backup procedures ensure that files — like documents, reports, letters, business data, and programs — can be restored if lost.

If information is created and stored on your computer system, you need some procedure for file backup.

The procedures in this chapter assume that your system doesn't have a tape unit. If you do have a tape unit, it uses 15-Mbyte cartridges — thus can hold the entire contents of one hard disk. You may prefer to use tape instead of diskettes for backup. If so, don't read this chapter. Instead, use the DUMP command described in the CLI manual. If you need to restore material, use the LOAD command. The device name of the tape unit is @MTC0 or @TAPE.

Scheduling Your Backups

The time your site spends on backups will vary with the rate that new information develops, and how important that new information is.

The system has several macros to help with backups. The first, FULL_BACKUP, copies all files unconditionally. The second, INC_BACKUP, copies only files created or modified since the last backup from that directory. The third, RESTORE, restores files from either kind of backup.

At minimum, we recommend a full backup each month and an incremental backup each week. This way, at worst case, you cannot lose more than a week's work.

You may want to do backups more often, based on the amount of work you can afford to lose — perhaps with a full backup every 2 weeks and an incremental backup on alternate days or even every day. With an incremental backup every day, you can't lose more than a *day's* work.

You *can* base your backups on the development of new information — for example, do full backups often during periods of high activity. But

— like any other relatively boring task — backups are more likely to happen if they are done at regular intervals.

Full Versus Incremental Backups

A full backup copies all files — which, for one fairly full hard 38.6-Mbyte disk, can take hours, and use many diskettes. An incremental backup copies only new and changed files — and may take only one or two diskettes and 10 minutes. Restoring material from both isn't difficult: you restore each incremental set (latest to earliest), then the full backup set.

If you do few full backups and many incremental backups, you'll spend less time doing backups and more time restoring files (for example, restoring 20 incremental sets and 1 full set). Also, you'll need to keep track of more backup sets. Another disadvantage is that *all* files that were ever backed up are restored from the backup diskettes. This means that, if you have to restore a whole disk, someone will need to go through the directories and delete all the old files that were deleted since the last full backup. (The computer can't tell the difference between files that were deleted intentionally and files deleted accidentally.)

On the other hand, if you do nothing but full backups, restoration is easy: you don't need to maintain or restore any incremental backups, and little cleanup is needed if you have to restore a whole disk. But, the amount of diskettes and time involved may be unacceptable. Generally, a good compromise is one full backup followed by from four to seven incremental backups.

Backup Sets of Diskettes

One hard disk can hold either 15 Mbytes (30,000) disk blocks or 38.6 Mbytes (75,000 disk blocks). You need 40 diskettes to back up a 15-Mbyte disk, and more for a 38.6-Mbyte disk. You'll need fewer diskettes when your disk is not full. To estimate how many diskettes you'll need, type `SPACE` ↵ and then look for the number following CUR (current) in the response. Divide that number by 720; the answer gives you the approximate number of diskettes you will need for a full backup.

Assume two or three 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, you must abort it; this means that 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 relatively fragile — thin plastic, protected only by a thin inner envelope and thicker outer envelope. 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. Take care to avoid touching the diskette surface (exposed in oval sections on each side of 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 where the diskette surface is exposed.* 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. When you insert a diskette, hold it by the edge, with label (seamless) side facing right and the write-enable notch up. 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 substituting a new diskette for the one in unit *n* (an *n* of 0 indicates the primary unit; an *n* of 1 means the secondary unit). 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, continue with the next diskette if possible — but realize that some data will be lost. And when the restoration is complete, you should bring down AOS and run FIXUP on your hard disk, since inconsistent information may have been copied to the hard disk.

- 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 C, between 50 and 125 degrees F).

- A magnetic field can erase part or all data on a diskette. Keep them away from magnets, electric motors, 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, done by the Disk Formatter). Diskettes you acquire 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.

- To write-protect a diskette, put tape over the write-protect notch (a quarter-inch slot on the edge of the inner envelope). A write-protected diskette cannot be written to: you will see error messages on attempts to write to such a diskette. You cannot open a write-protected diskette (OPEN macro) as a directory.

Using the FULL_BACKUP and INC_BACKUP Macros

The FULL_BACKUP and INC_BACKUP macros copy files from the working directory and subordinate directory to diskettes. They use the program MMOVE, supplied with AOS, to write to diskettes sequentially. If you need to abort a backup, use the control sequence CTRL-C CTRL-B. Don't do this unless you must, though — if a backup is aborted for any reason, it is incomplete, and should be started all over again.

The macros are easy to use — providing you have enough diskettes — but they will not serve their purpose unless you observe the following guidelines:

- Always back up from the same directory. Preferably, this will be the root directory (:), from the master CLI (PID 2). If you back up from the root, all user files (including CEO files) will be backed up, and can be restored easily and simply. It can be extremely complicated to restore using backups started from different directories.

  Users can back up at will from their own user directories, but this is no substitute for a system-wide backup. (For example, CEO documents are not necessarily stored in the AOS user directory, thus they may not be backed up if the backup starts from the user directory.)

- Do backups when the system is idle. If CEO or INFOS II processes are up, the backup macro will stop with a *...server process...* error message. And you'll need to bring down the multiuser environment (via **DOWN)**) and retry the backup.

- Allow enough time. Each diskette takes roughly 2 minutes to fill, and then needs to be changed.

- Do not abort the backup unless you must. If stopped for any reason, even a power failure or fatal error, the backup is incomplete — and it must be started all over again. Don't let anyone cut computer power (tantamount to a power failure) during the backup. This is *always* a no-no when AOS is running.

- Use write-enabled diskettes. If a diskette is write-protected, it cannot be written to, and the backup will stop until you insert a write-enabled diskette. Diskettes arrive write-enabled; to write-protect them, someone must tape over the write-enable slot on the inner envelope.
- Make sure the sequence of each diskette is maintained. The first time you do a full backup, write "FULL BACKUP" or the equivalent and the sequence number (1, 2, 3, ...) on the diskette label after each diskette is filled. You might also write the date. Use only a felt-tipped pen to write on a diskette label. If a diskette has no paper label, apply one to the seamless side of the inner envelope, then write on it as above.

  For incremental backup sets, write "INC BACKUP" or equivalent, the number, *and the date* on each envelope. The date is especially important if there will be multiple incremental backups between full backups.
- If the surface of a diskette has too many bad blocks (flawed areas that won't hold information), the backup program will display *TOO MANY BAD BLOCKS*. Discard the diskette and use another — and label the new diskette.

Backup Example

For this example, assume that an AOS system is built, with CEO and other support software, in April 1984. Information starts accumulating on it immediately. The person acting as system manager decides on a monthly full backup and weekly incremental backups.

The full backup will occur on the last Friday of each month. Following incremental backups will occur on the first, second, and third Fridays (with provisions for months with five Fridays and Friday holiday). All backups will be done from the root directory (:).

The first full backup, in April, takes only 10 diskettes; and the related incremental backups take only 1 diskette each. The May full backup takes 12 diskettes and the related incrementals take 1 or 2 diskettes each.

Now, we'll show the actions and dialog at the system console for the June full backup (on Friday, June 29), and for the first following incremental backup (on Friday, July 6).

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 (using the WHOS and BROADCAST macros), as shown in Chapter 5.

2. She returns to the master CLI, PID 2, on the system console:

```
) WHO ↲
PID: 13 OP CONO :CLI.PR                 (This CLI is PID 13, not PID 2.)

) BYE ↲                                 (Sign 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 with

```
) DOWN ↲
.
...(termination messages)...
.
```

4. She makes the working directory the root:

```
) DIR : ↲
```

5. She starts the full backup, including the /L switch to specify a listing file of filenames copied. This is your decision, but we recommend it — the listing tells which files were backed up.

```
) FULL_BACKUP/L=FILES_BACKED_UP ↲
.
** Full backup from directory : at 16:45:05 on 29-JUN-84 **
.
Press NEW-LINE when ready to begin backup.
```

6. She inserts the "FULL BACKUP" diskette labeled number 1 in the primary (rightmost) unit.

7. She presses ↵.

```
Beginning file backup -- a list of files dumped to diskette follows.
.
. (it verifies files copied to print queue or terminal...diskette fills
. up.)
Disk n is full. Mount next disk and type NEWLINE to continue.
```

8. She replaces the diskette with the next one and presses ↵.

9. She repeats steps 7 and 8 as long as the program prompts for another diskette. For this example, let's say she's up to diskette number 13. Some files are copied to the diskette. Then:

```
** Full backup of directory : complete at 17:02:04 **
.
)
```

The full backup is done. The listing file (optional, shown in this example as FILES_BACKED_UP) remains in the root directory. It can be printed (when the multiuser environment is up) and deleted as desired. Each backup listing should be stored in the same place as its set of diskettes. If you specify a listing file each time you do a full backup, simply delete the old one before starting the next full backup.

10. She stores all diskettes safely, in order, in their outer covers, 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, she would probably shut down, turn everything off, and go home.

The Incremental Backup

At the sample site, a week passes — and next Friday rolls around, time for the incremental backup.

1. As last week, Joan prepares to bring the multiuser environment down, making sure that no users will lose work, using the WHOS and BROADCAST macros.

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 ↲                            (Sign off this CLI.)
AOS CLI TERMINATING ...

You are now
PID: 2 OP OP :CLI.PR               (PID 2 — the master.)
```

3. She brings it down with

```
) DOWN ↲
.
...(termination messages)...
```

4. She makes the working directory the root:

```
) DIR : ↲
```

5. 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.JUL.6 ↲
.
** Incremental backup from directory : at 16:55:22 on 06-JUL-84 **

This backup will dump all files created or modified since 29-JUN-84....
.
Press NEW-LINE when ready to begin backup.
```

6. She inserts "INC BACKUP" week 1 diskette number 1 in the primary (rightmost) unit.

7. She presses ↲.

```
Beginning file backup -- a list of files dumped to diskette follows.
.
. (system verifies files copied...)
.
Disk n is full. Mount next disk and type NEWLINE to continue.
```

8. She removes the diskette, writes the date on its label (felt-tipped pen) replaces it in the envelope, and inserts the next diskette.

9. She repeats steps 7 and 8 as long as the program prompts for another diskette. Let's assume she's on the second diskette. Some files are copied to the diskette. Then:

   ```
   ** Incremental backup of directory : complete at 16:59:39 **
   .
   )
   ```

   This incremental backup is done. The listing file (shown in this example as FILES_BACKED_UP.JUL.6) 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.

10. She stores all diskettes safely, in order, in their outer covers, away from strong magnetic fields.

Yet another week passes, another Friday afternoon arrives — July 13 — and she does another incremental backup. The steps are the same as the last incremental backup, except that the dates are different and she uses a different listing file name.

As you can see, the procedure is methodical — and repetitive. But it can be extremly important. Restoring material from diskettes, the ones from this example, is shown in the next section.

Chapter 5 gives syntactical and practical details on the FULL_BACKUP and INC_BACKUP macros.

Restoring Material from Diskette

Restoring falls into two categories: restoring one or more files, and restoring an entire hard disk. The first category is the most common, and the easiest and fastest.

Restoring CEO files falls into the category of restoring one or more files, although it may take a long time.

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 diskette.

There are two things to consider when you do this kind of restoration — 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 they cannot be restored. Otherwise, use the backup that occurred soonest after the files were modified (incremental or full).

If you can't determine the date the files were modified, check the backup listing or listing file (if any); 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'll need to 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 a backup listing — especially for incremental backups.

After you decide on the diskette set(s), you need to choose a pathname template (unless you want to restore the entire backup). The RESTORE macro permits one argument, in which you can specify a directory, a specific file, or a directory and pathname template.

You must restore from the same working directory that the backup was done from. Assuming that this is true, you can restore all files in and below a directory (including subordinate directories), with the template

```
directory-pathname:#
```

If the backup/restore directory is the root (:), you can restore all files in and below a user's directory with the template

```
UDD:username:#
```

CEO (full product) Pathname Templates

From the root, you can restore a CEO user's document and mail files with the template

```
CEO_+:username:#
```

Pathname templates that specify a filename (like :UDD:SALLY:MYFILE+) aren't useful with CEO because you can't tell a document's filename from its CEO document name. CEO has its own document naming conventions — the document names have no relation to the filenames.

CEO Word Processor - Independent Pathname Templates

The Word Processor - Independent puts its documents beneath the user's initial working directory, in the directory specified when the CEO profile was created. Normally (by default) this directory is CEO_DRAWER. CEO Word Processor - Independent document names *are* the AOS filenames.

From the root, you can restore CEO Word Processing - Independent files with the following template:

```
UDD:username:CEO_DRAWER:#
```

File Restoration Examples

As an example, assume user Sally accidentally deleted two files, named REPORT.MAY and SUMMARY. She last modified them a week ago (if the lost files were last modified on different dates, you might need to work from the last full backup). You find and get the appropriate diskette set — which, let's say, was backed up from the root.

If Sally is a CLI user, who created the files with the CLI, SED text editor, or BASIC, you could use a filename template character. Assume the files were in Sally's user directory (not a subordinate directory). The two lost files have the letters MA in common, so you can use the template

```
UDD:SALLY:+MA+
```

The full command line, from the master CLI (PID 2) is

```
) RESTORE  UDD:SALLY:+MA+ ↵
```

If Sally had created the files in CEO (the full product, not Word Processor - Independent), and she had accidentally deleted "documents" named REPORT.MAY and SUMMARY, you could suggest that she check the CEO Wastebasket. In the full CEO product, 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 Sally's documents are not in the Wastebasket, you can restore them from the last backup. But you can't be specific, since CEO document names differ from their AOS filenames. The best choice is a template like `CEO_FILES:SALLY:#` — and then ask Sally to clean up her cabinet after the restoration, since some documents she doesn't want may have been restored along with the desired ones. The full command line, from the master CLI (PID 2) is

```
) RESTORE CEO_FILES:SALLY:# ↲
```

If Sally had created the files in the CEO Word Processor - Independent, the template would be `UDD:SALLY:CEO_DRAWER:#` — and again, Sally would need to check and perhaps clean up by deleting unwanted documents after the restore. The full command line, from the master CLI, would be

```
) RESTORE UDD:SALLY:CEO_DRAWER:# ↲
```

Shortening a Restoration

When you mention a specific filename (like :UDD:TOM:MYFILE) in a restoration, the restoration will end and the CLI prompt will return *as soon as the file has been copied to the hard disk.*

It will tend to take longer if you use a template like :UDD:TOM:MYF+, because the restoration will continue through the last diskette, even after the desired file has been copied. This happens because AOS can't tell, until it reaches the end of the last diskette, that there is no matching filename in the backup set.

So, if you use a template and see that all the filenames you care about have been copied to the hard disk, you can abort the restoration by typing `CTRL-C CTRL-B`. (If you don't abort it, the system will force you to mount each remaining diskette, and wait while it checks for matching filenames.)

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 just one mistake with the specific filename, the system will take you all the way through the diskette set, and not restore the file you want restored. If you think this may be true, you can abort the restoration with CTRL-C CTRL-B and start again.

Restoring an Entire Hard Disk – When and How

The time may come when you need to restore all backed-up material to a hard disk. This can happen when the hard disk wears or fails in such a way that the system can no longer read it — then the disk must be replaced. This kind of failure may have occurred when you type **26H** and nothing happens; or when the Disk Formatter (DFMTR) aborts with the message ***TOO MANY BAD DISK BLOCKS***, or ***INCONSISTENT DIB INFO.*** Or, it may be the case when FIXUP reports ***INCORRECT DISK FORMAT***, or ***CAN'T DELETE ROOT DIRECTORY.***

Or, if you upgrade your system from one hard disk to two, you must back up all material, reformat the disks, and restore the material.

If you have checked with Data General, and either acquired a new disk or decided to rebuild the old one, follow these steps. The steps apply even if you have two hard disk units, and are rebuilding only one unit.

1. Make sure the disk(s) is connected properly — described in the hardware manual shipped with this one.
2. Get the most recent *AOS system* diskettes you received from DG. There should be four or five of these.
3. Return to Chapter 2 and execute all the numbered steps there; then install other software products individually as shown in Chapter 2. Then, if you have done backups, return here to restore user files (you need not configure your system again).

   If you haven't done any backups, user files cannot be restored. And, you must configure your system again as described in Chapter 2. There's no point in reading further in this section if you have no backups.

4. Get all your incremental backup sets and your last full backup set of diskettes. If no incremental backups were done since the last full backup, use only the full backup.

5. Insert the first diskette of the last incremental backup (or full backup, if no incremental backups were done since) in your primary unit, and type RESTORE⤶ without an argument:

   ```
   ) RESTORE ⤶
   .
   ```

6. As the program finishes each diskette and prompts for the next, replace the diskette in the primary unit with the next one.

7. Repeat step 6 until you see the message

   ```
   Restoration of directory : complete at time on date.
   .
   )
   ```

8. Next, restore the remaining incremental backup sets (if any), latest to earliest. With each set, proceed as above: insert the first diskette in the primary unit, type

   ```
   ) RESTORE ⤶
   ```

   and proceed until you see the *Restoration* message.

   Continue with the incremental backups until you have restored them all.

9. Now restore the most recent full backup (unless you did it above, in which case you're done). Insert the first diskette in the primary unit and type RESTORE⤶ as above.

   As the program finishes each diskette and prompts for the next, replace the diskette in the primary unit with the next one.

10. If you see a *CONTROL POINT DIRECTORY MAX SIZE EXCEEDED* message, abort the restoration with CTRL-C CTRL-B. The message means that the hard disk 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 restore this incremental backup.

    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.

If users can't or won't free some disk space, you must do it. Use the master CLI, PID 2, on the system console: turn Superuser on (SUPERUSER ON↵) and delete some old user files. You can delete user directories as shown in Chapter 5, DELETE command.

When you have some disk space free, say 500 to 1000 blocks or more (use SPACE :↵), retry the restoration that produced the error.

11. Make sure you return all diskettes to their outer envelopes and store them safely.
12. You're done! You've recreated the entire hard disk(s) — hopefully having lost only a little work (the files created or changes made since the last backup occurred).

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_BACKUP 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 (neither will the FSTAT macro). Chapter 5 gives syntactical and practical details on the RESTORE macro.

What Next?

This chapter gave details on backup procedures: handling diskettes; planning; and using FULL_BACKUP and INC_BACKUP to back up and RESTORE to restore.

You might want to learn about hardware upgrades and software updates in the next chapter or computer languages in Chapter 10.

Adding New Software, Software Revisions, and New Hardware to Your System

8

Read this chapter when

- you have just acquired a new device, like a USAM communications multiplexor, and installed it in your desktop computer;
- you have acquired a new software product, like XODIAC networking software, CEO Software, or a computer language;
- you receive an update of a software product you already have on your system.

This chapter tells you how to handle new hardware and software acquired *after* your first system has been running for a while. The major sections are

- After Adding New Hardware
- Adding New DG Software
- New Revisions of DG Software
- Getting Help from DG
- What Next?

After Adding New Hardware

Hardware upgrades are available for many desktop configurations. For example, you can add a communications multiplexor, terminals, or other hardware to many systems. Your DG sales representative can tell you more about this.

After you install your new hardware, and test it as described in the hardware documentation, you should install any new software received with the hardware as described in the next section.

Note that if you add a new hard disk, you will need to format it together with your old disk before you can use it. This means doing a full backup (Chapter 7), then running the Disk Formatter (ERASE all files), and specifying 2 hard disks. Follow the steps given in Chapter 7, under "Restoring an Entire Hard Disk – When and How".

Adding New DG Software

When you upgrade your hardware, you may receive a new version of AOS software designed to run it. Or, you may acquire new software like XODIAC or CEO.

If you receive a new revision of AOS, install it first, as described in this chapter.

If — with or without a new AOS — you receive XODIAC software, and never ran it before, install XODIAC after installing AOS, as described in Chapter 2.

If you receive new CEO software, you must install the software before you reconfigure AOS (if you need to run CONFIGURE again). CEO requires the following steps before you run CONFIGURE:

For CEO Electronic Office

1. Install XODIAC networking — if not installed.
2. Install INFOS II — if not installed.
3. Install Sort/Merge — if not installed.
4. Install CEO.
5. Run the CONFIGURE macro; and for a remote CEO, create CEO profiles.
6. Run the UP macro.
7. Create CEO profiles — if needed for local CEO.

For CEO Word Processing - Independent

1. Install Sort/Merge — if not installed.
2. Install CEO.
3. Create CEO profiles — if needed.

For other products, like a language, configuration doesn't interrelate with any other product.

Installing all software products, and running all macros, are described in Chapter 2.

New *revisions* of existing software are described next.

New Revisions of DG Software

DG continually improves its software products and updates its technical manuals. It sends the improved products and manuals to all customers on the optional *Software Revision Service.*

If you subscribe to the Software Revision Service, you will periodically get diskette sets and new manuals. Generally, you should install the new software.

Installing a New AOS Revision

Each AOS revision includes new versions of *all AOS program and support files*, on an AOS diskette set. It also includes all pertinent AOS manuals, including a printed *Release Notice* with changes we have not as yet incorporated into the manual. Each revision of AOS is one integer number greater than the last; for example, 5.00, 6.00, 7.00.

After receiving a new AOS revision, read the Release Notice to check for any special warnings. The Release Notice also names all program files and pertinent manuals in the release. Each file in the release will replace the current version on the hard disk (if there is a current version). This will bring *all* your AOS system files up one revision level.

The files shipped in revisions are those in the root, and directories :UTIL and :HELP. An OP user directory, :UDD:OP, and profile, :UPD:OP, are also shipped — but these will not replace the original OP profile (since permanence is set on for the original).

To install a new revision, proceed as follows:

- Shut down AOS (type `BYE↵` to the master CLI).
- Insert the new AOS diskette 1 of the new revision in the primary (rightmost) diskette unit.
- Bring up and use the Installer program (INSTL) from diskette:

  ```
  20H
  ```

  (Wait 20 seconds or so.)

  ```
  PROGRAM NAME?   INSTL ↵
  ```

  (Wait 20 seconds or so.)

  ```
  AOS INSTALLER (INSTL) REV x.xx
  ```

  ```
  -- INITIALIZING THE DISK
  ```

  (Wait 20-30 seconds.)

  ```
  -- STARTING SYSTEM INSTALLATION
  ```

  (Wait for 1 to 2 minutes.)

```
-- SYSTEM INSTALLED

DONE!

.
!
```

- Bring up AOS and load the new files into the root directory:

```
26H

.
PROGRAM NAME?    AOS ↲

INITIALIZING SYSTEM DATA ON DISK --
```

(Wait a minute or so.)

```
INITIALIZATION COMPLETE

AOS REV x.xx

DATE (MM/DD/YY)?    3 20 85 ↲
```

(Type the current date.)

```
TIME (HH:MM:SS)?    13 20 ↲
```

(Type current time, 24-hour clock.)

```
DO YOU WANT TO INSTALL AOS SYSTEM SOFTWARE [N] ?    Y ↲
```

(Type Y ↲)

```
MOUNT AOS DISKETTE 2 -- TYPE NEW-LINE WHEN READY
```

Change diskettes: remove AOS diskette 1 and insert AOS diskette 2.

Press ↲.

AOS now copies the files from diskette into the root directory. This may take 2 or 3 minutes. When all new files have been copied to the hard disk, the CLI prompt appears on the system console.

```
AOS CLI REV n date time
)
```

- Change diskettes again: remove the AOS diskette from the unit, replace it in its envelope, and insert the next AOS diskette.

```
) INSTALL AOS ↲

Please insert the diskette with desired DG-supplied AOS software
into your primary diskette unit -- @DPM0.
```

```
Press NEW LINE when you are ready to begin installation.
```

Press ↵.

```
filename DELETED
```
(name of file installed)
```
filename DELETED
.
)
```

When all files on this diskette have been installed, the system console will prompt for a new one:

```
This disk n is exhausted. Mount disk  and press NEW-LINE to continue.
```

Remove the diskette from the primary unit, insert the next diskette, and press ↵. Repeat this step, replacing the diskette and pressing ↵, until you see

```
AOS software installation is complete. Please remove diskette from
the primary unit.
```

Remove the diskette and replace it an outer envelope.

- If you received the General Language Development Package (GLDP), be sure to install the software from these diskettes (since many of the GLDP programs are revision-locked to AOS and may not work properly with a different revision of AOS).
- You're done. You've installed the new revision of AOS. You need not reformat disks or diskettes for the new AOS revision (the Disk Formatter produces revision-independent formats).

If, for any reason, you wish to install an *old AOS revision*, get out the old revision's diskettes and follow the procedure above.

As you can see, installing a revision isn't difficult. However, it does assume one thing:

> that you haven't reconfigured the directory structure shipped by DG. For example, if a DG-supplied program has been moved from its original directory, the new revision will not replace it (there will be two revisions of the same program on the hard disk).

Thus, if you want the update mechanism to work properly (and for several other technical reasons) you should never move a DG-supplied program or file out of the directory in which it was supplied.

Needless to say, keep all the system diskettes you receive from DG, and those you make yourself, in a safe place.

New Revisions of Other DG Products

The software products included with AOS — INFOS II, Sort/Merge, and MP/BASIC — have a revision procedure similar to that of AOS. If you receive new revisions of any of these along with AOS, you should consider installing them, according to the installation criteria given in Chapter 2. Don't install a product if you don't need it — but if your system has the old version, be sure to install the new one.

Other software products, like CEO and XODIAC, also have a similar revision procedure.

For every software product, you use the INSTALL macro to install new revisions. INSTALL includes a delete switch (to delete existing matching filenames regardless of their creation dates), so, when you install, you'll see a lot of *DELETED...* messages. Don't be alarmed by these. This also means that you can install an older product over a newer one — but don't do this unless you have been advised by DG to do it or know that you must do it.

For every product, read the product Release Notice before installing. Then, to install, use the INSTALL macro and the product procedure described in Chapter 2.

Getting Help from DG

When a customer buys a desktop system, he or she can also buy the right to call special telephone numbers for help. This is called the *Telephone Subscription Service* and it covers a limited period of time. When the Telephone Subscription Service period expires, a customer can extend it with a service called Application Helpline.

Another optional service is the *Online Information Service*, an electronic newsletter on desktop software, which you can tap if your system has a modem.

Yet another option is the *Software Revision Service*, which provides you with new revisions of software as DG creates them. (Installing new revisions is described above.)

Lastly, the *Full Service* option includes telephone access to DG software specialists, the Software Subscription Service, the Online Information Service. It may also include the right to file *Software Trouble Reports (STRs)*.

The special telephone numbers, for customers in North America, are given in Chapter 14, the error chapter.

What Next?

This chapter explained how to configure your system to hardware upgrades, new software, and new revisions to existing software.

Next, you might want to read about DG computer languages in Chapter 10 graphics, in Chapter 13, or the SED text editor in Chapter 9.

Using the SED Text Editor and Other GLDP Products

9

Read this chapter when

- you want to create or edit text, using the SED text editor;
- you want an introduction to other GLDP (General Language Development Programs) products, like Link or the SWAT debugger.

The SED text editor program allows you to write and edit text. SED and other GLDP products are intended to help you write and build your own programs.

This chapter shows you how to use the SED text editor, in a working session; then it describes the other GLDP products. The major sections are

- A Session with SED
- Other General Language Development Programs
- What Next?

Before you can use SED or any GLDP program, the program must have been installed on the hard disk. Installing GLDP products is described in Chapter 2.

A Session with SED

This section shows you how to use the SED text editor. It assumes you are starting in your user directory or in a subordinate directory. With SED, you can edit only the files that you own. You always own files in your user directory.

The SED prompt is an asterisk (*) at the top of your screen. You type SED commands next to the * prompt. During the session, you'll use the ESC key. ESC tells SED to stop accepting text and wait for another command, but ESC does not echo (appear) as a character on the screen.

You can type SED commands in either upper- or lowercase. This chapter shows uppercase for SED commands and abbreviations, and upper- and lowercase for the text written to the file.

In this session, you'll create and edit a sample file named TEXT_FILE. The dialog includes several deliberate errors that you should type as shown.

Get into your initial user directory by typing

`) DIR :UDD:username ↵` (`username` is your username; for example, SAM)

Start SED by typing

```
) SED  TEXT_FILE ↵
```

SED announces itself, and you tell it to create the file as follows

```
SED  Rev. n Input file -  TEXT_FILE
Do you want TEXT_FILE to be created?   Y ↵
*
```

Typing `Y↵` told SED to create the file. A different character, and SED would have asked *Start over?.* If it asks *Start over?,* press ↵ and begin again from the CLI, where you can check file names with the FILESTATUS command (or FSTAT macro).

The line at the top of the screen separates the "*" prompt and SED command line from the file's text. Here, the file is new and there is no text.

Appending and Listing Text

To append text to the file, type

```
* APPEND )
```

When SED is executing a command, the prompt and command remain at the top of the screen. SED always displays line numbers (unless you tell it not to), so the the text field looks like this:

```
1
```

The cursor is next to the 1. Type some sample lines of text:

```
1   This is source line 1. )
2   This is spurce line 2. )
3   This is source line 3. )

ESC
```

The APPEND command (shortest abbreviation is AP) tells SED to append text to the end of the current page (there is only one page). Here, you're appending text from the terminal. To append text from a disk file, you would type `APPEND FROM pathname`.

The line numbers before the text are for your convenience; they do not become part of the file.

The current line is the last line appended. You can easily tell the current line because it is *bright*, whereas the surrounding lines are dim.

SED stores each line of text in the file as long as you end the line with a). When you press the ESC key, SED returns the * prompt and waits for another command.

```
* LIST ALL )

1   This is source line 1.
2   This is spurce line 2.
3   This is source line 3.
```

The LIST command (shortest abbreviation is L) dislays a range of lines. The argument ALL (shortest abbreviation is A) specifies all lines in the file.

Add some text:

```
* AP ↵

4   This is line 4. ↵
5   This is source line 6. ↵

ESC
```

Modifying (Editing) Text

The MODIFY command allows you to edit existing lines of text. When you modify, SED sets position at the beginning of each line. To edit, you can either type over characters from the beginning of the line or set position with cursor control and type over characters. To fix the error in line 2, type

```
* MOD  2 ↵

2    This is spurce line 2.

CTRL-F  CTRL-F →
```

The two CTRL-Fs move the cursor to spurce and → (right arrow) moves it to the incorrect character, p. Type

```
o ↵

2    This is source line 2.

3    This is source line 3.
```

The o overwrote p and pressing ↵ told SED that you were finished with the line. SED then displayed the next line, placing the cursor on T.

Now to insert source in line 4. Skip line 3 by pressing

```
↵

4    This is line 4.

CTRL-F  CTRL-F  CTRL-E  source□  CTRL-E ↵

4    This is source line 4.

5    This is source line 6.
```

The ↵ skipped the current line (3) and displayed the next line (4) for editing. Two CTRL-Fs moved the cursor to *line*. CTRL-E started a text insert; and you typed source□ and ended the text insert with CTRL-E. Then ↵ told SED you were done with the line. With experience, you'll be able to do this kind of thing in seconds.

MODIFY is *the* line editing command. If you omit an argument, SED gives you the current line; if you type a number or relative number (like +20 or -10), SED gives you the specified line. SED stays in MODIFY mode, displaying lines sequentially, until you press ESC or reach the end of the page. So press

```
ESC
```

As you modify, SED places the cursor at the beginning of each line. You then use cursor control characters to edit the line; and when you are done, you press ↵ to tell SED that you are ready to edit the next line. Press the ESC key when you finish modifying text and SED waits for the next command.

The cursor control/screen-editing characters work the same way for both SED and the CLI. They work in SED's APPEND, MODIFY and INSERT commands, and while you are typing SED commands. They are

→ key Moves the cursor forward one character.

← key Moves the cursor backward one character.

↑ or ↓ keys Moves the cursor up or down one line. In Modify mode, it maintains current column position as it moves up or down a line.

ERASE EOL or CR keys Deletes all characters to the right of the cursor. CR continues to the next line.

CTRL-A Redisplays the last command or the last text appended or inserted. In MODIFY mode, CTRL-A moves the cursor to the end of the line. In APPEND or INSERT mode, it repeats the last line you typed.

CTRL-B Moves the cursor one word backward.

CTRL-E Inserts new text or terminates an insert.

CTRL-F Moves the cursor forward one word — to the beginning of the next word.

CTRL-Q Resumes a display suspended by CTRL-S.

CTRL-S Suspends display; useful during a LIST ALL command for a big file.

DEL key Deletes the preceding character.

HOME key Moves the cursor to beginning of the line.

TAB key Produces a tab. Tabs are mandatory before statements in FORTRAN or COBOL programs; they also help you format conventional text.

Inserting and Deleting Text

At this point, you've appended, listed, and modified text in the file. The text looks like this:

```
1       This is source line 1.
2       This is source line 2.
3       This is source line 3.
4       This is source line 4.
5       This is source line 6.
```

Insert a line between 4 and 5 by typing

```
* IN  5 ↵

5      This is source line 5. ↵

6
```

INSERT inserts one or more lines *before* the specified line. If you omit a line number, it inserts text before the current line. INSERT mode continues, with SED displaying updated line numbers, until you hit ESC. So, press

```
ESC
```

To insert text from a disk file instead of the terminal, you can type `INSERT FROM pathname ↵`.

Delete line 6 by typing

```
* DEL  6 ↵

.    .
5    This is source line 5.
```

DELETE deletes the line (or range of lines, like 4 6) that you specify. If you omit line numbers, DELETE deletes the current line. If you make a mistake with DELETE, SED lets you fix it: type

```
* UNDO ↲

.    .
6    This is source line 6.
```

UNDO is useful when you accidentally delete text. Having undone the the DELETE, delete line 6 again:

```
* D  6 ↲

Command not unique, correct the command:

* DE  6 ↲

.    .
5    This is source line 5.
```

As you can see, abbreviations that are too short do no harm; SED simply displays an error message and beeps. To fix it, type more of the command name. (CTRL-A is handy here: it redisplays the command; and you can use cursor keys, type only one or so additional letters, and type ↲ to execute the command again.)

Finding Text

Finding text strings is an important part of the editing process. To try it, type

```
F  '1.' ↲

String not found, correct the command:

* F  '1.'  ALL ↲

1    This is source line 1.
```

The FIND command searches for a text string. If found, SED positions at the line. If you omit an argument (for example, 20 LAST or ALL), SED searches from the current line position onward.

Your first FIND command failed because the line position was line 5 (after the DELETE), and text string 1. doesn't exist after line 5. The second FIND command specified ALL lines, so SED searched all lines and found 1 in the first line.

You can use either apostrophes ('string') or quotation marks ("string") to specify a text string.

You have now run SED and created a file, and appended, listed, and modified text with cursor control characters. You've inserted and deleted lines and found text. The commands involved were APPEND, LIST, MODIFY, INSERT, DELETE, and FIND. These commands — and BYE — are the most common and will allow you to edit any text file.

The commands that follow simply make editing easier.

Setting POSITION

Unless you change it, the current line position is at the last line appended, inserted, or modified, or just after the end of the line deleted or found with FIND. SED displays a range of lines around the current line, which is bright. The POSITION command changes the current line position.

Type

```
* F  '2.' ↵
```

```
String not found, correct the command:
```

```
* PO  1 ; F  '2.' ↵
```

```
2     This is source line 2.
```

Here, trying to find 2., you hit the same error as before. But, instead of typing `FIND '2.' ALL↵`, you set position on line 1 and typed `F '2.'↵`. The end result is the same; the search started before the string 2. so it found 2. But POSITION searches are a little more limited than ALL searches because they can't scan the first line. (The semicolon simply allows you to stack SED commands.)

POSITION is useful when you want to insert text, and for the commands described next.

Moving and Duplicating Text

The ability to move and duplicate text quickly and easily is a major benefit of a text editor.

The text in TEXT_FILE looks like this:

```
1       This is source line 1.
2       This is source line 2.
3       This is source line 3.
4       This is source line 4.
5       This is source line 5.
```

Type

```
* MOVE  2  BEFORE  1 ↵

1     This is source line 2.
2     This is source line 1.

* MOV  2  BEF  1 ↵

1     This is source line 1.
2     This is source line 2.
.     .

* MOV  1  3  ONTO  TEXT_FILE.FOO ↵

FILE  TEXT_FILE.FOO  CREATED

1     This is source line 4.
2     This is source line 5.

* PO  1 ↵
* IN  FROM  TEXT_FILE.FOO ↵

1     This is source line 1.
2     This is source line 2.
3     This is source line 3.
4     This is source line 4.
5     This is source line 5.
```

As you saw, MOVE can move a range of text elsewhere in the file (the two MOVE...BEFORE commands). And it can move a range of text into another file, creating that other file if need be. If the other file already exists, MOVE *appends* the text to it. Having moved the text out of the file, you then set POSITION to restore it, and inserted it. File TEXT_FILE.FOO remains on disk in the working directory.

Type

```
* DUP  2  5  ONTO  TEXT_FILE.FOO ↵
```

DUPLICATE works the same way as MOVE, but doesn't remove the specified lines. It simply copies them to the destination, within or outside the file. File TEXT_FILE.FOO now has seven lines of text: three from the MOVE command, four from the DUPLICATE command.

The destination argument in MOVE or DUPLICATE commands can be BEFORE line-number, AFTER line-number, or a file pathname.

MOVE and DUPLICATE, in conjunction with APPEND or INSERT, can save time with files that have many occurrences of similar text strings. You can simply copy the text in the file as needed.

Substituting Text

Type

```
* SUBS  'new source'  for  'source'  ALL ↵
```

```
1    This is new source line 1.
2    This is new source line 2.
3    This is new source line 3.
4    This is new source line 4.
5    This is new source line 5.
```

SUBSTITUTE allows you to substitute one text string for another text string within a specified range. SED displays each changed line. If you omit a range argument, SED changes every matching text string from the current line onward.

Displaying Edit Status

Type

```
* DIS ↵

Edit file name -   :UDD:username:TEXT_FILE

Current Page:     1    Last Page: 1
Current Line:     5    Last Line: 5
   .
   .
```

DISPLAY gives quite a lot of information, as you can see: file pathname, current page, current line (on page), last page, last line (on current page), and other items we won't explain here.

(The page will always be 1 unless you split the file with a command described in the *SED User's Manual.* A single page can have up to 1,024 lines, 15 or so printed pages — enough for most programs and documents.

The DISPLAY command is most useful when you forget the file name or the number of lines there are.

SED Function Keys

Across the top of your terminal keyboard is a row of function keys. A cutout called a template fits over the keys, labeling their functions. (This has *no* relation to filename template characters, like +.) If you don't have a SED template, you can order one. The part number is 093-000361. Function keys are also described in the *SED User's Manual,* number 093-000249.

Function keys can make editing a lot easier. To illustrate, we'll need to make the file a little longer. Type

```
* DUP ALL ON FOO; DU ALL ON FOO ↵
.    .
* AP FROM FOO; AP FROM FOO ↵
.    .
.    .(new text)
```

Adding two TEXT_FILEs to FOO, then two FOOs to TEXT_FILE, makes the file long enough.

Now, look at the row of function keys. Press the first (leftmost) key. (The template labels it UP ONE SCREEN.)

SED displays text, and you should note the new current line. You'll see that pressing the first key moved the position backward.

Now press the *second* function key, directly to the right of the first, and note the new current line again. (The template labels this as DOWN ONE SCREEN.) Press the second key again and note the new current line.

You'll see that sequential use of the second function key moves the position forward one entire screen; and sequential use of the first key moves the position *backward* one entire screen.

These function keys are very handy when you want to scan or review text. To scan the whole file without using function key, you must type sequential `POSITION +20↵` commands, or type `LIST ALL↵` and use CTRL-S and CTRL-Q. Using a function key is easier than either of these.

Getting HELP

Like the CLI, SED has a HELP command which you can use to get details on SED commands. As with the CLI, we didn't mention HELP earlier because you need some background to understand its messages.

Type

```
* HELP ↵
.
...(list of commands)...
.

* HEL  APPEND ↵

APPEND
.
.  (description of APPEND command)
.
```

HELP is very handy when you're unsure of a command's syntax or capabilities.

Leaving SED with BYE

Type

```
* BYE ↲

Output file - :UDD:username:TEXT_FILE
)
```

The BYE command ends the editing session and updates your file with all your changes. Then it returns to the CLI.

If you have changed an existing file, SED asks you if you want a backup file. For example, re-edit TEXT_FILE:

```
) SED  TEXT_FILE ↲

SED Rev  ...

* DEL  LAST ↲
.
.

* BYE ↲

Do you want to save the original file as a backup file?   Y↲

Output file -   :UDD:username:TEXT_FILE
Backup file is   :UDD:username:TEXT_FILE.BU

)
```

The backup file is useful if you need to check the original contents of the file.

SED allows only one backup file for any file; so, if a backup filename.BU exists, and you tell SED to save the original as backup, SED will delete the older filename.BU and replace it with the newer version.

If you don't want a backup file, press ↲ in response to the ...*backup file?* question.

Summary

In this SED session, you've created and edited a file, using the commands SED (a macro), APPEND, LIST, MODIFY, INSERT, DELETE, FIND, POSITION, MOVE, DUPLICATE, SUBSTITUTE and DISPLAY. You also used function keys, and the BYE command. These commands, and the function keys, allow you to do nearly all the editing you want.

Three new text files result from the session: TEXT_FILE, FOO, (created by the SED MOVE command), and backup file TEXT_FILE.BU. SED also created a file named TEXT_FILE.ED for its own use, but you can ignore this.

Printing Files

If your system has a printer, you can print any text file with the CLI QPRINT command; for example,

```
) QPRI  TEXT_FILE ↓
QUEUED, SEQ=n, QPRI=127
)
```

Now you can go to the printer and pick up the printed file. Operating the printer is described under "Printing Files" in Chapter 4. You *can* print text from the SED editor, but it's easier from the CLI.

SED Command Summary

Table 9-1 summarizes the SED commands we've used. The *SED User's Manual,* 093-000249, describes all these commands in more detail.

Table 9-1 SED commands used in the session (continues)

Action	Command Form	Example
Edit a file.	SED pathname	) SED MYFILE ↵
Add text to the file.	APPEND *[pathname]*	* APPEND ↵ *1* This is a file. ↵ ESC
Examine what you have written.	LIST *[range]*	* LI 1 20 ↵
Change one or more lines.	MODIFY *[address]*	* MOD 2 ↵
Insert new lines of text.	INSERT	*IN 16 ↵ *16* On the 9th.. ↵ ESC
Delete one or more lines.	DELETE *[range]*	* DEL 18 20 ↵
Find a word or phrase.	FIND 'str' *[range]*	* F '1984' ALL ↵
Change position.	POSITION address	* PO -10 ↵
Move text within the file or onto another file.	MOVE *[range]* BEFORE address MOVE *[range]* AFTER address MOVE *[range]* ONTO pathname	* MOV 2 3 BEF LAST ↵

Table 9-1 SED commands used in the session (concluded)

Action	Command Form	Example

Copy text within the file or onto another file.	`DUPLICATE` *[range]*
	`BEFORE address`
	`DUPLICATE` *[range]*
	`AFTER address`
	`DUPLICATE` *[range]*
	`ONTO pathname`
		`* DUP 1 3 BEF 10 ↵`

Change many occurrences of a word or phrase to another word or phrase.	`SUBstitute 'str' FOR 'str'` *[range]*	`* SUB 'Yes' for 'No' 1 20 ↵`
Display edit status.	`DISPLAY`	`* DIS ↵` *Edit file...*
Advance a screen.	Function key 2.	
Back up a screen.	Function key 1.	
Get help.	`HELP` *[keyword]*	`* H ↵`

Close and update file and return to the CLI.	`BYE`	`* BYE ↵`
		`)`

Other General Language Development Programs

Along with SED, in the General Language Development Package (GLDP), you received a number of programs and files. These programs consume a lot of disk space. You need not keep all of them; depending on the needs of your system, you can delete some of them. You can always reinstall them later, if needed, by installing the GLDP again.

Table 9-2 lists and introduces the GLDP programs (alphabetically). The + template character is explained in Chapter 4.

Table 9-2 General Language Development Programs (continues)

Program Name	Filename(s) and Comments
Debuggers	There are two debuggers. The high-level language debugger is SWAT. SWAT.PR, SWAT.OL, SWATI.OB (and several +.F77 and +.PL1 examples) — are installed in :UTIL:SWAT. SWAT is very useful for debugging in FORTRAN 77 and PL/I; if you have neither of these languages, you should delete this :UTIL:SWAT directory. The assembly language debugger — DEBUG.OL — is installed in the root directory. To use it, start the program in the debugger via `DEB programname↵`.
Disk file editor	DEDIT.PR — is installed in :UTIL. This editor allows you to edit the contents of actual disk file *addresses*.
Display program	DISPLAY.PR and DISPLAY.OL — are installed in :UTIL. This program can display the contents of any file, or even a diskette that hasn't been opened. It shows the numeric and ASCII contents of each file "address" in any of several formats, with all characters, even nonprinting ones, displayed. It can also display EBCDIC files.
Expanded error message file	ERMES — has more messages than the standard ERMES file supplied with AOS. It includes text messages for SED, Link, SWAT, and languages. This ERMES contains everything you'll need — and then some. Do not delete ERMES. ERMES is always installed in the root directory. Along with ERMES are shipped error message .OB files — in :UTIL:ERMES — for every product, so that you can build your own custom error message file. In most cases, you won't need :UTIL:ERMES. The INSTALL macro gives you the option not to retain this directory. If you *do* retain the ERMES directory, you can use the macro BUILD_ERMES.CLI to build a custom ERMES. Macro BUILD_ERMES builds an ERMES using *all* files whose names end in ERMES.OB (in directory :UTIL:ERMES).

Table 9-2 General Language Development Programs (concluded)

Program Name	Filename(s) and Comments
File compare programs	FILCOM.PR (to compare binary files) and SCOM.PR (to compare text — source — files) — are both installed in :UTIL. As you develop a program, it's sometimes useful to compare different revisions of source (text) versions. It's also occasionally useful to compare binary (.OB or .PR) versions. Generally, SCOM is more useful than FILCOM — it works with any text file. If you know you won't be comparing two files, you can delete both programs.
INFOS II development system	Replaces and extends the INFOS II shipped with AOS — installed in : (root), :UTIL, and :UTIL:INFOS_II. This INFOS includes examples, language interface .OB files, and other things you may want for program development. If INFOS is not installed, the INSTALL macro allows you to skip INFOS installation (which involves three or more diskettes).
Library file editor	LFE.PR and LFE.OL — are installed in :UTIL. The library file editor allows you to build libraries of your most commonly used program routines. Generally, unless you plan to do a fair amount of high-level language development, you can delete LFE.
Link	LINK.PR and LINK.OL — are installed in :UTIL. Link is required to build programs in any compiled language (FORTRAN, COBOL), but is not needed for BASIC or Interactive COBOL programs. Generally, you'll want to keep Link.
Macroassembler	MASM.PR, MASMXR.PR, and MASM.PS — are installed in :UTIL. MASM is needed only for assembly language programming or if you want to create your own central error message file (a custom ERMES). In most cases, you can delete the macroassembler (it's big).
Sort/Merge manuals	Explain the Sort/Merge utility supplied with AOS.
SED text editor	SED.PR and SED.OL are installed in :UTIL:SED. SED help files are installed in :HELP. Generally, you won't want to delete SED.

If you decide to delete any of the files in the General Language Development Package, note the names deleted. If you ever install another GLDP revision, or re-install the old one, you'll need to delete these files again. If — when you try to delete, you get an *ACCESS DENIED* error message, find the message in Chapter 14, the error chapter.

What Next?

After the session and summaries, you have a working knowledge of the SED editor and some sense of the content of the General Language Development Package. You can proceed to another chapter of interest: perhaps programming in Chapter 10 or graphics in Chapter 13.

Programming with DG Computer Languages 10

Read this chapter when

- you want to write a computer program using a DG BASIC language — either the supplied MP/BASIC or optional Business BASIC;
- you want to write a program using a *compiled* language: FORTRAN, COBOL (two versions), Pascal, or PL/I.
- you want to know about building programs in FORTRAN, COBOL, Pascal, or PL/I.

This chapter shows you how to build programs with DG computer languages. It assumes that you have some experience with the language you want to use. The major sections are

- Intepreters and Compilers
- Using BASIC
- Using FORTRAN (FORTRAN 77, 5, or IV)
- Using COBOL (AOS COBOL or Interactive COBOL)
- Using SP/Pascal
- Using PL/I
- What Next?

All these languages are available for desktop systems. MP/BASIC is included with AOS software. A language must be installed on the hard disk before you can use it. Installation is explained in Chapter 2.

Interpreters and Compilers

A computer program is a series of instructions that a computer can execute. It originates with *statements,* usually written as familiar words like PRINT or WRITE. After a person writes the statements, they are translated from the familiar words to a binary form that the computer can understand.

The software that does the translation is called an *interpreter* (BASIC), or a *compiler* (FORTRAN, COBOL, PL/1).

With an interpreter-based language, you execute an interpreter to create and run the program. The interpreter checks each line as you type it; then, it helps you run the program. When you are done, you leave the interpreter. To run the program again, you must execute the interpreter again.

In a compiled language, you type the statements using a text editor. All the program statements form a *source program.* After you finish typing the source program, you run the compiler on it. The compiler checks for errors; and, if it finds none, it builds a file called an *object file* with the statements translated into binary. If the compiler does find errors, you fix them by correcting program statements with the text editor; and you recompile. Next — for most languages — you need to run the Link program on the object file. Link sets up the object binary so that it can execute in computer memory.

Finally, you execute the program; and, if it does not run properly, you go back to change the source file, recompile, and relink. There is another option: examining the executing program with a debugger. A debugger can set *breakpoints* wherever you want to stop the program. At each breakpoint, you can check variable values to see what went wrong. In a compiled language, you usually have to repeat the text editing, compiling, linking, and executing steps several times until the program runs properly.

Interpreted languages are easy to use and allow programs to be written quickly — benefits that have made BASIC the most popular language for personal computers.

Compiled languages take more work than interpreted languages, but the programs they produce run much faster. Also, compiled programs can be larger than interpreted programs, since they don't need to share space with an interpreter.

Using BASIC

There are several BASIC languages available for desktop systems. AOS includes MP/BASIC, a familiar BASIC, based on the ANSI standard. Optional are Business BASIC, a business-oriented BASIC that offers screen management and indexed sequential (ISAM) file management, as well as an INFOS II interface; and Extended BASIC, a versatile BASIC that runs on all DG computers.

This section tells you how to start up, write a program, and stop MP/BASIC and Business BASIC (the sample program also works in Extended BASIC). On the simplest level (shown here), the syntax of the BASICs is identical.

Here are the steps you follow to create a BASIC program:

1. Get into BASIC by typing

   ```
   ) BASIC ↵            (for MP/BASIC)
   ```

   or

   ```
   ) BBASIC ↵           (for Business BASIC)
   ```

2. Write or edit a series of BASIC program statements.

3. Run the program with the BASIC command:

   ```
   * RUN ↵
   ```

4. If the program runs as you want it to, go to step 6.

5. Identify the problem using BASIC runtime error messages or dynamic debugging. This means inserting STOP statements at strategic points, which lets you print variable values, then continue the program — until you've found the bugs. Then return to step 2 and fix the offending statement(s).

6. Save the program on disk by typing the BASIC command `LIST "pathname"↵`. Later you can bring it back into memory with the command `ENTER "pathname"↵`.

7. You're done! Type

   ```
   * BYE ↵
   ```

   to sign off BASIC.

About DG BASICs

A BASIC program is a series of BASIC statements. Each statement begins with a number between 1 and 9999 (up to 65534 for MP/BASIC). BASIC checks the syntax of each line as you type it. When you type the command RUN↵, BASIC executes the statements sequentially by number; thus, the program can do useful work.

MP/BASIC variable and array names can include up to 32 or more letters or — optionally — numbers or underscores (for example, INTEREST_1). All names must begin with a letter. String variable names end with $; for example, NAME3$. You can type variable and array names, keywords, and strings in uppercase, lowercase, or any combination.

Business BASIC variable and array names can include up to six letters and numbers, and must begin with a letter; for example, MONTH2. String variable names can be up to five letters and numbers, and $; for example, BLURB$. All Business BASIC names and keywords are uppercase.

In either BASIC, you can declare array names with the DIM statement; for example, DIM BARRAY(100). For comments, you can use the REM statement.

```
10  REM  -- Dimension array ALEN for the needed values.
20  DIM  ALEN(100)
```

While you are typing a program, or at any point, you can examine its statements with the LIST command. To change an existing statement, type its line number, then the new text. When you're satisfied with a program, write it to disk with the command LIST "pathname↵". Later, you can read it back into memory with the command ENTER "pathname"↵. From BASIC, you can print the program on the line printer by typing LIST "@LPT"↵.

To start work on another program, type NEW↵ and proceed. To sign off BASIC, type BYE↵.

You can execute a BASIC program only from BASIC. You can't do it from the CLI.

Before you start programming in BASIC, we suggest that you create a BASIC directory. This will put your BASIC programs in one place and prevent conflicts with other programs that have the same names. For example, type

```
) DIR/I ↵
) CREATEDIR  BASIC ↵
) DIR  BASIC ↵
)
```

To get into BASIC, type

```
) BASIC ↵            (MP/BASIC)
```

or

```
) BBASIC ↵            (Business BASIC)
```

BASIC will announce itself by displaying a program banner; for example

```
MP/BASIC Revision 3.0
*
```

When the asterisk prompt appears, you are in BASIC and can start typing commands and statements. (If you get a *FILE DOES NOT EXIST* message, BASIC may not have been installed; ensure that it has been installed via the procedures given in Chapter 2.)

A Sample BASIC Program

To familiarize yourself with BASIC, try the program shown in Figure 10-1. Press ↲ after typing each line. (In MP/BASIC, you can use lowercase if you want. The interpreter will identify errors with a caret pointer (^). To reduce keystrokes, you can use screen editing characters like CTRL-A, ⟵, ⟶, and CTRL-E, and so on (described in Chapter 4).

```
10 PRINT “THIS PROGRAM COMPUTES FACTORIALS.”
20 PRINT “TYPE AN INTEGER FROM WHICH YOU WANT THE FACTORIAL.”
30 INPUT FACT
40 REM - SAVE ORIGINAL VALUE. THEN LOOP, REDUCING VALUE BY 1
50 REM - AND MULTIPLYING UNTIL THE MULTIPLIER BECOMES 1.
60 LET FTOTAL = FACT
70 FOR I = FACT-1  TO  1  STEP  -1
80 FTOTAL = FTOTAL * I
90 NEXT I
100 PRINT  “THE FACTORIAL OF ” ;FACT; “ IS ”; FTOTAL
110 END
```

Figure 10-1 BASIC program example

After typing the program, list all its statements with

```
 *  LIST ↵

10 PRINT “THIS PROGRAM COMPUTES FACTORIALS.”
20 PRINT “TYPE AN INTEGER FROM WHICH YOU WANT THE FACTORIAL.”
30 INPUT FACT
40 REM - SAVE ORIGINAL VALUE. THEN LOOP, REDUCING VALUE BY 1
50 REM - AND MULTIPLYING UNTIL THE MULTIPLIER BECOMES 1.
60 LET FTOTAL=FACT
70 FOR I=FACT-1 TO 1 STEP -1
80   LET FTOTAL=FTOTAL*I
90 NEXT I
100 PRINT “THE FACTORIAL OF ”;FACT;“ IS ”;FTOTAL
110 END
```

Note how BASIC compressed your spacing and indented the statement within the FOR/NEXT loop. Now, run the program with * RUN ↵

```
THIS PROGRAM COMPUTES FACTORIALS.
TYPE A NUMBER WHOSE FACTORIAL YOU WANT.
?   7 ↵
THE FACTORIAL OF  7  IS  5040

END AT 0110
*
```

Next, list the program to disk, leave BASIC, re-enter BASIC, ENTER the program, and leave BASIC again:

`* LIST "FACTORIAL" ↵`	Used this way, the LIST command writes the current program to disk under the specified file (path) name. (If you get an *ACCESS DENIED* error message, try a pathname of :UDD:username:BASIC:FACTORIAL to force the program into your BASIC directory.)
`* BYE ↵`	Leave BASIC.
`) BASIC ↵` or `BBASIC ↵`	Start MP/BASIC or Business BASIC.
`...` *`BASIC`* `...`	BASIC displays its program banner.
`* ENTER "FACTORIAL" ↵`	The ENTER command brings a program that was LISTed to disk back into memory.
`* BYE ↵`	Leave BASIC.

Unless you save a program on disk (for example, with the LIST command), all work done on it during this BASIC session vanishes when you leave BASIC. You can use any valid AOS pathname for the program, if you own the directory(ies) involved. To keep a program in the directory from which you executed BASIC, use just the filename.

If a file with the name you specify already exists, BASIC will display a *File already exists* error message. You must delete the old file by typing `DELETE "pathname"↵` before you can list the program under this name.

Now, change the program so that it will rerun until you type CTRL-C CTRL-A to interrupt. First, get back into BASIC, enter the program, and list it. Then add the needed statements:

```
* 106 PRINT "TO EXIT FROM THIS PROGRAM, TYPE CTRL-C CTRL-A." ↵
* 108 GOTO 20 ↵
```

And run it again:

```
* RUN ↵

THIS PROGRAM COMPUTES FACTORIALS.
TYPE A NUMBER WHOSE FACTORIAL YOU WANT.
?   8 ↵
THE FACTORIAL OF  8  IS 40320
TO EXIT FROM THIS PROGRAM, TYPE CTRL-C CTRL-A.

TYPE A NUMBER WHOSE FACTORIAL YOU WANT.
?   9 ↵
THE FACTORIAL OF  9  IS 362880

TO EXIT FROM THIS PROGRAM, TYPE  CTRL-C CTRL-A.

TYPE A NUMBER WHOSE FACTORIAL YOU WANT.   CTRL-C CTRL-A    (Type CTRL-C
                                                            CTRL-A)

...STOP AT 0020          (IKEY AT 0020 in Business BASIC)
*
```

Now list the changed program to disk using its original name:

```
* LIST "FACTORIAL" ↵
... File already exists          (It displays an error message.)

* DELETE "FACTORIAL" ↵
* LIST "FACTORIAL" ↵
*
```

You're done. You've written, run, and modified a BASIC program.

To create and run more complex programs, follow the steps shown above, using the BASIC or BBASIC command to start BASIC; the ENTER command to read a saved program into memory; and the LIST command to see the statements. If a program is too long to fit on the screen, use the form `LIST s TO e↵` where `s` is the starting statement number and `e` is the ending statement number you want to see. Or, you can type `LIST↵` and use CTRL-S and CTRL-Q to suspend and continue display. Change and run the program as desired. When you have finished, save it on disk by typing `LIST "pathname"↵`.

If you want to have the program write to a disk file or printer, it will need to open the file or printer and write to it, using file I/O statements explained in the BASIC books described next.

MP/BASIC Documentation

For all MP/BASIC commands, statements, and functions, see the *MP/Basic Reference*, number 093-400005.

Business BASIC Documentation

For more of an introduction, try *A Guide to Using Business BASIC (AOS/VS, AOS, RDOS, DOS)*, 069-000028.

The heart of Business BASIC is covered in *Business BASIC Commands, Statements, and Functions*, 093-705005; *Business BASIC Subroutines, Utilities, and BASIC CLI*, 093-705006; *Business BASIC Technical Concepts*, 093-705004; and the *BusiGEN*^{TM} *User Guide*, 069-705011, about the BusiGEN program generation utility.

System management issues are explained in *Business BASIC System Management (AOS/VS, AOS, RDOS, DOS)*, 093-705007.

FORTRAN Programming

FORTRAN is one of the oldest and most popular programming languages. This section covers three different DG FORTRAN compilers: FORTRAN 77 (the most modern FORTRAN, nicknamed F77), FORTRAN 5, and FORTRAN IV.

These are the steps you follow to create a program in FORTRAN:

1. Create or edit a source file using the SED text editor, adding FORTRAN statements and comments. Running SED is described in Chapter 9.
2. Compile the source file by typing

   ```
   ) F77  pathname ↲          (For F77)
   ```

   or

   ```
   ) F5  pathname ↲           (For FORTRAN 5)
   ```

   or

   ```
   ) F4  pathname ↲           (For FORTRAN IV)
   ```

   For F77, you include the /DEBUG switch, if you want to use DG's native language SWAT debugger on the program. This compile line (`F77/DEBUG pathname↵`) is needed to include SWAT in the final program.

3. If there are compilation errors, return to step 1 and fix the offending statement(s). If there are no errors, continue.
4. Link the object file to produce an executable program:

   ```
   F77LINK  pathname [subprograms]              (For F77)
   ```

   or

   ```
   F5LD  pathname [subprograms]              (For FORTRAN 5)
   ```

   or

   ```
   XEQ  LINK pathname [subprograms] FORT.LB              (For FORTRAN IV)
   ```

   For F77, include the /DEBUG switch (`F77LINK/DEBUG pathname↵`) if you want to use SWAT to debug the program.
5. Execute the program with the CLI command

   ```
   ) XEQ pathname ↵
   ```

6. If the program runs the way you want it to, go to step 9.
7. Identify logic errors using runtime error messages or incorrect output. For F77, if you included the /DEBUG switch twice, you can debug the program via `SWAT pathname↵`.
8. Go to step 1 and fix the erroneous statement(s).
9. You're done!

Writing a FORTRAN Program

This section shows two example programs: one for F77 and one for FORTRAN 5/FORTRAN IV. The two examples do exactly the same thing; only the syntax differs slightly. Each program asks for a number, computes and displays the factorial of the number, and gives you the option to run it again.

Before you start programming in FORTRAN, we suggest that you create a FORTRAN directory. This will encourage you to put all your FORTRAN programs in one place and prevent conflicts with other programs that have the same names. For example, type

```
) DIR/I ↵
) CREATEDIR  F77 ↵          (for FORTRAN F77)
) DIR  F77 ↵
)
```

Now, use the SED text editor to create the source file. For F77, use the name FACTORIAL.F77. For FORTRAN IV/FORTRAN 5, use the name FACTORIAL.FR (Actually, you can use any valid filename, but the compilers recognize the suffixes .F77 or .FR shown above, and the sample command lines we show will have the name FACTORIAL).

For example,

```
) SED  FACTORIAL.F77 ↵
```

Type one of the following programs: Figure 10-2 for F77, or Figure 10-3 for FORTRAN 5 or FORTRAN IV. Don't forget to insert a tab (use the TAB key) before each FORTRAN statement that doesn't have a C in column 1 or a statement label. The F77 compiler allows upper- and lowercase keywords. The other compilers require all keywords to be in uppercase.

```
C    This F77 program computes factorials. It uses double precision
C    real numbers to allow large values without integer overflow.

        double precision   FACTORIAL, FTOTAL
        character*1        ANSWER

        print *, “This program computes factorials.”
10      print *, “Type a number whose factorial you want.  ”
        read *, FACTORIAL

C    Save original value. Then loop, reducing value by 1, multiplying,
C    and repeating until the multiplier becomes 1.

        FTOTAL = FACTORIAL
        do 100  I  =  FACTORIAL -1,  1, -1
          FTOTAL = FTOTAL * I
100     continue

        print *, “The factorial of  ”, FACTORIAL, “ is  ”, FTOTAL
        print *
        print *, “To run again, press NEW LINE. To stop, type ’S NEW LINE’. ”
        read (*, 110) ANSWER
110     format (A)
C    If the person does not type  S  and does not type  s,  stop.
        if ( (ANSWER(1:1) .ne. “S”) .and. (ANSWER(1:1) .ne. “s”) ) goto 10
        end
```

Figure 10-2 FORTRAN 77 program example

```
C  This FORTRAN 5/IV program computes factorials. It uses double
C  precision real numbers to allow large values without integer overflow.

        DOUBLE PRECISION   FACTORIAL, FTOTAL

        TYPE "This program computes factorials."
10      ACCEPT "Type a number whose factorial you want.  ",FACTORIAL

C    Save original value. Then loop, reducing value by 1, multiplying,
C    and repeating until the multiplier becomes 1.

        FTOTAL = FACTORIAL
        DO 100  I  =  FACTORIAL -1,  1, -1
          FTOTAL = FTOTAL * I
100     CONTINUE

        WRITE (10, 110)  FACTORIAL, FTOTAL
110     FORMAT  (1X, "The factorial of  ", F9.0, " is  ", F20.0, / )
        ACCEPT "To run again, type 1<NEW LINE>. To stop, type 0<NEW LINE>. ", IANSWER
C    If the person does not type 0, repeat.
        IF (IANSWER .NE. 0) GOTO 10
        END
```

Figure 10-3 FORTRAN 5/FORTRAN IV program example

Compiling a FORTRAN Program

Now to compile the source program FACTORIAL. Depending on your FORTRAN, type

```
) F77  FACTORIAL ↵
```
(for F77)

or

```
) F5  FACTORIAL ↵
```
(for FORTRAN 5)

or

```
) F4  FACTORIAL ↵
```
(for FORTRAN IV)

Nearly always, the compiler will find errors the first time you compile a source program. (In this case, the example is error-free, but — for most programs — you can expect compiler error messages.)

F77 and FORTRAN 5 compiler error messages are quite explicit, telling you clearly what lines are wrong, and what's wrong with them. FORTRAN IV compiler messages are numbers, which you must look up in the *FORTRAN IV User's Manual*, Appendix B. (With FORTRAN IV, errors that the compiler reports may have occurred in the *preceding* line.)

If the compiler reports errors with one of these examples, you may have made a typing mistake. Use the SED editor to fix it; and try the compile line again.

Creating the Program File with Link

The Link command line for FACTORIAL is

```
) F77LINK  FACTORIAL ↵           (for F77)
```

or

```
) F5LD  FACTORIAL ↵             (for FORTRAN 5)
```

or

```
) XEQ  LINK  FACTORIAL  FORT.LB ↵           (for FORTRAN IV)
```

While Link builds the program file, it displays

```
LINK REVISION n  ON  date  AT  time
=FACTORIAL.PR  CREATED
```

In FORTRAN, or any other high-level language, Link errors are rare. If you ever *do* receive a Link error message, and the text doesn't tell you how to fix the problem, see the *AOS Link User's Manual.*

Executing the FORTRAN Program

Now, you can execute FACTORIAL.PR. To execute this or any other program, type the XEQ or EXECUTE command (or shortest unique abbreviation), the program name, and ↵. You can omit the .PR suffix. For example,

```
) XEQ  FACTORIAL ↵

This program computes factorials.
Type a number whose factorial you want.   8 ↵          (Try 8)
The factorial of  8.  is  40320.

To run again, .....              (Program types instructions.)
```

To run the F77 program again, press ↲. To run the FORTRAN 5/IV program again, type 1↲.

```
Type a number whose factorial you want.   10 ↲          (Try 10)
The factorial of  10.  is  3628800.
```

```
To run again, .....            (It types instructions.)
```

To stop the F77 program, type S↲ or s↲. To stop the FORTRAN 5/IV program, type 0↲. The program stops and the CLI returns:

```
)
```

FORTRAN (and other compiled languages) have two types of error condition: *compiler errors* , which involve things like syntax; and *runtime errors* , which occur when you run the program. Runtime errors reveal problems that the compiler can't detect, like a filename that the program can't find.

Again, F77 and FORTRAN 5 have good runtime error messages, which are usually specific enough to allow you to fix the problem. FORTRAN IV doesn't identify the error but reports it as *ERROR n*. Code *n* is described in the *FORTRAN IV User's Manual.*

If you can correct a runtime error without changing the program (for example, by moving a needed file into the same directory as the program), fine. If not, you need to repeat the cycle of editing the source to fix the error, compiling, linking, and executing.

If you cannot identify the problem, you can insert numerous PRINT/TYPE statements to monitor the values in variables, and pinpoint the location of the program — or, with F77, you can recompile and relink with the /DEBUG switch and use the SWAT debugger.

Using the SWAT Debugger

With F77, DG's SWAT debugger can really ease the debugging phase of program development. To use SWAT, compile your program with the /DEBUG switch. If you have a printer, it will help to get a listing with the /L switch. For example,

```
) F77/DEBUG/L=@LPT  pathname ↲
```

The /L switch gives you a printed listing with each line numbered; this gives you perspective as you debug the program — but this switch is not required.

Link using the /DEBUG switch:

```
) F77LINK/DEBUG  pathname ↵
```

Start up the program in SWAT:

```
) SWAT  pathname ↵
.
>            (The SWAT prompt is >.)
```

In SWAT, you can set breakpoints by line number with the BREAKPOINT command; list source lines to check logic (LIST); start or run the program with CONTINUE; examine variables at breakpoints with TYPE; get help at any point (HELP); and leave SWAT with BYE.

For example, a SWAT session with the FACTORIAL program might involve the following dialog:

```
) F77/DEBUG/L=@LPT  FACTORIAL ↵
.
) F77LINK/DEBUG  FACTORIAL ↵
.
) SWAT  FACTORIAL ↵
.
> BREAK  15 ↵
Set at .MAIN. 15

> CON ↵

Type a number whose factorial you want.    5 ↵

Breakpoint trap at .MAIN. 15.

> TYPE  FACTORIAL ↵
5.
> BYE ↵
SWAT TERMINATED
)
```

The FACTORIAL program is too small to make good use of SWAT. But in larger programs, when you cannot isolate a problem, SWAT can be invaluable.

FORTRAN Summary and Documentation

You're done. You've written, compiled, linked, and executed a FORTRAN program. To create and run more complex programs, follow the steps shown above, using the text editor to write or fix the source program, then compile it, link it, and execute it. If needed, for F77, you can compile and link it with the SWAT debugger, then debug it. Repeat the development steps until the program runs the way you want it to.

If you want to have the program write to a disk file or printer, it will need to open the file or printer and write to it, using OPEN and WRITE statements defined in the FORTRAN documentation.

F77 Documentation

For more on the F77 language, see the *FORTRAN 77 User's Manual*, 093-000162. For more on the SWAT debugger, see the *SWAT® Debugger User's Manual*, 093-000258.

For information on using FORTRAN 77 and AOS to best advantage, see the *AOS FORTRAN 77 (F77) Environment Manual*, 093-000273.

FORTRAN 5 and IV Documentation

The FORTRAN 5 and FORTRAN IV languages are covered, respectively, in the *FORTRAN 5 Reference Manual*, 093-000085, and *FORTRAN IV User's Manual*, 093-000053.

Using FORTRAN 5 and AOS to best advantage is explained in the *FORTRAN 5 Programmer's Guide (AOS)*, 093-000227,

The *FORTRAN Commercial Subroutine Package*, 093-000107, gives programmers some FORTRAN 5 business routines. And the *FORTRAN QCALLS Reference Manual*, 093-000239, gives the FORTRAN 5/IV to operating system calls.

COBOL Programming

COBOL (Common Business-Oriented Language) is the most popular language for business data management. COBOL programs resemble English — with paragraphs, sentences, clauses, and words. Well-written COBOL programs are — to a large extent — self-documenting.

COBOL also offers ISAM (Indexed Sequential Access Method). With ISAM, a program can find a record by index key (for example, a customer name), or it can find records sequentially (perhaps to list all customers). ISAM works well in any situation where people need to find a record by one of several keys, or to find records alphabetically. It suits both large and small business situations.

There are two COBOLs available with desktop systems: AOS COBOL and Interactive COBOL (ICOBOL). AOS COBOL provides ISAM through the INFOS II file management system; it allows multiple index keys to access one record; and it conforms to ANSI's 74 standard for the COBOL language (number X3.23-1974). ICOBOL has its own, internal ISAM; and it allows primary and alternate keys to access one record. AOS COBOL requires a Model 30 computer to run; ICOBOL can run on any DESKTOP GENERATION computer.

The details on COBOL with ISAM are outside the scope of this book. However, this section *does* show a sample COBOL program and how to develop it — using either AOS COBOL or ICOBOL. The example program works in either COBOL.

These are the steps you follow to create a program in COBOL:

1. Create or edit a COBOL source file. With AOS COBOL, use the SED text editor (Chapter 9). With ICOBOL, you can use either SED or the ICOBOL text editor, IC/EDIT, described in the *IC/EDIT Interactive COBOL Editor* manual.

2. Compile the source file by typing

   ```
   ) COBOL  pathname ↵            (for AOS COBOL)
   ```

   or

   ```
   ) ICOBOL  pathname ↵           (for ICOBOL)
   ```

   If you will want to use the interactive debugger, include the /D switch; for example, type `COBOL/D pathname↵` or `ICOBOL/D pathname↵`.

3. If there are compilation errors, return to step 1 and fix the offending line(s). If there are no errors, continue.
4. For ICOBOL, go to step 5. For AOS COBOL, use the CBIND program to link the object file, producing an executable program:

   ```
   ) CBIND  pathname ↵            (AOS COBOL)
   ```

   If you will want to use the interactive debugger, include the /D switch; for example, `CBIND/D pathname↵`.
5. Execute the program by typing

   ```
   ) XEQ  pathname ↵            (For AOS COBOL)
   ```

   or

   ```
   ) ICX  pathname ↵            (For ICOBOL)
   ```
6. If the program runs the way you want it to, go to step 9.
7. Identify logic errors using runtime error messages or erroneous output. If you included the debugger switches, you can debug the program:

   ```
   ) DEB  pathname ↵  then  USER ↵          (for AOS COBOL)
   ```

   or

   ```
   ) IDEBUG  pathname ↵            (for ICOBOL)
   ```
8. Go to step 1 and fix the erroneous line(s).
9. You're done!

Writing a COBOL Program

The COBOL program in this section asks for a number, computes the factorial of the number, prints the factorial, and gives the person the option to run it again.

Before you start programming in COBOL, we suggest that you create a COBOL directory. This will encourage you to place all your COBOL programs in one place and prevent conflicts with other programs that have the same names. Type

```
) DIR/I ⤶
) CREATEDIR  COBOL ⤶
) DIR  COBOL ⤶
)
```

Now, use a text editor (SED or IC/EDIT) to create the source file. Use the name FACTORIAL.CO (actually, you can use any valid filename, but .CO is the conventional COBOL suffix, and the sample command lines we show will have the name FACTORIAL).

For example,

```
) SED  FACTORIAL.CO ⤶
```

Type the COBOL program shown in Figure 10-4. Don't forget to insert one or more tabs (use the TAB key) as shown. The COBOL compiler requires all keywords (but not text strings in quotes) to be in uppercase.

```
*  This COBOL program computes factorials.

IDENTIFICATION DIVISION.
PROGRAM-ID.  FACTORIAL.
ENVIRONMENT DIVISION.

DATA DIVISION.
WORKING-STORAGE SECTION.
* Declare variables for input, output, and edited output from factorial
* routine; and declare the control variable (I) and the answer variable.

77      FACTORIAL       USAGE IS COMPUTATIONAL PICTURE S99.
77      FTOTAL          PICTURE 9(18).
77      EDITED-FTOTAL   PICTURE Z,ZZZ,ZZZ,ZZZ,ZZZ,ZZ9.
77      I               PICTURE 9(18).
77      ANSWER          PICTURE X(1).

PROCEDURE DIVISION.
        DISPLAY "This program computes factorials.".
CALCULATION-LOOP.
        DISPLAY "Type a number whose factorial you want. "
                WITH NO ADVANCING.
        ACCEPT FACTORIAL.
        MOVE 1 TO FTOTAL.
        MOVE 2 TO I.

        PERFORM FTOTAL-COMPUTATION UNTIL I IS GREATER THAN FACTORIAL.

        IF FTOTAL IS LESS THAN 10000000000000000 THEN MOVE FTOTAL TO EDITED-FTOTA
                DISPLAY "The factorial of ", FACTORIAL, " is ",
                        EDITED-FTOTAL
        ELSE
                DISPLAY "Factorial of ", FACTORIAL, " is too large for format.".

        DISPLAY " ".
        DISPLAY "To run again, press NEW LINE.  To stop, type 'S NEW LINE'. "
                WITH NO ADVANCING.
        ACCEPT ANSWER.
        IF ANSWER IS NOT EQUAL TO "S" AND ANSWER IS NOT EQUAL TO "s"
                THEN GO TO CALCULATION-LOOP.
        STOP RUN.

FTOTAL-COMPUTATION.
        COMPUTE FTOTAL = FTOTAL * I.
        ADD 1 TO I.
```

Figure 10-4 COBOL program example

Compiling the COBOL Program

Now to compile the source program FACTORIAL. Depending on your COBOL, type

```
) COBOL  FACTORIAL.CO ↲          (for AOS COBOL)
```

or

```
) ICOBOL  FACTORIAL.CO ↲         (for ICOBOL)
```

Nearly always, the compiler will find errors the first time you compile a source program. (In this case, the example is error-free, but — for most programs — you can expect compiler error messages.)

COBOL compiler error messages are quite explicit, telling you what lines are wrong, and what's wrong with them. If the compiler reports an error with one of these examples, you may have made a typing error. Use the text editor to fix it; and try the compile line again.

The AOS COBOL compiler produces an object file that must be built (via Link) into an executable program; proceed to the next section. The ICOBOL compiler produces an intermediate file that you can execute via the ICOBOL interpreter; skip to "Executing the ICOBOL Program".

Creating and Executing the Program File — AOS COBOL

The Link command line for FACTORIAL is

```
) CBIND  FACTORIAL ↲
```

When — for more elaborate programs — you want to include a debugger, include the /D switch (CBIND/D instead of CBIND).

While Link builds the program file, it displays

```
LINK REVISION n  ON  date  AT  time
=FACTORIAL.PR  CREATED
```

In COBOL, or any other high-level language, Link errors are rare. If you ever *do* receive a Link error message, and the text doesn't tell you how to fix the problem, see the *AOS Link User's Manual.*

Now, you can execute FACTORIAL.PR. To execute this or any other program, type the XEQ or EXECUTE command (or shortest unique abbreviation), the program name, and ↵. You can omit the .PR suffix. For example,

```
) XEQ  FACTORIAL ↵

This program computes factorials.
Type a number whose factorial you want.    8 ↵            (Try 8 ↵)
The factorial of  +08  is  40,320

To run again, press NEW LINE. To stop, type 'S NEW LINE'.
```

To run the program again, press the NEW LINE key.

```
Type a number whose factorial you want.    10 ↵           (Try 10↵)
The factorial of  +10  is  3,628,800

To run again, press NEW LINE. To stop, type 'S NEW LINE'.
```

To stop the program, type S↵. The program stops and the CLI returns:

```
)
```

AOS COBOL (and other compiled languages) have two types of error condition: *compiler errors*, which involve things like syntax; and *runtime errors*, which occur when you run the program. Runtime errors indicate problems that the compiler can't detect, like a filename that the program can't find.

AOS COBOL has explicit runtime error messages, which are usually specific enough to allow you to find the problem. If you can correct a runtime error without changing the program (for example, by moving a needed file into the same directory as the program), fine. If not, you need to repeat the cycle of editing the source to fix the error, compiling, linking, and executing.

If you cannot identify the problem, you can use the interactive AOS COBOL debugger.

Using the AOS COBOL Debugger

An interactive debugger can really ease the debugging phase of program development. To use it, compile your program with the /D switch. If you have a printer, ask for a listing with the /L switch; for example,

```
) COBOL/D/L=@LPT  pathname ↵
```

The /L= switch gives you a printed listing with each line numbered; this will help give you perspective as you debug. It is not required.

Link using the /D switch:

```
) CBIND/D  pathname ↵
```

Start up the program with the DEB command; then type USER↵:

```
) DEB  pathname ↵

AOS User Debugger
0... 1... 2.... 3....

! USER ↵
PORT NUMBER?
```

The *PORT...* question allows you to run the debugger on a different terminal — for more elaborate programs. Avoid this issue by pressing

```
↵
*
```

In the debugger, you can set breakpoints by line number using the command SET line-number↵; start or run the program with CON↵; examine data items at breakpoints with DISPLAY name↵; and exit with STOP↵. Details on the AOS COBOL debugger appear in the *COBOL Reference Manual,* 093-000223.

The FACTORIAL program is too small to make good use of the debugger. But, for larger programs, when you cannot isolate a problem, it can be invaluable.

You're done with AOS COBOL — skip to "COBOL Summary".

Executing the ICOBOL Program

To execute an ICOBOL program (compiled program), you start up the program in the interpreter. The command to do this with the FACTORIAL program is

```
) ICX  FACTORIAL ↵
```

The ICOBOL interpreter now reads FACTORIAL and starts executing it.

```
This program computes factorials.
Type a number whose factorial you want.   8 ↵          (Try 8)
The factorial of  +08  is  40,320.

To run again, press NEW LINE. To stop, type 'S NEW LINE'.
```

To run the program again, press the NEW LINE key.

```
Type a number whose factorial you want.   10 ↵          (Try 10)
The factorial of  +10  is  3,628,800

To run again, press NEW LINE. To stop, type 'S NEW LINE'.
```

To stop the program, type S↵. The program stops and the CLI returns:

```
)
```

Compiled languages have two types of error condition: *compiler errors*, which involve things like syntax; and *runtime errors*, which occur when you run the program. Runtime errors involve problems that the compiler can't detect, like a filename that the program can't find.

ICOBOL has good runtime error messages, which are usually specific enough to allow you to fix the problem. If you can correct a runtime error without changing the program (for example, by moving a needed file into the same directory as the program), fine. If not, you need to repeat the cycle of editing the source to fix the error, compiling, linking, and executing.

If you cannot identify the problem, you can use the interactive ICOBOL debugger.

Using the ICOBOL Debugger

An interactive debugger can really ease the debugging phase of program development. To use it, compile your program with the /D switch. To execute, start the program in the ICOBOL debugger instead of the interpreter:

```
) IDEBUG pathname ⤶
```

ICOBOL debugger commands are described in the utilities manual (below).

COBOL Summary and Documentation

You're done! You've written, compiled, and executed a COBOL program. For more complex programs, the development steps are similar to those above: write the source file, compile, and fix syntax errors; then link (CBIND) and execute (AOS COBOL) or start in interpreter (ICOBOL). To use a debugger, include the /D switch to compile; and specify the debugger in the link/execute steps.

If you want to have a program write to a disk file (standard or ISAM) or printer, the program will need to declare the file or printer, open it, and write to it, using file I/O keywords defined in the COBOL documentation. For AOS COBOL ISAM, you may also want to read more about the INFOS II system in Chapter 11.

AOS COBOL Documentation For details on the AOS COBOL language, see the *COBOL Reference Manual*, 093-000223. For more on the INFOS II system, see the *INFOS® II System User's Manual*, 093-000152.

ICOBOL Documentation For details on the ICOBOL language, see the *Interactive COBOL Programmer's Reference Manual*, 069-705013. For user information, try the *AOS Interactive COBOL User Guide*, 069-705015. For more on utilities, read *AOS Interactive COBOL Utilities*, To learn about the IC/EDIT editor (if you don't want to use SED), see the *IC/EDIT Interactive COBOL Editor*, 055-004-02.

Pascal Programming

Pascal was created in 1971 by Nicklaus Wirth, as a teaching language that would encourage good programming practice. This section describes DG's SP/Pascal.

These are the steps you follow to create a program in SP/Pascal:

1. Create or edit a source file using the SED text editor, adding Pascal statements and comments. Running SED is described in Chapter 9.
2. Compile the source file by typing

   ```
   ) SPC  pathname ↵
   ```

3. If there are compilation errors, return to step 1 and fix the offending statement(s). If there are no errors, continue.
4. Link the object file to produce an executable program:

   ```
   ) SPCLINK  pathname ↵
   ```

5. Execute the program with the CLI command

   ```
   ) XEQ  pathname ↵
   ```

6. If the program runs the way you want it to, go to step 9.
7. Identify logic errors using runtime error messages or incorrect output.
8. Go to step 1 and fix the erroneous statement(s).
9. You're done!

Writing an SP/Pascal Program

The example SP/Pascal program in this section does the same thing as the other examples: it asks for a number, computes and displays the factorial of the number, and gives you the option to run it again.

Before you start programming in Pascal, we suggest that you create a Pascal directory. This will encourage you to put all your Pascal programs in one place and prevent conflicts with other programs that have the same names. For example, type

```
) DIR/I ↵
) CREATEDIR  PASCAL ↵
```

```
) DIR  PASCAL ↲
)
```

Now, use the SED text editor to create the source file. Use the name FACTORIAL.PAS (actually you can use any valid filename, but the compiler recognizes the suffix .PAS shown above, and the sample command lines we show will have the name FACTORIAL).

For example,

```
) SED  FACTORIAL.PAS ↲
```

Type the following program, shown in Figure 10-5. For readability, you might want to indent lines (use the space bar or TAB key) as shown.

```
program FACTORIAL(output,input);

var
    NUMBER,I:            integer;
    FTOTAL:              double_real; { Avoid integer overflow. }
    ANSWER:              char;

begin
    writeln('This program computes factorials.');
    repeat
        write('Type a number whose factorial you want: ');
        readln(NUMBER);
        FTOTAL := NUMBER;
        for I := NUMBER - 1 downto 1 do FTOTAL := FTOTAL * I;
        writeln('The factorial of ',NUMBER,' is',FTOTAL);
        writeln;
        write('To run again, press NEW LINE.  To stop, type S NEW LINE: ');
        read(ANSWER)
        until (ANSWER='s') or (ANSWER='S')
end.
```

Figure 10-5 SP/Pascal program example

Compiling an SP/Pascal Program

Now to compile the source program FACTORIAL. Type

```
) SPC FACTORIAL ↵
```

Nearly always, the compiler will find errors the first time you compile a source program. (In this case, the example is error-free, but — for most programs — you can expect compiler error messages.)

SP/Pascal error messages are quite explicit, telling you clearly which lines are wrong, and what's wrong with them.

If the compiler reports errors with one of these examples, you may have made a typing mistake. Use the SED editor to fix it; and try the compile line again.

Creating the Program File with Link

The Link command line for FACTORIAL is

```
) SPCLINK FACTORIAL ↵
```

While Link builds the program file, it displays

```
LINK REVISION n  ON  date  AT  time
=FACTORIAL.PR  CREATED
```

In Pascal, or any other high-level language, Link errors are rare. If you ever *do* receive a Link error message, and the text doesn't tell you how to fix the problem, see the *AOS Link User's Manual.*

Executing the Pascal Program

Now, you can execute FACTORIAL.PR. To execute this or any other program, type the XEQ or EXECUTE command (or shortest unique abbreviation), the program name, and ↵. You can omit the .PR suffix. For example,

```
) XEQ FACTORIAL ↵

This program computes factorials.
Type a number whose factorial you want:   8 ↵          (Try 8)

The factorial of  8  is  4.032000000000e+04
```

```
To run again, .....               (Program types instructions.)
```

To run the program again, press the NEW LINE key.

```
Type a number whose factorial you want:  10 ↵          (Try 10)

The factorial of  10  is  3.6288000000000e+10

To run again, .....               (It types instructions.)
```

To stop the program, type S↵ or s↵. The program stops and the CLI returns:

```
)
```

Pascal (and other compiled languages) have two types of error condition: *compiler errors,* which involve things like syntax; and *runtime errors,* which occur when you run the program. Runtime errors reveal problems that the compiler can't detect, like a filename that the program can't find.

SP/Pascal has good runtime error messages, which are usually specific enough to allow you to find the problem.

If you can correct a runtime error without changing the program (for example, by moving a needed file into the same directory as the program), fine. If not, you need to repeat the cycle of editing the source to fix the error, compiling, linking, and executing.

If you cannot identify the problem, you can insert numerous WRITELN statements to monitor the values in variables, and pinpoint the location of the program.

Pascal Summary and Documentation

You're done. You've written, compiled, linked, and executed an SP/Pascal program. To create and run more complex programs, follow the steps shown above, using the text editor to write or fix the source program, then compile it, link it, and execute it. Repeat the development steps until the program runs the way you want it to.

If you want to have the program write to a disk file or printer, it will need to open the file or printer and write to it, using I/O statements defined in the SP/Pascal documentation, the

SP/Pascal Programmer's Reference Manual, number 069-400203

PL/I Programming

PL/I, which stands for Programming Language I, is a very versatile, structured language, useful for almost everything — including number, bit, and string manipulation.

The program development steps with PL/I are identical to those with FORTRAN 77 (F77), described earlier. The conventional source filename suffix is .PL1 for PL/I.

The PL/I compile and Link command lines are as follows.

```
) PL1  pathname ↵
) PL1LINK  pathname ↵
```

In PL/I, you can use the SWAT debugger by including the /DEBUG switch in both compile and link lines, then typing `SWAT pathname↵` — as shown for FORTRAN 77.

Writing a PL/I Program

The following PL/I example does what the other examples do: it asks for a number, computes the factorial of the number, prints the factorial, and gives the person the option to run it again.

Before you start programming in PL/I, we suggest that you create a PL/I directory. This will help you put all your PL/I programs in the same place and prevent conflicts with other programs that have the same name. Create the directory as for FORTRAN, but with the name PL1.

Then, start the SED text editor; for example

```
) SED  FACTORIAL.PL1 ↵
```

and type the program as shown in Figure 10-6. When done, compile and link as shown previously for FORTRAN 77. Then execute it (`XEQ FACTORIAL↵`).

```
/*--------------------------------------------------------------------*/
main:   PROCEDURE;
        DECLARE  fnumber                   FLOAT BINARY(53),
                 infile                    FILE,
                 answer                    CHARACTER(1) VARYING;

        OPEN FILE(infile) STREAM INPUT TITLE(“@INPUT”) ENV(LINEEND);

        /* This procedure finds the factorial, using function 'fact'. */

restart:
        /* Ask person for a number. */
        PUT SKIP LIST( “This program computes factorials” );
        PUT SKIP LIST( “Type a number whose factorial you want .” );

        /* Get the number and print result. */
        GET LIST( fnumber );
        PUT SKIP LIST( “The factorial of ”,fnumber,“ is ”,fact(fnumber) );

        /* Ask for another number. */
        PUT SKIP LIST( “To run again, press 'NEW LINE.'” );
        PUT SKIP LIST(“To stop, press 'S NEW LINE.'” );

        /* Get answer.  If answer is NEW LINE, run again.  */
        READ FILE( infile ) INTO( answer );
        IF answer = “<NL>”C
          THEN GOTO restart;
          ELSE STOP;

        /*------------------------------------------------------------*/
        fact:   PROCEDURE( n ) RETURNS (FLOAT BIN(53)) RECURSIVE;
                DECLARE n FLOAT BIN(53);

                /* This recursive function calculates the factorial.  */
                IF n = 1
                  THEN RETURN(1);
                  ELSE RETURN( n * fact(n-1) );
                END;  /* fact procedure. */
        /*---------------------------------------------------------*/
        END; /* main program. */
/*-----------------------------------------------------------------*/
```

Figure 10-6 PL/I program example

PL/I Documentation

For a primer on PL/I, see *Plain PL/I*, 093-000216. Details on the PL/I language appear in the *PL/I Reference Manual*, 093-000204.

What Next?

If you want to try other desktop software, proceed to the appropriate chapter; or, if you want, review earlier material. Or, you can start writing your own programs, using the pertinent DG language manuals described above.

Managing Records With Sort/Merge and INFOS II

11

Read this chapter when

- you want to learn to use the Sort/Merge utility;
- you want to know more about INFOS II file management.

Sort/Merge and the INFOS II file management system are included with AOS to support CEO software — and you can use one or both of them to catalog, store, and modify records. Sort/Merge is useful whenever you want to reorder records. INFOS II software provides ISAM file management for COBOL or PL/I applications; it also offers sophisticated data management for the experienced applications programmer.

This chapter has the following sections:

- Records and Record I/O
- Sort/Merge
- The INFOS II File Management System
- What Next?

Records and Record I/O

Records pertain to all computer programming — not just to Sort/Merge. Records are simply groups of information that a program reads from and writes to devices like the terminal and disk files. When a program reads a record, this is called input; when it writes a record, this is called output. Both operations are called record input/output, or record I/O.

Records can be read and written by a text editor, in which case they are lines of text like these — in which each line is a record read from the terminal and written to disk. Or, records can be read and written by a computer program that you write; or they can be read and written by Sort/Merge.

Computer systems can use any of several different methods to distinguish one record from the next record. The most common methods are

- using a special character to separate the records. The system treats all characters up to the next special character as a record. Records separated this way are called *data-sensitive* records, and the special character is called the *delimiter.* A common delimiter is ⤸ (the NEW LINE character). Each line of text you type using the CLI or a text editor is treated as a data-sensitive record; the ⤸ you press at the end of the line is the delimiter.

  The system sees three different characters as data-sensitive delimiters: ⤸, form feed (page break), and null. If a file is data-sensitive, and a program reads one of these characters from it, the system treats the characters preceding the delimiter as the record.

  Data-sensitive records are most useful for writing and reading text, because the delimiters ⤸ and form feed (page break) occur as natural line and page terminators in the text.

- using a constant length for the records. The system treats each group of this length as a record. Such records are called *fixed-length* or *fixed* records. With fixed-length records, the system does not look for a delimiter; it just considers each group of characters as a record.

Fixed records are useful for information that adapts well to a constant length, like telephone numbers.

A program specifies the kind of record format (data-sensitive or fixed length) when it *opens* a file. The open sets up a channel between computer and file through which records can flow.

To use Sort/Merge on a file with data-sensitive records, you must tell it to treat records as data-sensitive.

Record Fields

Fields are character positions within records. For example, take the following record (shown in Figure 11-1).

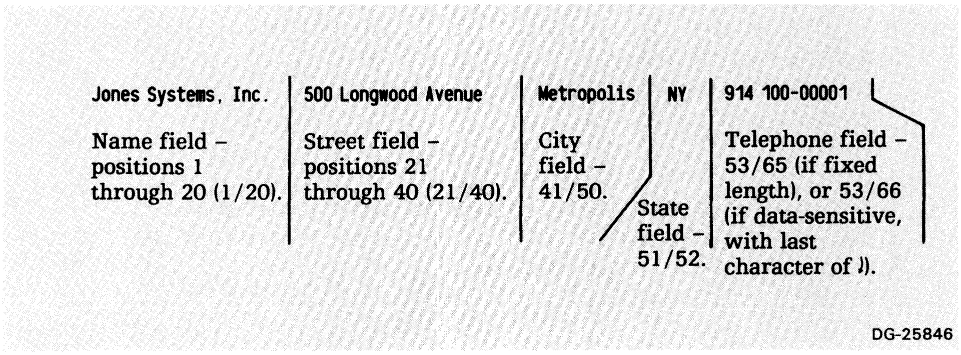

Jones Systems, Inc.
500 Longwood Avenue
Metropolis
NY
914 100-00001
Name field – positions 1 through 20 (1/20).
Street field – positions 21 through 40 (21/40).
City field – 41/50.
State field – 51/52.
Telephone field – 53/65 (if fixed length), or 53/66 (if data-sensitive, with last character of ↓).
DG-25846

Figure 11-1 Anatomy of a record

This record has five fields. Other records in the same file would normally also have five fields, with identical starting/ending character positions. This allows all the records to be read (and the fields changed as needed) consistently. It also allows the records to be sorted by any of the five fields. For example, one could sort by name key (positions 1/20) or state key (51/52), or both. ("Sort" means reordering records by numeric or alphabetical order.)

Sort/Merge

To give Sort/Merge instructions about records, you usually must use a file called a *command file.*

A Sort/Merge command file is a sequence of statements within a text file. You can create command files with the SED text editor. A typical command file (for a data-sensitive "Type" file of records like the one above) looks like this:

```
% This command file does a data-sensitive sort. Records can
% be up to 100 characters long (includes NEW LINE delimiter).
% It sorts on a key field of positions 1 through 20.

 INPUT FILE IS “INFILE”,
 RECORDS ARE DATASENSITIVE UPTO 100 CHARACTERS.
 OUTPUT FILE IS “INFILE.OUT”.
 KEY 1/20.

% The next statement prevents  “Record too short” errors when a
% NEW LINE character is the first char of a record (two NEW LINEs
% occur in a row, as for a paragraph).
 IF “<012>” = 1/1 THEN SKIP.
 SORT.
 END.
```

A typical command file for a fixed-length record sort (again using the record structure shown above) might be

```
% This command file does a fixed-length record sort. Records are
% 63 characters long. There are two keys: 49/50 (as in the sample
% record above, to sort by state) and 1/20 (to sort by name).
% The keys tell Sort to sort by state, and by name within each state.

 INPUT FILE IS “INFILE”,
 RECORDS ARE 63 CHARACTERS.
 OUTPUT FILE IS “SORTED.BY.STATE”.
 KEY 49/50.
 KEY 1/20.
 SORT.
 END.
```

To sort using one of these command files, you'd use the CLI command format

```
SORT/C=command-file-name[/O] [/L=@CONSOLE]
```

For example, if you named the command file CMD.FILE, you'd type

```
) SORT/C=CMD.FILE/O/L=@CONSOLE ↲
```

Sort would read the input file specified by the command file (INFILE), sort the records by the specified characters, and put the sorted records in the output file.

Use the /O switch to have Sort overwrite (delete) the output file if it exists; you need this switch if there is an old file by this name in the working directory.

Use the /L=@CONSOLE switch to have Sort report errors to the terminal. When Sort hits an error, it usually terminates — and you must correct the error (often in the command file) and try the Sort command again. If you don't use the /L= switch, and an error occurs, Sort will give no details. Without details, correcting the error in the command file (or input file) is more difficult.

Details on the Sort/Merge utility appear in the following manuals:

> *Sort/Merge with Report Writer User's Manual*, number 093-000155; or
>
> *Sort/Merge Utility User's Handbook (AOS)*, number 093-000176 (a pocket summary).

Installing Sort/Merge is described in Chapter 2.

The INFOS II File Management System

INFOS II file management system is a database-oriented file management system that lets users create, maintain, and use large databases in multiterminal or batch environments. It is a superset of an ISAM file system.

INFOS II is a hierarchical system that offers multiple levels of index files to access records in a database file. Keys in each level give fast access to the database records.

There can be many different index structures for one database, each providing its own kind of access (for example, customers by name and invoices by number). This variety of possible index structures gives INFOS II its data access versatility. By providing different paths to a single record, multiple indexes eliminate the need for record duplication in the database.

INFOS II databases can be accessed by application programs written in AOS COBOL, FORTRAN 77, FORTRAN 5, PL/I and other languages. INFOS II databases can also be accessed by the PRESENT interactive query program, for graphics and data summaries.

The basic INFOS II system is shipped with AOS because it is needed by the CEO Electronic Office (full product). The entire INFOS II system, with language interface files and examples, is shipped on its own group of diskettes in the General Language Development Package. Installing either the basic or full INFOS II product is covered in Chapter 2.

INFOS II applications are outside the scope of this book. For more on using INFOS II, you can read the following manuals:

> *INFOS® II System User's Manual (AOS)*, 093-000152, for programmers.
>
> *INFOS® II Storybook*, a narrative story of INFOS, number 069-000019.

A program named PRESENT allows terminal users to cull information — for example, for reports — from INFOS II databases. PRESENT can also create charts and graphs. PRESENT documentation is described in the graphics chapter, Chapter 13.

What Next?

In this chapter, you've read about two products for record manipulation: Sort/Merge and INFOS II. Using either product requires *some* computer expertise, but — like any data management software product — each can save enormous amounts of tedious work.

The next chapter describes using communications products that let your desktop system communicate with another computer system. Read it if you want to communicate with a different system.

Communicating with Another System 12

Read this chapter when

- you want to learn how to log on to the remote host system;
- you want to send or receive information directly from another computer system (for example, send or receive a file).

This chapter outlines the DG products that allow desktop systems to communicate with other systems: larger DG systems, other desktop systems, and information banks that you can dial up. The major sections are

- XODIAC Networking System
- DG/XAP™ for File Transfer
- DG/GATE™ for Information Banks
- Installation and Operation
- What Next?

For your desktop system to communicate with another system, it needs *communications hardware*: a device called a USAM (Universal synchronous/asynchronous multiplexor), with either one or four lines. There must be a link between the systems: a direct wire or a phone line (connected to the remote system via a modem).

A phone line has great advantages. The primary one is the huge international telephone network, whose lines reach practically everywhere. To use a phone line, you need a *modem*, also called a *data set*. With a modem, you dial the other system's number on a phone, wait for the appropriate tone, then make the modem connection.

In addition to the hardware described above, your system needs the appropriate software: XODIAC, DG/XAP, or DG/GATE.

XODIAC Networking System

The XODIAC networking system allows a desktop computer to communicate with a large, central DG host system, and with any other desktop system connected to the host. All systems involved must also be running XODIAC software. XODIAC is needed for a centralized CEO Electronic Office (to make Mail and Calendar generally accessible). It runs invisibly with CEO. XODIAC also allows a desktop system user to log on to the central host just as if it were his or her own system. XODIAC for desktop systems works over an asynchronous line.

The XODIAC products are

- The network interface, X.25, connects a DG/IS to any public or private network that follows CCITT Recommendation X.25; it is available by itself or in a XODIAC package with other agents;
- The Resource Management Agent (RMA) and the Virtual Terminal Agent (VTA) manage network resources and allow users to log on to any host;
- The File Transfer Agent (FTA) allows fast file transfer across a network;
- The Remote INFOS II Agent (RIA) and the Remote Database Agent (RDA) provide access to remote INFOS II or DG/DBMS databases.

Installing XODIAC software is explained in Chapter 2. Operating XODIAC requires little effort on the desktop side. The AOS system UP macro executes a XODIAC macro — UP.XODIAC.CLI — that starts the needed XODIAC processes. It also tells you when to dial the remote system, if your communication line relies on a modem. The AOS DOWN macro executes the XODIAC macro, DOWN.XODIAC.CLI, to bring the XODIAC system down.

Using XODIAC to Log On to the Central Host

Part of the XODIAC software known as its Virtual Terminal Agent, VTA, enables you to log on to the central host system, and use it the same way as its local users can. You must have a valid user profile on the remote host system. (A remote *CEO* requires that the usernames and passwords on both systems be the same, but VTA requires only that you *know* a username and password on the remote host system.

To start remote logon, you type X UVTA HOST ↵. XODIAC then makes the connection, via a companion program called UVTA. The central host asks for username and password. After you type a valid username and password, the other system logs you on in the remote directory :UDD:username; and you can do anything allowed there by your access privileges. For example, you can execute programs to get central data, send messages to remote users (using the ?.CLI macro — same as WHOS.CLI — and the CLI command SEND), and so on. To log off and return to your desktop system CLI, type BYE↵ twice.

For example, assume your username and password on the remote system are ADRIAN and ADR — and that you want to greet another user and run a program on the central system. You could type

```
) X UVTA HOST ↵

Virtual Terminal Agent Rev x.xx
Call is Complete

*** AOS/VS Revision n  / Press NEW LINE to begin logging on ***

↵            (Press ↵ to start remote logon.)

Username:   ADRIAN ↵
Password:   ADR ↵            (The password doesn't echo on the terminal.)

Last previous logon ....

AOS/VS CLI  REV n  date  time

) ? ↵            (Check users with ?.CLI macro.)
.
.
PID: 24  CHARLIE  024  :CLI.PR
.

) SEND 24 Charlie: how about discussing the report over lunch? Adrian ↵

From PID 24: Sure. 12:45 at the corner okay?

↵
) SEND 24 Fine -- agreed. ↵

) XEQ WEEK_SUMMARY ↵
.
.
```

```
. (Local desktop terminal displays summary information from
. the remote host.)

) BYE ↲            (Start signoff of remote host.)

AOS/VS CLI TERMINATING  date  time
.
.
Clear - Remote user closed connection.

> BYE ↲            (Type BYE↲ again to the VTA > prompt.)
```

This sample session shows how easy it is to log on to the remote host, send messages (even to CLI users, not in CEO), execute programs, and log off and return to the desktop system CLI.

XODIAC Documentation

The XODIAC manuals are the *XODIAC^{TM} Network Management System User's Manual*, the *XODIAC^{TM} Network Management System — Guide for Managers and Operators*, and the *X.25 Protocol User's Manual (AOS and AOS/VS)*.

DG/BLAST and DG/XAP for File Transfer

DG/BLAST and DG/XAP provide file transfer over an asynchronous line. The remote system must also be running the same program.

DG/XAP has several manuals of its own: *Generating and Running DG/XAP^{TM}*, and *Using DG/XAP^{TM}*. For more information on DG/BLAST, refer to the *How to Use DG/BLAST manual.*

DG/GATE for Information Banks

DG/GATE allows a desktop computer to emulate a terminal produced by any of several major manufacturers. This enables you to dial up and use another computer system (for example, an information pool with electronic bulletin boards or closing market prices). The other system need not be a DG system.

DG/GATE features easy, menu-driven operations and programmable function keys. It can even auto-dial and log on to remote host systems. And, it can record all intersystem dialog in a disk file for later review.

DG/GATE is further described in *Generating, Running and Using DG/GATE^{TM}*.

Other Data General Information System Software

There are a number of other Data General Information System (DG/IS) products that allow DESKTOP GENERATION systems to communicate with other systems: larger DG systems, other DESKTOP GENERATION systems, information banks that you can dial up, and IBM systems.

RJE80, HASP II, RCX70, and DG/SNA allow a DG/IS to talk with an IBM host. While RJE80 and HASP II are batch oriented and send and receive files, RCX70 and Data General System Network Architect (DG/SNA) are interactive; they can receive screen formats and data from an IBM host and send back data to the host. The products are summarized as follows:

- RJE80 provides file transfer between your DESKTOP GENERATION system and any other system that runs RJE80. IBM products that can run RJE80 include the IBM 2780 and 3780 data communications terminals. Virtually all DG systems also run RJE80, which means that this emulator allows you to transfer files between any two DG systems.
- The HASP II workstation allows the DG/IS to emulate an IBM HASP II workstation. It can communicate with any IBM system that supports HASP workstations and runs HASP, ASP, JES2, or JES3. It can also communicate with any DG system that runs HASP II.
- RCX70 allows interactive communications between a DG/IS and any IBM 360/370 host that supports an IBM 3271 terminal cluster controller.
- DG/SNA is a family of products that allows AOS or AOS/VS to communicate with an SNA. Through DG/SNA, a DG/IS can emulate an IBM 3276 controller, with DG terminals and line printers behaving like attached IBM terminal or printers. It also allows applications programs to communicate over an SNA. The DG/SNA products are as follows:

  DG/SNA (the main control program) allows database applications programs to access DG/SDLC and the communications link;

  DG/SDLC (the SDLC interface to the communications hardware) sends and receives information over the communications lines attached to your computer;

SNA/3270 allows DG terminals to emulate IBM 3278/3276 terminals while the host treats DG line printers as if they were IBM 3286/3289 printers;.

DG/SNA/3278/APL allows DG D450 and D460 terminals to emulate IBM SNA/3278 terminals that have the APL character set.

SNA/RJE allows you`to submit batch jobs to a Remote Job Entry (RJE) subsystem running on your network host. Although not identical to any one IBM RJE workstation, SNA/RJE offers most of the capabilities of an IBM 3776 or 3777 Multiple Logical Unit terminal.

There are a number of manuals that describe DG/IS products. RJE80 is described in *RJE80 Control Programmer User's Manual (AOS and AOS/VS)*; *HASP Workstation Emulator User's Manual (AOS and AOS/VS)* tells about HASP II while RCX70 has two associated manuals: *RCX70 Reference Manual (AOS)* and *RCX70 Terminal Operator's Guide.*

The following books discuss DG/SNA: *DG/SNA Programmer's Reference Manual*; *DG/SNA Operator's Guide*; *SNA/APILU2 User's Manual*; *SNA/3270 Operator's Guide (AOS and AOS/VS)*; *DG/SNA DG/3278 User's Terminal Guide*; *SNA/RJE Operator's and User's Guide*; and *SNA/3278/APL Operator's Guide.*

Installation and Operation

Each communications product must be installed on the hard disk before it can be used. XODIAC installation is explained in Chapter 2. Installing another product *usually* involves the general procedure described in Chapter 2, using the product name given in Chapter 5, INSTALL macro.

What Next?

This chapter has given a brief sketch of the communications products available with desktop systems. Next, you might want to check graphics products described in Chapter 13.

Using Graphics 13

Read this chapter when

- you want to learn something about DG's software graphics products.

This chapter sketches the graphics hardware and DG software available with desktop systems. The major sections in this chapter are

- Graphics Hardware from DG
- Graphics Software from DG
- What Next?

Graphics Hardware from DG

Desktop computers — especially the Model 10/SP — include several graphics features. Model 10/SP system console monitor, for example, has a graphics command set and multiple window capability.

As options, DG offers several terminals with additional graphics features. They are

- Model 6262 color monitor CRT, with 16 colors, as system console on Model 10/SP systems only;
- DASHER D410/D460 terminal, as system console or user terminal on Model 20 and Model 30 systems.

Printers available with desktop systems include

- Model 4434 printer, an economical dot-matrix printer with graphics capability that can serve both as a printer and plotter;
- Model 4435 plotter, a color graphics plotter that can plot charts and pictures using felt-tipped pens.

Desktop systems support two input devices

- Model 4436-A mouse, an input device that you move across a flat surface.
- Model 4437 data tablet, an input device that translates graphic information into digital information.

To work properly, the printer, plotter, and/or terminal must have been identified to the CONFIGURE macro. And, the multiuser environment must be running (UP macro).

Model 10/SP System Console Monitor (Standard and Color)

Both the standard and color Model 10/SP system console monitors (screen) have some inherent graphics capability. Each monitor includes a graphics command set for character or bit-mapped graphics. For bit-mapped graphics, main memory is bit-mapped to the system console screen.

The graphics command set includes typed commands like SET FOREGROUND COLOR (color is either green or black on the standard system console, or 1 of 16 choices on the color monitor), START REVERSE VIDEO, or DEFINE CHARACTER. The command set has programmed versions of these commands (these are control sequences that graphics products like TRENDVIEW use).

The graphics commands work on both the standard and color monitors — the only difference is that certain color commands have no effect on the standard monitor. All commands and control sequences are described in the *Model 10/SP Monitor and Keyboard User's Manual*, number 014-000770.

DASHER D410/D460 Terminals

The D460 terminal features reverse video, multiple windows, slow scrolling, compressed type, and user-definable characters. It is supported by TRENDVIEW and GKS software. The D410 terminal, although it is not a graphics terminal, does have reverse video, multiple windows, slow scrolling and compressed type.

Model 4434 Printer

The model 4434 is a versatile, economical dot-matrix printer that — via its graphics capability — can work as a plotter. It accepts 8-bit ASCII codes, which can add 128 characters to the standard set, for special symbols. And it offers a choice of standard type (80 characters per line) or compressed type (160 characters per line).

The Model 4434 printer has a firmware program that allows you to change certain printer characteristics. Operating the model 4434 firmware and hardware is covered in the appropriate hardware *Operating* manual, described in the Preface.

Model 4435 Plotter

This plotter uses two color pens and can plot either on paper or film (for projectable vu-graphs). To add additional colors to a plot, you can change pens.

Operating the model 4435 hardware is covered in the appropriate hardware *Operating* manual, described in the Preface.

Model 4436-A — Mouse

The mouse is a handheld box that connects by cable to a graphics display. The user program, via GKS calls, gets screen coordinates from mouse input and translates them into graphic structures like a straight line or a changed cursor position. GKS is described later in this chapter.

Operating the mouse is described in the *Operating* manual described in the Preface.

Model 4437 — Data Tablet

The data tablet is a large tablet with an electric grid beneath its surface and uses either a cursor puck or a stylus to generate input signals. The user program, via GKS calls, can access coordinates from tablet input and translate them into graphic structures like a straight line or changed cursor position. GKS is described later in this chapter.

Operating the data tablet is further explained in the *Operating* manual described in the Preface.

Graphics Software from DG

The DG graphics software products available with AOS include

TRENDVIEW Charting Package

PRESENT Information Presentation Facility

Graphical Kernel System (GKS)

TRENDVIEW Charting Package

TRENDVIEW is a charting package that can create text files and draw charts from these files, or can draw charts from program-created files. It can also produce color graphics on the Model 10/SP color monitor.

You can have TRENDVIEW draw a chart in one of two ways: from the CLI (after creating a chart file with either TRENDVIEW or SED); or from TRENDVIEW itself. In either case, the multiuser environment must be up (UP macro). TRENDVIEW can draw on different devices; by default, it draws on the screen, but you can also send output to a specified plotter. The command syntax is as follows:

From the CLI,) prompt, type

```
) XEQ TRENDVIEW/DEVICE=@PLOTTER/TYPE=P4435  filename ⤶
```

The filename must contain the information you want plotted, and TRENDVIEW commands that specify the desired chart. You can write these into the filename using a text editor or TRENDVIEW.

From the TRENDVIEW * prompt, specify the device type as P4435, thus:

```
* DEVICE  @PLOTTER TYPE=P4435 ⤶
```

After you execute TRENDVIEW (or, from TRENDVIEW, after you type the CHART command), the plotter will start. The prompt [) or *] won't return to your terminal screen until the plot is done.

To install TRENDVIEW, use the TRENDVIEW diskette and the general installation procedure given in Chapter 5, INSTALL macro.

TRENDVIEW is described in the *TRENDVIEW® Graphics Charting Package User's Manual*, 069-700008.

PRESENT Information Presentation Facility

PRESENT is an information presentation facility that can select stored information to produce reports, and — via TRENDVIEW — create bar, line, and pie charts. PRESENT can be executed directly from the CLI or indirectly through a CEO Electronic Office menu option. PRESENT is described in the *PRESENT Information Presentation Facility User's Manual*, 093-000168.

Graphical Kernel System (GKS)

The Graphical Kernel System (GKS) is a collection of industry-standard graphics-creating routines, which allow FORTRAN 77 or PL/I programs to draw pictures. The programs access GKS routines via runtime calls. Using GKS routines requires some experience with the FORTRAN or PL/I languages.

To install GKS, use the GKS diskette(s) and the general installation procedure given in Chapter 5, INSTALL macro.

GKS is described in the *Graphical Kernel System Reference Manual,* number 093-000355.

What Next?

This chapter has outlined DG's graphics hardware and software.

You've finished the substance of this manual. You might want to examine earlier chapters, or simply use AOS on your desktop system. Enjoy.

Formatting and Copying Diskettes 14

Read this chapter when

- you want to hardware format a diskette;
- you want to software format a diskette, making it into an AOS directory.
- you want to copy a diskette;

This chapter tells you how to hardware format diskettes, and how to software format them, making them into AOS directories. In addition, it tells how to copy one diskette to another when you have either one or two diskette drives. The major sections of this chapter are

- Do You Need to Hardware or Software Format Your Diskette?
- Hardware Formatting a Diskette (Customer Diagnostics)
- Software Formatting a Diskette
- Copying a Diskette
- What Next?

Do You Need to Hardware or Software Format Your Diskette?

Although diskettes look like phonograph records, they are much more delicate. You must handle and store them carefully. (See the section in Chapter 7 called "Handling and Storing Diskettes" for information on how to protect diskettes.) Unlike a phonograph record, you cannot take a diskette out of its protective folder and use it as is. You must prepare a diskette for use. The preparation that the diskette requires is called formatting.

There are two types of formatting: hardware and software formatting. Hardware formatting creates disk blocks so that the system can read and write to the diskettes. All diskettes must be hardware formatted, no matter how you plan to use them. For your convenience, Data General diskettes are hardware formatted in the factory; therefore, you do not need to do it yourself. If you have diskettes from another manufacturer, however, you must hardware format them before using them.

The software formatter (DFMTR) writes tables on diskettes. AOS needs these tables to use the diskette as a directory. In addition, DFMTR creates a bad block table that allows the operating system to cope with bad blocks on a disk or diskette. Only diskettes that you use as AOS directories must be software formatted. If you are using a diskette for backup, or with an operating system other than AOS (such as MS_DOS or CP-M/86 for example), you do not need to software format it.

Table 14-1 describes the formatting operations you need for different uses of the diskette.

Table 14-1 When to Hardware or Software Format a Diskette

Use	Hardware Format	Software Format (Make into an AOS Directory)
For backup (with FULL_BACKUP or INC_BACKUP)	Y	N
For backup (with MOVE command)	Y	Y
For MS_DOS or CP-M/86 storage and access	Y	N
For storing and gaining access to individual files and directories	Y	Y
For copying from one diskette to another	Y	Optional

When you use a diskette, you should label it to tell what is on it and whether or not it is hardware formatted, software formatted, or both. If you have an unlabeled diskette and want to know what it needs, insert it in the primary (right) unit, and try opening it from AOS. Type OPEN ⤶ after you have placed the diskette in the unit. The system displays one of the following messages:

- *ERROR: HARD ERROR, DEVICE 000 020...* means the diskette needs hardware formatting.
- *ERROR: DISK AND FILE SYSTEM REVISION NUMBERS DON'T MATCH* means the diskette is hardware formatted, but not software formatted. It might be a backup diskette, written to with the FULL_BACKUP or INC_BACKUP macros; or it might be blank.
- *Diskette has been OPENed as directory: :path:xxxxx* means that the diskette was software formatted and named xxxxx. It's an AOS directory and is ready for you to use in the same way that you use other AOS directories.

Hardware Formatting a Diskette (Customer Diagnostics)

The Customer Diagnostics diskette can hardware format diskettes and do many other things, like test system hardware and copy diskettes.

You can copy the Customer Diagnostic diskette using its own copy function.

To hardware format a diskette, follow these steps.

1. If AOS is running, shut it down (type `DOWN ↵` and `BYE ↵`). If AOS isn't running and the power is off, turn the power on.
2. Insert the DG Customer Diagnostics diskette in the primary (right) unit.
3. Type `20H` to access the Customer Diagnostic diskette in the unit. After about 20 seconds, the diagnostics program displays

   ```
   AMOUNT OF MEMORY FOUND: n KB

   CURRENT INVENTORY LIST

   .
   . (devices)

   IS THIS ... CORRECT (YES OR NO)?
   ```

4. If the list is not correct, you should change it as described in the Customer Diagnostics chapter of your hardware Testing manual. (For the purpose of formatting diskettes, you can always answer `YES ↵`.) For example,

   ```
   YES ↵
   ```

   Some 20 seconds pass; then the hardware formatter displays another menu:

   ```
   DG SERIES DIAGNOSTIC SYSTEM
   .
   .
   3 - DISPLAY DISKETTE MENU
   .
   ENTER NUMBER OF ACTION DESIRED:
   ```

5. Choose the number of the diskette menu. Here, it's 3, so type

   ```
   3 ↲
   ```

   A few seconds pass; then the hardware formatter displays the diskette menu:

   ```
       DISKETTE UTILITY MENU
      ======================

   1. FORMAT DISKETTE
   2. FORMAT AND VERIFY DISKETTE
   .
   .
   SELECTION OPTION (1..n)
   ```

6. Hardware formatting takes 30 seconds for a 368-Kbyte diskette. A format with verification (VERIFY) takes an additional 5 minutes. When the system verifies the diskette, it checks to make sure that the diskette is in good condition. If you are storing important data on the diskette or if you are using the diskette for backup, you should verify the diskette to protect yourself from possible loss.

   To do a format with verification, type

   ```
   2 ↲
   ```

   The system displays the following

   ```
       DISKETTE FORMATTER
      ====================

       *** WARNING ***
   THIS PROGRAM DESTROYS ALL DISKETTE DATA

   Remove all diskettes from drives to prevent data loss.

   SELECT DRIVE (0,1)
   ```

7. Just to be prudent, remove the Customer Diagnostics diskette from the primary unit. It needn't be inserted for you to format diskettes (although it is needed if you want other operations).

8. Insert the diskette you want to hardware format in a unit, say the primary (right) one.

9. Specify the diskette unit either by typing `0` ↵ (for the right unit) or `1` ↵ (for the left unit). For example,

   ```
   0 ↵
   ```

   ```
   Select DG 9-sector or IBM 8-sector format (DG,IBM):
   ```

10. You must use DG 9-sector format with any operating system, except CP/M-86 which will work with both DG 9-sector and IBM 8-sector format. So you'll nearly always answer

    ```
    DG ↵
    ```

    The diskette formatter now tries to hardware format the diskette:

    ```
    FORMATTING DISKETTE IN DRIVE n
    ```

    If the hardware formatter encounters an error — like a misinserted diskette, or no diskette inserted in the unit — it displays *`UNIT n FATAL ERROR...`* and *`Hit any key to continue.`* Remove and re-insert the diagnostic diskette, and return to step 5 — skip step 8.

    After 30 seconds or so, it displays

    ```
    FORMATTING FINISHED
    ```

    If you select the "Format and Verify" option, the hardware formatter displays a *`VERIFYING`* message and spends 5 minutes verifying the diskette. Then it displays

    ```
    VERIFY FINISHED
    n SOFT ERRORS
    n PERMANENT ERRORS
    ```

    and it returns to the diskette utility menu.

11. The diskette is now hardware formatted. Remove it from the unit, and apply a diskette label with a note to indicate the date and that it is hardware formatted.

    To hardware format another diskette, return to step 5. To stop the Diagnostics program, type the break sequence, (press `CMD` and `BREAK/ESC`).

    You can use the diskette(s) you've just hardware formatted for backup using the backup macros. If you want to use any of them for an AOS directory, continue to the next section. To copy a diskette, skip to the copy section.

Software Formatting a Diskette

When you software format a diskette, you make it into an AOS directory. To make a diskette into an AOS directory, access the directory to which you want it to be subordinate, insert your diskette in the right (primary) unit, run the software formatter (DFMTR) as described in Chapter 5, and open the diskette. After the diskette is open, the diskette acts like an ordinary AOS directory, allowing you to write files to it and read files from it until you close it or the system shuts down.

In the following example, a user makes the diskette into a directory that is subordinate to the user directory (:UDD:JACK). The user first accesses the user directory, software formats a diskette with the software formatter (DFMTR), then opens the diskette. In the formatting process, the user has named the diskette MEMOS. At this point, other AOS files can be moved to the diskette. After you open a diskette, you can access it until you close it.

```
) DIR :UDD:JACK ↲

) X DFMTR ↲

AOS DISKETTE FORMATTER (DFMTR) REV x.xx

DO YOU WANT TO SAVE ANY INFORMATION ON YOUR DISKETTE
TYPE Y (YES) TO SAVE INFORMATION, OR
   N (NO) TO ERASE YOUR DISKETTE.    N ↲

THIS WILL ERASE ALL FILES FROM YOUR DISKETTE.
TYPE Y (YES) IF YOU ARE SURE YOU WANT TO DO THIS, OR
TYPE ANYTHING ELSE TO START OVER.    Y ↲

MOUNT DISKETTE TO FORMAT, AND TYPE NEW LINE WHEN READY.    ↲

DISKETTE NAME (1 TO 31 CHARS) [ ]   MEMOS ↲

DO YOU WANT THIS DISKETTE TO BE PROTECTED?
TYPE Y FOR YES, N FOR NO.    N ↲

ANALYZING YOUR DISK FOR BAD BLOCKS
NUMBER OF OPERATIONS REMAINING: n
```

(3 minutes pass)

```
TOTAL BAD BLOCKS: 0

DONE!

) OPEN ↲
Diskette has been OPENed as
:UDD:JACK:MEMOS            (Diskette pathname)
```

Copying a Diskette

When you copy one diskette to another in AOS, both diskettes must be hardware formatted. The source diskette can, but need not be, an AOS directory. For example, you can copy the Customer Diagnostics diskette, which is not in AOS format.

Copying Using Two Diskette Units

When you copy from one diskette to another, the data from the source diskette overwrites the data on the destination diskette. Therefore, all of the original data on the destination diskette becomes deleted. To make sure you don't accidently overwrite the source diskette, you can write protect it (with tape) as described in "Handling and Storing Diskettes" in Chapter 7.

To copy a diskette, follow these steps.

1. Insert the source and destination diskettes in the diskette units. As a rule of thumb, put the source diskette in the primary (right) unit, devicename `@DPM0`. Put the destination diskette in the secondary (left) unit, devicename `@DPM1`. Don't try to open either diskette.

2. Type the COPY command, using the format:

   ```
   COPY destination-diskette source-diskette
   ```

   For example,

   ```
   ) COPY @DPM1 @DPM0 ↲
   ```

   After approximately 2 minutes, the CLI) prompt returns.

3. The copy is done; you can remove both diskettes from their units. Be sure to remove the write-protect tape that you just put on the diskette if you wish to write to your diskette.

Copying Using One Diskette Unit

To copy one diskette to another using one unit, you will copy the source diskette to a file on the hard disk, remove the diskette, insert the destination diskette, and copy the disk file to the destination diskette.

You destroy all the data on the destination diskette (replace it with the data on the source diskette). So, it's important to make sure the destination diskette doesn't contain information you want to keep.

Follow these steps to copy one diskette to another when you want to use only one diskette unit.

1. Insert the source diskette in the primary unit.
2. Type the COPY command using the format:

   ```
   COPY filename @DPMO
   ```

   For example, for the Customer Diagnostics diskette, you might type

   ```
   ) COPY CUST_DIAG @DMPO ⤸
   ```

   After 1 or 2 minutes, the CLI) prompt returns.
3. Remove the source diskette from the unit; insert the destination diskette in the unit. Use the COPY command again, this time with the diskette name as the destination file. Use the format:

   ```
   COPY @DPMO filename
   ```

   For example, as above, type

   ```
   ) COPY @DPMO CUST_DIAG ⤸
   ```

   As before, after 1 or 2 minutes, the CLI) prompt returns.
4. The copy is done. You can remove the copy from the unit.
5. To conserve space, delete the disk file; for example, type

   ```
   ) DEL/V CUST_DIAG ⤸
   DELETED CUST_DIAG
   )
   ```

What Next?

This chapter explained how to hardware format diskettes, make them into AOS directories, and copy them. Now you've finished the substance of this manual. You might want to check the glossary, or error messages (Chapter 15), or simply enjoy using AOS on your DESKTOP GENERATION system.

Responding to Errors and Error Conditions 15

Read this chapter

- if AOS stops with a FATAL ERROR or HARD ERROR message;
- when power returns after a power failure;
- whenever you don't understand what's happening with the system;
- when you want to know what DG phone number to dial for help.

This chapter tries to describe every error message and condition (including no response, ''nothing'') that you may receive while running AOS — and tells you how to recover from the error. It can't cover *all* products: for CP/M, MS-DOS, INFOS II, CEO, MP/BASIC, or other product error messages, see the pertinent product manual.

The major sections are

- Error Messages and Conditions
- Phone Numbers to Dial for Help
- What Next?

Error Messages and Conditions

The following table, Table 15-1, describes AOS and related program errors, alphabetically by message (if any). The table gives the message, the program or situation where it may occur, the cause of the problem, and tells how to recover from the error.

Table 15-1 Error messages and recovery (continues)

Message	Source, Possible Cause(s), and Action
(nothing, or no response)	At power on. The computer may not be receiving power or the computer unit was turned on before other devices in the system. Verify that the computer unit and any separate unit (like a second hard disk unit) are plugged in and that the outlet is live. Turn on power sequentially: first the system console, then the cartridge tape module (if present), the second hard disk (if present), and finally, the computer unit.
*	At power on. The system console is not turned on, or it is not on line, or it is set too dim. Check power, the ON/OFF switch on the back of the screen, and the ON LINE light. To check for dimness, try the control under the right front corner of the terminal.
	At AOS startup, while installing AOS from diskette, when you type 20H to start from diskette. Perhaps the diskette is inserted backwards. Remove and reinsert it, as shown in Chapter 2, step 3.
	At AOS startup (general). perhaps you typed the wrong characters (not 26H). Type the break sequence (CMD and BREAK/ESC keys). This should produce the *!* prompt and you can type 26H. *Don't* type the break sequence if AOS is up —)prompt.
	During AOS operations: Type CTRL-Q (to undo any CTRL-S that suspended display). If this works, you've recovered. If it doesn't work, type CTRL-O to undo any CTRL-O. If this works, fine. If it doesn't work, type CTRL-O again and then read the next text message in this table (During normal AOS operations).
	During normal AOS operations. If AOS is running, and power is still on to the computer, AOS may be deadlocked. Type CTRL-C CTRL-C on the system console. If there is no response or if the system really seems inert, you must break the deadlock and force a shutdown. Type the break sequence (CMD and BREAK/ESC keys), to get the *!* prompt. Type I and 14R next to the *!* prompt. Then continue as explained under ***AOS PROCESSING ABORTED***.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
!	From the system console loader program. It means that the computer is halted. If you see this when you have just turned power on, or shut down AOS, it is normal: to start AOS, type 26H. Or, you can turn power off. If you see this ! when AOS was running, it means that AOS is frozen; someone may have typed the break sequence. To return control to AOS, type P.
? !	If, with a European keyboard, you get a ? ! message, use the numeric keypad on the far right of your keyboard to type the numbers.
@^^**&^^^ (text stream and beeps)	From the CLI. You may have told the system to type a binary file on your terminal (this can happen if you type TYPE CLI.PR⤶ or a template like TYPE +⤶). When it is displaying a binary file, the terminal ignores most CTRL characters. So, press the COMMAND key (CMD) and simultaneously press ERASE PAGE. If this doesn't restore normal display, type CTRL-C followed by CTRL-A. Then press the CMD and the ERASE PAGE keys again. The CMD/ERASE PAGE sequence tells the terminal to exit from graphics mode; and CTRL-C CTRL-A terminates the command. If this doesn't work, type CTRL-C followed by CTRL-B, and then press CMD/ERASE PAGE to terminate the current process. If the terminal remains frozen, you must turn off the computer unit power and run FIXUP (described under *DISK IS IN USE* error message).
ABNORMAL SHUTDOWN	From AOS, after you shut down: This may mean that you removed an OPEN diskette from its slot without closing it (CLOSE macro); if so, AOS could not close the diskette and the shutdown failed. In any case, start again (26H) and run FIXUP as described under the *DISK IS IN USE* message. If a diskette was removed without being closed when the system was up, you should find it, insert it in unit 0 and run FIXUP on the diskette. (For more information, refer to the OPEN, CLOSE, and FIXUP macros in Chapter 5.)

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
ABORT -- message	From the Disk Formatter (DFMTR) or Installer (INSTL). Retry from the beginning. If this message recurs, it may mean hardware formatting is needed on the disk(ette). For instructions on hardware formatting, refer to Chapter 14, Formatting and Copying Diskettes. If the device involved is a hard disk, you might want to seek help from DG (described later in this chapter). From FIXUP. Find the message in this table. From AOS at startup. There may be a message or number to explain. If there is a number and it's 243, this means you must run FIXUP. Type 26H, then FIXUP↵. From the CLI or EXEC. A utility program hit a fatal error condition and can't continue. If the message allows you to correct the problem, do so. Otherwise, try to find the message that follows *ABORT* in this table.
ACL DELETED, FILE (filename)	From FIXUP. The access control list (that describes the protection status of the file) has been deleted. After you bring AOS up again (or after the diskette is fixed), the original owner should protect this file (PROTECT macro). If the file was supplied by DG, the owner is user OP.
AOS PROCESSING ABORTED STRIKE 'D' FOR AOS DUMP, 'S' FOR SHUTDOWN, OR 'R' FOR OPEN FILE REPORT	You aborted processing via the break sequence and 14R — usually because AOS seemed inert or deadlocked. If the condition that caused you to abort processing is a recurrent condition, you may want to submit an STR to DG (to help get the problem solved). If you want to submit an STR, you must do an AOS DUMP. Press D; then proceed as described under *AOS SYSTEM DUMP*.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
	To skip the dump and simply shut down, press the S key. The ESD routine now tries to shut the system down, *FILE SYSTEM RESTART* *NOW RESTARTING DEVICE 26 UNIT 0* *FLUSHING BUFFERS* *OPEN FILE PROCESSING* *SYSTEM SHUTDOWN* *nnnnnn* *!* And you can reboot AOS (26H, AOS, and so on). If ESD fails (with a *FATAL AOS ERROR* message), then type the break sequence, and restart (26H). It will say *DISK IS IN USE...* Run FIXUP as described under the *DISK IS IN USE...* message.
AOS SYSTEM DUMP *LOAD DISKETTE FOR DUMPING* *STRIKE ANY KEY WHEN READY*	From AOS. A fatal error has prevented AOS from continuing. To do a dump insert a fresh diskette in the primary unit, press a key, and wait for the diskette to be filled. The system console will prompt you for other diskettes as needed. After all memory has been copied to diskettes, the system console will say *STRIKE 'S' FOR AOS SHUTDOWN,* *OTHER KEY TO HALT.* Press the S key. The ESD routine now tries to shut the system down, as shown under *AOS PROCESSING ABORTED.* To *skip* the AOS DUMP and force an emergency shutdown, type the break sequence (CMD and BRK/ESC keys). Then, next to the *!* prompt, type I and 14R. Follow the dialog as shown under *AOS PROCESSING ABORTED.*
ATTEMPT TO ACCESS PROCESS NOT IN HIERARCHY	From the CLI. Either the process doesn't exist (try a WHOS↵ command), or your command cannot be executed because your process is not the father of the target process. In the latter case, use the master CLI (PID 2 on the system console) and try the command again.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
Calendars are unavailable, see your CEO manager.	From CEO, after you type CEO ↵. CEO cannot access its Calendar database. If the CEO Main and Calendar are on a remote system, the cause may be no connection (you may have misdialed for the modem connection), or the remote system is not up with CEO running. Phone the remote system and check. If the remote system is down or not running CEO, you cannot use CEO Mail or Calendar until the remote system starts them up. If everything is okay on the remote system, or if you are running CEO with local Mail and Calendar, the problem is likely on your local system. At the system console, make sure other terminal users are logged off. Type BYE ↵ and DOWN ↵ then UP ↵. For remote CEO, redial to make the modem connection.
Calendar for [username] is unavailable	From CEO. The cause and solution are the same as described under the message *Cannot find your Inbox* , for user *username.*
CALLER NOT PRIVILEGED FOR THIS ACTION	From the CLI. Your command requires a privilege or process ID that you lack (for example, the Superuser privilege, which only the master CLI has). This message often means that your command (or a command issued by your macro) is restricted to the master CLI. For example, only the master CLI, PID 2, can do backups from the root directory or set the system date or time. Using the master CLI on the system console, retry the command.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action

Cannot close specified diskette unit.	From system, after you type `CLOSE ↵` or `CLOSE name ↵`. You do not have a diskette opened in a way that the CLOSE macro can close it. It may also mean that you don't have a diskette open.
	The easiest way to recover is to check to see that the diskette is inserted in the unit in which you opened it. Then enter the following commands:
	`) DIR/I ↵`
	`) FILES/TYPE=LNK ↵`
	`linkname`
	`) CLOSE name ↵` (If the diskette is open) (name is the name that follows *`..floppy.username`*; for example, if the operating system displays *`..floppy.ANN.XXDIR`* then you would type
	`) CLOSE XXDIR ↵`
	You can now remove the diskette from the unit.
	Link names that the operating system uses to open diskettes have the characters *`..floppy`*. If the system doesn't display a name that has ..floppy after you type the FILES command, this means you don't have a diskette open, thus CLOSE isn't needed. Generally, diskette operations are easier if you use only the primary (right) unit. This way, you can just type OPEN and CLOSE without remembering the directory names you used to open.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
Cannot find your Inbox. See your CEO manager.	From CEO, after you type CEO↲. This means that your AOS/CEO profile is wrong or missing. Check with the system manager to make sure that matching AOS and CEO profiles have been created (as in Chapter 2, Figure 2-2). If you are using remote CEO, the remote system, its multiuser environment, and CEO must all be up and running; if they are not running, you can't use CEO Mail and Calendar. After getting things squared away on the system, try running CEO again (type CEO↲).
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. If this message recurs, you might want to seek help from DG (described later in this chapter).
CAN'T DELETE ROOT DIRECTORY	From FIXUP. See *Incorrect disk revision number*, for the hard disk.
CAN'T INITIALIZE LD, RUN FIXUP OVER IT	From the CLI. The diskette was open (OPEN macro) when it was removed from its slot or when AOS shut down abnormally. Thus, AOS was not able to close it. Run FIXUP as follows:) XEQ FIXUP ↲ FIXUP closes the diskette; you can then open it with the OPEN macro. FIXUP for diskettes is detailed in Chapter 5, under FIXUP.PR.
CHECKSUM ERROR	From AOS or a support program. The unit hardware couldn't read the diskette or tape. Retry. If the error recurs, try another diskette/tape or different unit (if possible). For a tape unit, sometimes cleaning the tape or cleaning the unit's read/write heads with an alcohol-soaked cotton swab will help. For a diskette drive, you might try the Customer Diagnostics Diskette Cleaning Program.
CLEANUP FILE DOES NOT EXIST	From EXEC. It can appear after you type UP↲, and originates in a CEO command. CEO may not have been installed properly. Try to create a cleanup file for the printer as described in the appropriate CEO *Managing...* manual.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
CONTROL POINT DIRECTORY MAX SIZE EXCEEDED, FILE file	From the CLI. This message means that your command cannot be completed because the directory is full. If the *file* is subordinate to your user directory, you can enlarge it with SPACE command (for example, via SPACE MYDIR 1000↵). If the *file* is : (the root), this means the hard disk is full. Some files must be deleted for normal AOS operations to continue. Perhaps you can free some space by deleting some of your files. If this is a multiuser system, tell users to delete files they don't need. Users can copy files to diskette to save them if desired, using the full and incremental backup macros. Note that when you restore backed up files, AOS assigns each file *the current date as its creation date*. This means that the FILESTATUS command with the /BEFORE/TLM= switches won't help identify "old" files after the restoration. It also means that your next *backup* must be a full backup (FULL_BACKUP macro). Details on restoring the hard disk appear in Chapter 7, section "Restoring the Hard Disk". If users can't or won't free some disk space, use the master CLI, PID 2 on the system console: turn Superuser on (SUPERUSER ON↵) and delete some old user files. You can delete user directories as shown in Chapter 5, DELETE command.
COULDN'T ACCESS CODE FOR MESSAGE	From the CLI or EXEC. The :ERMES file, which has text for all error codes, is invalid or missing. You'll need to do another installation. If you have the General Language Development Package (GLDP), install from the GLDP diskettes as shown in Chapter 2. Otherwise, install AOS, as described in steps 20 through 25 in Chapter 2. Ignore the *FILE DELETED* messages.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
Currently executing server program will not permit file backup at this time....	From FULL_BACKUP or INC_BACKUP macros. There are several programs (notably INFOS II and CEO) that cannot be running when you do a backup. This is true because the programs have files open, which prevents the newest version of the files from being backed up. (If the open files *were* backed up, the backup would not be reliable.) One way to bring INFOS and CEO processes down is with the DOWN macro — but before typing `DOWN↵`, make sure all CEO and text editor users are logged off. Then try the backup again.
DEVICE ALREADY IN USE	From the CLI. The diskette is open. Close it with the CLOSE macro; then retry desired operation.
Device Error: Check device and press NEWLINE to continue.	From the MMOVE program (executed by the FULL_BACKUP, INC_BACKUP, and RESTORE macros). The diskette has been misinserted (perhaps backwards). Reinsert the diskette and press ↵.
DIRECTORY DELETE ERROR	From the CLI. The directory you tried to delete has subordinate directories. You may not really want to delete this directory. Check all the directories within the directory by typing `) FSTAT/DIR dirname:#↵` Then delete selectively. If you *really* want to delete the directory, repeat the delete command with the # template.
DIRECTORY IN USE -- CANNOT DELETE	From the CLI. The directory you tried to delete is being used; perhaps it is someone's working directory, or it is in someone's searchlist. If you really want to delete the directory, make sure that no one is in it, wait a short while, and try again.
DISK AND FILE SYSTEM REVISION NUMBERS DON'T MATCH	From AOS. The diskette you tried to open was hardware formatted but not software formatted. Run the AOS Disk Formatter (DFMTR.PR) to software format the diskette (see Chapter 5, DFMTR.PR, and Chapter 14, "Formatting and Copying Diskettes" for more information.)

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
DISK ERROR, DEVICE d STATUS n, RETRIES n, LDU ID = id, LOGICAL ADDRESS=n PHYSICAL ADDRESS=n	From FIXUP. FIXUP hit a serious disk error condition. If FIXUP continues (no abort), you may want to run a Disk Formatter SAVE on the disk after it finishes. To do this, specify SAVE (not ERASE). If the Disk Formatter recommends running FIXUP again, do so. If FIXUP aborts, retry it. If it aborts again, take the action described above (Disk Formatter SAVE).
DISK IS IN USE -- MUST RUN FIXUP ON THE LDU	From AOS at startup. When it asks *PROGRAM NAME?,* type `FIXUP`↵ instead of AOS. FIXUP will close the hard disk (which may take several minutes). You'll see messages like *FIXING LDU .... 'xxxx' NOW...* *REPAIR IN ...* *DONE!* If you see a message like *FILE DELETED* , or *FILE REBUILT* , find the message in this table.
Diskette is not in directory format	From the CLI macro OPEN. Either the diskette has not been formatted with the Disk Formatter, or it overwritten by a backup macro since it was formatted. You cannot open the diskette; try another one or run the Disk Formatter (Chapter 5, DFMTR.PR) on it.
Diskette name does not match the name you specified.	From CLI macro OPEN. You specified a diskette directory name, but the diskette has a different name. The macro tells you the real directory name. If the real name is okay, proceed with normal operations. If you don't want to use this diskette, close it (`CLOSE directory-name`). Then replace it with the diskette you think is the proper one; and try the OPEN again.
ERROR message	Look for the *message* in this table.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
ERROR: INITIALIZATION PRIVILEGE DENIED	You do not have write access to this diskette. If you have Superuser privileges, turn on Superuser and open the diskette. If you do not have Superuser privileges, use the software formatter (DFMTR.PR) and indicate that you want to SAVE your files. In this way, you can remove the protection from the diskette. See DFMTR.PR in Chapter 5 for more information.
ESD UNSAFE -- RUN FIXUP	From AOS. It's unsafe to run ESD. The operating system recognizes an error, but can't fix it. To recover, press the CMD and ESC keys. Then restart the system by entering 26H. Enter FIXUP when the system asks for the program name.
EXCEPTION REPORT	See *UNEXPECTED ERROR REPORT...* in this table.
EXEC NOT AVAILABLE	From the CLI. You typed a QPRINT or other command that requires EXEC — but EXEC isn't running. Type UP↵.
FATAL AOS ERROR . *AOS SYSTEM DUMP* *LOAD DISKETTE FOR DUMPING STRIKE ANY KEY WHEN READY*	From AOS. A panic has prevented AOS from continuing. It's not your fault! (unless you turned power off without shutting AOS down first). If you want help from DG, you must do an AOS DUMP and submit the dump diskette(s) to DG. Follow the directions given earlier in this table under *AOS SYSTEM DUMP.* To *skip* the AOS DUMP and force an emergency shutdown, type the break sequence (CMD and BRK/ESC keys). Then, next to the *!* prompt, type I and 14R. Then proceed as described earlier under *AOS PROCESSING ABORTED.*
FATAL DISK ERROR	From FIXUP. If this message follows a *DISK ERROR* message, it means that FIXUP hit a new bad block. Run a Disk Formatter SAVE on the disk (be sure to SAVE all files). The Formatter should find a new bad block. 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 may mean hardware problems. Try FIXUP again. If it fails with the same message, you may want to seek help from DG (described later in this chapter).

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
FATAL ERROR:message	From AOS, while you are bringing up the system. If the *message* is number 25, it means that a needed file wasn't loaded. You may need to repeat the INSTALL procedure from the beginning. If the message is text, try to find it in this table. In either case, try to run ESD by typing `I` and `14R`.
FILE ACCESS DENIED	From the CLI or other program. This means that you don't have access rights to the file — as is normally true of files outside your user directory. If the file *is* in your user directory, type `PROTECT filename↵` and try the command again. If the file is outside your user directory, ask the owner if he or she will unprotect it. AOS system files are owned by user OP — and normally such files should not be unprotected. If you absolutely *must* access the file (perhaps to read or delete it), use the system console and get to the master CLI, PID 2 (type `WHO↵` then `BYE↵` if the PID isn't 2). Then type `SUPERUSER ON↵`, and retry the operation you want with the file. If the file is not in directory :UTIL, or people will need to access it often, you can give everyone owner access to it by typing `UNPROTECT pathname↵`. When you're done, type `SUPERUSER OFF↵` and secure the system console by typing `LOGON username↵`. If this message appears from the SED editor, SED then asks *Start over?* Press ↵ to say no. And read the explanation above.

FILE ALREADY EXISTS	From AOS. You tried to open a diskette, but you have an old link name in your directory. To open the diskette, you must find the old link name and delete it. To find the old link name, enter the following commands:
	`) DIR/I ↵`
	`)files/type=lnk ↵`

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action

	The link name of the diskette appears in one of the following formats:
	`..FLOPPY.username.directory-name`
	`..FLOPPY.username`
	After you enter the following command, you can open your diskette without an error.
	`)DELETE ..FLOPPY.username.directory-name ↵`
	Now open the diskette.

FILE DELETED (FILE filename)	From FIXUP. FIXUP deleted this file because information about it was missing or inconsistent. If you really need the file, restore it from diskette when AOS is up. Chapter 2 tells how to install system files; Chapter 7 tells how to restore user files.
FILE DOES NOT EXIST	While bringing up the system (after it asks *PROGRAM NAME?* and you type a name). The program names you can run at this point are AOS (the system), FIXUP (after abnormal shutdown), and DFMTR or INSTL (when building a system for the first time on a hard disk). From EXEC, when you type UP↵ after running CONFIGURE. You may have made a mistake in the CONFIGURE macro. Type DOWN↵, then CONFIGURE↵ and respecify your AOS system configuration (skip the XODIAC and CEO configuration dialogs, if you are asked about them). Then, type UP↵ again. If you see this message in other situations, like after typing a CLI command, it means that the system could not find the file you specified. You may have mistyped the name. If so, retype or try FILESTATUS with a template character (for example, +). Or try specifying a different directory. *FILE DOES NOT EXIST* is a common error message.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
FILE REBUILT (file filename)	From FIXUP. The file was not intact, and FIXUP tried to reconstruct it. If ***filename*** is a normal AOS filename, then the file retained its original name. But if the filename has the form ?AAAAAAAAAC or ?AAAAAAAAAD or ?AAAAAAAAAE This means that FIXUP had to rename the file. If FIXUP displays one or more *FILE REBUILT (FILE ?AAAA...)* messages, you should check the files after AOS is up again as follows. Using the master CLI (PID 2), type SUPERUSER ON↵. Get into the pertinent directory, and type `*) FILES/AS/S ?AAA- ↵` Have the person rename the file to its original name (if possible). The file retains its original type (PRG for program, CPD for directory, others described in the CLI manual) and its original creation date. The type or creation date may help you identify the file. Don't worry if there are multiple ?AAA- filenames in the directory; just rename each ?AAA file until you have renamed them all.
FILE(S) MAY BE MISSING	From FIXUP. FIXUP found inconsistencies in this directory, and corrected them; but 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 backup listing). If any are missing, restore them from backup diskettes (Chapter 7); or, if system files are missing, INSTALL AOS again (Chapter 2), steps 11 on.
FILE SPACE EXHAUSTED, file	The tape is full; the dump is invalid. Type REWIND @MTC0↵ to rewind the tape, and retry the dump using a template that specifies fewer files.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
FIXUP CHECKSUM ERROR	From FIXUP. The FIXUP disk file is unstable. Get DG-supplied AOS diskette 1, insert it in the primary unit and type 20H. When it asks *PROGRAM NAME,* type FIXUP↵. Run FIXUP and restart AOS. Then type OPEN↵, DIR□name↵ (where name is the name that appears after you type OPEN), MOVE/DEL□:□FIXUP↵, then CLOSE↵; remove and store the diskette.
FORMS DO NOT EXIST	From a CLI QPRRINT/FORMS= command or from CEO. When a user or CEO specifies QPRINT/FORMS=filename, EXEC looks for the forms filename in directory :UTIL:FORMS. If this directory doesn't exist, create it, move the forms files to it, and unprotect the forms files: type CREATEDIR :UTIL:FORMS↵, move forms files to FORMS, then type UNPROTECT :UTIL:FORMS↵. In the future, to create forms or cleanup files, run FCU or CLEANUP in directory :UTIL:FORMS.
From Pid n : CEO.... *EXEC* *INFOS* *X.25*	From CEO, EXEC, INFOS, or X.25 (XODIAC). The program is reporting an exception condition, usually an error condition. If the name is EXEC or X.25, look for the message in this table. Otherwise, the message may be self-explanatory (for example, if your system is connected to a central host and the host shuts down, X.25 will report this). If the message doesn't give enough information to allow you to act, bring the multiuser environment down (DOWN↵) then UP↵ again. If the message recurs, check the message in the manual(s) that describe the other program.
FROM SYSTEM message	Find *message* in this table.
HARD ERROR, DEVICE 020 00n, STATUS = n	From the CLI. This message means that AOS cannot read or write to a diskette. (For *DEVICE 022* or *026* , skip to the pertinent message.) The cause of the diskette error might be a new bad block (flawed area that can no longer hold information) or, the write-enable notch is covered. *DEVICE 020 000* indicates the primary (rightmost) diskette; *DEVICE 020 001* indicates the secondary diskette.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
	Or, the diskette may not be hardware formatted (different from software formatting with the Disk Formatter). Diskettes you buy from DG arrive hardware formatted, but diskettes you buy elsewhere, are not hardware formatted: to do so, use the hardware formatter described in Chapter 14, "Formatting and Copying Diskettes." If the diskette is from DG or if you know that it's hardware formatted (perhaps you've used it with AOS before), then you must software-format it: run the Disk Formatter (Chapter 5, DFMTR.PR) on the diskette. Refer to Chapter 14 for more information on formatting. If you *must* use the diskette (for example, you want to install software from it), remove it from the unit and retry from the first diskette in the set.
HARD ERROR, DEVICE 022 000, STATUS - n	From CLI or AOS utility. *DEVICE 022* means that the problem involves the tape unit. The tape on the unit may be unusable. Retry the command. If the error recurs, try another tape. If it recurs again, try cleaning the tape read/write heads with an alcohol-soaked cotton swab. If this doesn't help, you may want to seek help from DG (described later in this chapter).
HARD ERROR, DEVICE 026 00n, STATUS - n	From the CLI or an AOS utility. *DEVICE 026* means that the problem is on the hard disk. If AOS is still up, it's a good idea to shut it down in an orderly way. If the system console is displaying *FATAL AOS ERROR* , run ESD (type I and 14R). Run the Disk Formatter, 26H, DFMTR↲, and specify a SAVE format as follows: *AOS DISK FORMATTER (DFMTR) REV x.xx* *DO YOU WANT TO SAVE...YOUR DISK [Y/N] Y ↲* The Formatter now checks the disk surface for new bad blocks: *ANALYZING YOUR DISK FOR BAD BLOCKS* *NUMBER OF OPERATIONS REMAINING: m* *TOTAL BAD BLOCKS: n* *A BAD BLOCK WAS ALLOCATED, MUST RUN FIXUP*

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
	Run FIXUP (type 26H, then FIXUP↵). FIXUP will probably display *PART OF FILE MAY BE MISSING* , or, *FILE(S) MAY BE MISSING*. Note the directory and filenames that FIXUP mentions. Bring up AOS if you can, and check the directories and filenames for missing files. You may need to restore one or more missing or damaged files from backup media. If a file is damaged, rename it so that the backup file can be restored. If AOS won't come up, this means that one of the system files was probably damaged. Install AOS again, via Chapter 2, step 11 through the end of the numbered steps. Ignore the *FILE DELETED* messages. This procedure replaces all AOS files and all should be well.
ILLEGAL DUMP FORMAT	From the INSTALL or RESTORE macros or MMOVE program. If followed by the message *Disk 1 must be mounted first,* this message means that the diskette in the primary unit is *not* the first diskette of its set. Replace this diskette with the first diskette in the set, and try the macro again. If the *ILLEGAL* message is not followed by *Disk 1 must be mounted first,* this means that the diskette was not written by an AOS backup macro or MMOVE program. Perhaps the diskette is an AOS directory, created with the Disk Formatter (try to OPEN it). Or, it may have been written with the CLI command COPY or DUMP. In any case, the diskette has the wrong format. Try to find the correct diskette, swap it with the one in the primary unit, and repeat the command.
ILLEGAL FILE TYPE, FILE	From the CLI. The operation you specified is impossible because the file type is wrong. For example, the file you tried to TYPE or XEQ is a directory. This can happen if you use the wrong template with the TYPE command. Check the pertinent file(s) with FILESTATUS (or FSTAT).

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
INCONSISTENT DEVICE INFORMATION BLOCK (DIB) INFO	From the OPEN macro (CLI). The diskette was not formatted as an AOS directory. Perhaps it is a CP/M-86 or MS-DOS diskette, or a diskette used for backup. See action under the message *INCONSISTENT DIB...* , next.
INCONSISTENT DIB INFO, DISK FORMAT INCORRECT	From the Disk Formatter (DFMTR). The Formatter does not recognize the diskette or disk as a directory. This means that · the diskette was never formatted by the AOS Disk Formatter (perhaps it is new or in CP/M-86 or MS-DOS format); or · the diskette was last written to as a device, not a directory (for example, by a backup macro or non-AOS program); or · the diskette or disk was written to by an uncontrolled program. It cannot be used as a directory in its present form. To use it as a directory, you must run an ERASE format on it (rerun and specify ERASE by typing *Y↲* , then *N↲*. If the device is a hard disk with material you want to keep, you might want to seek help from DG (described later in this chapter) before running an erase format.
Incorrect disk format revision number...	From FIXUP. For a diskette — The diskette was not formatted as an AOS directory. Perhaps it is a diskette used for backup, or a CP/M-86 or MS-DOS disk. If not, and you were using the diskette to back up or restore files, the operation is incomplete; and you must restart the backup or restore from the beginning. For the hard disk — This message means a serious problem with the hard disk. You may want to seek help from DG (described later in this chapter).
INTERNAL CONSISTENCY ERROR IN EXEC *...* *HAVE EVERYONE LOGOFF AND THEN TERMINATE THIS EXEC*	From EXEC. EXEC is having problems. BROADCAST a warning message to all users, then type DOWN↲. And try UP↲ again. If this message recurs, you may want to seek help from DG (described toward the end of this chapter).

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
Invalid Remote Username-Password Pair - FILE :NET:....	This means that your username and password are not the same on the remote system as they are on your local desktop system. The username and password must be the same on both systems for the network operation (and CEO) to work. Also, for you to use CEO over the network, there must be a CEO profile with your user ID on both systems. To correct the problem, someone on the remote system must create (or rename) a user profile to have the same username and password as you have on the desktop system. If CEO is involved, the remote system manager must also create a CEO profile with your user ID.
NAME BLOCK ADDR = n *ACL BLOCK ADDR = n* *SYSBOOT BLOCK ADDR = n*	From FIXUP. If FIXUP was running on a diskette,you will probably need to run a Disk Formatter ERASE (erasing all files), before you can use the diskette. If you really need the information on the diskette, or you see this message from FIXUP when running on a hard disk, you may want to seek help from DG (described later in this chapter).
NO CONSOLES ENABLED	From EXEC, when you type UP↵. This is normal if you have no user terminals (only the system console).
NOT A COMMAND OR MACRO	From the CLI. The text string you typed isn't recognizable as a CLI command or macro. This usually means that you made a typing error (for example, you typed TT instead of TY); if so, retype the command or macro text. This message will also appear if you abbreviate a macro name (for example, if you type FST↵ for FSTAT); if so, type the entire macro name. Macro names cannot be abbreviated.
NO SUCH COOPERATIVE	From EXEC, when you type UP↵ after running CONFIGURE. You may have made a mistake in the CONFIGURE macro. Type **DOWN**, then CONFIGURE↵ and respecify the AOS configuration. Don't reconfigure XODIAC or CEO (if asked). Then try UP↵ again.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
Only the master CLI can run xxx	From CONFIGURE, DOWN, FULL_BACKUP, INC_BACKUP, PROFILE, RESTORE, or UP macros. The macro requires privileged status — the process that runs it must be the master CLI, PID 2. On the system console, type WHO↵, if the answer is PID 2, try the macro again. If the answer is not PID 2, type BYE↵, then try the macro again.
PHYSICAL UNIT FAILURE	See *HARD ERROR...* in this table.
@PRINTER COOPERATIVE TERMINATED	From EXEC. The printer managing process, XLPT, stopped abnormally. Printing cannot continue until XLPT is restarted. Make sure users won't lose work; and type DOWN↵ and then UP↵.
PROCESS TERMINATION, PID n	From the CLI. This means that a process (like INFOS, XODIAC, or CEO) has terminated. Normally, you should see this message only after typing DOWN↵ or TERMINATE pid↵. If you see it while the system is running normally, this may mean an essential process has terminated abnormally. You should bring everything down (DOWN↵), then UP↵ again. Warn other users (if any) before typing DOWN↵.
READ ACCESS DENIED	From the CLI or other program. You lack the needed privilege. See the *FILE ACCESS DENIED* message.
Remote Post Office or Network Unavailable	From CEO, after you try to mail something (CEO with remote Mail and Calendar only). The cause and solution are similar to those described in *The Post Office is Closed.*
SOFT ERROR, DEVICE mmm nnn, STATUS = oooooo	From the CLI or AOS utility. This means that a recoverable error occurred on a diskette or hard disk. A read or write operation failed, then succeeded after AOS retried it. An *mmm nnn* of *020 000* , indicates an error on the primary (rightmost) diskette; *020 001* indicates the secondary diskette; and *026 00n* indicates the hard disk. Soft errors are a danger sign — eventually, they can become hard errors. For a diskette used as a directory, you might want to copy the files to the hard disk, format a new diskette, copy the files to the new diskette, and discard the error diskette. If the diskette is being used for backup, replace it before you do the next backup.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
	If this message does not reappear when you try another diskette, it means that the original diskette is bad: throw it away. If the device is *026 nnn* , this means that a hard disk has failed. You may want to seek help from DG (described later in this chapter). On a hard disk, we suggest that you run a Disk Formatter SAVE format. Shut down AOS, type 26H, type DFMTR↵, and respond N↵ to the ERASE question. This may detect some incipient bad blocks.
The Post Office is Closed.	From CEO, after you type CEO↵. CEO cannot acces its Mail database. If the CEO Mail and Calendar are on a remote system, the cause may be no connection (you may have misdialed for the modem connection), or that the remote system is not up with CEO running. Phone the remote system and check. If the remote system is down or not running CEO, you cannot use CEO Mail or Calendar until the remote system starts them up. If everything is okay on the remote system, or if you are running CEO with local Mail and Calendar, the problem is likely on your local system. At the system console, make sure other terminal users are logged off. Type BYE ↵ and DOWN ↵ then UP ↵. For remote CEO, redial to make the modem connection.
This LDU was not in use, fixing is not necessary	From FIXUP. You need not run FIXUP, although running it will do no harm, and may slightly improve file system internal structures. FIXUP gives you a chance to cancel. Type Y↵ or ↵, depending on your wishes.

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
TOO MANY ATTEMPTS, *CONSOLE LOCKING FOR* *10 SECONDS*	From EXEC at logon. You've made several mistakes entering your username and password. No harm done; wait 10 seconds and start again. Or, if you are dialing in to a remote system, hang up, redial, and start again.
TOO MANY BAD DISK BLOCKS	From the Disk Formatter. The Disk Formatter will * say *DEVICE mmm nnn;* if the *mmm* is *020* , this means the device is a diskette: The diskette in the primary unit is unusable. Try another diskette. If this message appears again, there is likely a problem with the diskette unit — you may want to seek help from DG (described later in this chapter). If this message does not reappear when you try another diskette, it means that the original diskette is bad: throw it away. If the device is *026 nnn,* this means that a hard disk has failed. You may want to seek help from DG (described later in this chapter).
UNEXPECTED ERROR REPORT *RETRIES EXHAUSTED* *TIME ....*	From X.25 (XODIAC networking system). The remote system cannot be reached. Perhaps the remote host is not running.
EXCEPTION REPORT *LINK: ASYNC_LCF* *....* *TIME*	Or when you dialed a number over a modem line, but the number doesn't belong to the remote system. If CEO is running, and its Mail and Calendar databases are on the remote host, you may see CEO messages like *The Post Office is Closed.* Exit from CEO when convenient. Then, redial the remote number to reconnect to the remote host. After making the connection, start CEO again (type CEO↵).

Table 15-1 Error messages and recovery (continued)

Message	Source, Possible Cause(s), and Action
UNKNOWN MESSAGE CODE n	From the CLI or a program running under AOS. If *n* is 40131, this probably means the diskette is write-protected. You cannot OPEN and may not be able to access write-protected diskettes from AOS. Remove the diskette from its unit, strip the tape from its write-protect notch, re-insert it, and try again. For any *n* , this message means the program cannot find text in file :ERMES for the error code. For an *n* other than 40131, check and reinstall ERMES as explained in the message *COULDN'T ACCESS CODE FOR MESSAGE.*
Username already exists...	From the PROFILE macro. A profile with this username already exists. Rerun the macro and choose another name. (Or, you can rename the old profile or delete the old profile and recreate it. But if someone is already using the old profile, and has created files in this user directory, either renaming or deleting could cause confusion and/or anger.)
Username is required. Please...	From the PROFILE macro. You must specify a username (and password), 1 to 15 filename characters. Rerun the macro and try again.
USER SPECIFIED FORM DOES NOT MATCH FORM IN PRINTER	From EXEC, on a printed page. Generally, this error occurs when CEO and EXEC have different settings for printer lines per page. If someone has recently changed the maximum number of lines per page (via the CONFIGURE or PRINTER_REDEFINE macros), use PRINTER_REDEFINE to restore the old figure (usually the default, 66).
WARNING message	Look for the *message* in this table.
WRITE ACCESS DENIED, FILE ...	From the CLI. Your command cannot be executed because you don't have write access to the file. You are in someone else's protected directory. If you are trying to open a diskette, check the working directory; and make it your user directory if needed. If you absolutely *must* finish this operation, see *FILE ACCESS DENIED* for action.

Table 15-1 Error messages and recovery (concluded)

Message	Source, Possible Cause(s), and Action
Wrong disk. Please mount disk n and type NEWLINE to continue.	From the RESTORE macro or MMOVE program. Remove the diskette from the primary unit. From the original group of diskettes used for the backup, find the correct diskette *(n)* insert it, and press ↵.
You do not have a CEO profile, contact your CEO manager...	From CEO, after you type CEO↵ or CEO.WP↵. There isn't a CEO profile for your username. Perhaps you made a mistake when you created CEO profiles. Or maybe you didn't create one for yourself. In any case, you must create another CEO profile. Return to the master CLI by typing BYE↵ (if the master is not already running). Then, if you have either the full CEO running locally or the Word Processor - Independent, create the profile ("Creating CEO Profiles", in Chapter 2); then continue from there (LOGON username↵). If you have the full CEO product running remotely, you must run the CONFIGURE sequence again. Don't change the AOS or XODIAC configuration; just create a new CEO profile. Before starting, you may want to phone the central site to verify usernames.
YOU HAVE SONS. DO YOU WISH TO TERMINATE?	From the CLI. Generally, type N↵. Then bring the multiuser environment down in an orderly way by typing DOWN↵. If you type Y↵, it will shut down, but INFOS II and CEO databases may be damaged by the forced shutdown (if you have them).

Getting Help

If you acquired your DESKTOP GENERATION system from an authorized dealer, consult this dealer for help. If you *didn't* acquire your system from an authorized dealer, dial

1-800-DATAGEN

for the latest information on service and maintenance contracts.

What Next?

This — the last chapter in the book — described error conditions, recovery, and phone numbers to dial for help.

Next, you may want to review earlier material, or (more likely) just *use* your desktop system.

Glossary

This glossary describes computer-related terms that may be new to you — either as words or in relation to DG's software products.

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.

Advanced Operating System — see AOS.

access control list (ACL) — a list of privileges, associated with every directory and file, that specifies the type of access allowed for any user. In desktop systems, the access control mechanism has been simplified to the status of protected and unprotected. Protected files (PROTECT macro) are owned by the user only and accessible to no one else; unprotected files (UNPROTECT macro) are owned by every user.

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.

AOS/VS (Advanced Operating System/Virtual Storage) — DG's Advanced Operating System for MV/Family 32-bit ECLIPSE computers.

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 — 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-US, language-specific characters (for example, the UK currency symbol).

asynchronous line — a communications line that uses an asynchronous protocol to transmit characters. In such a protocol, each character has its own "framing" information: traditionally 1 start bit (before the character) and 1 stop bit (after the character). Asynchronous lines are generally used for terminals and for intersystem communication.

backup — files copied for safekeeping, usually onto magnetic diskettes or tape.

bad block — on the magnetic surface of a disk or diskette, a bad block is a flawed area that won't hold information. The Disk Formatter notes such areas so the operating system will avoid them. If AOS encounters a new bad block, it displays a ***HARD ERROR*** message.

BASIC language — Beginner's All-purpose Symbolic Instruction Code, an easy, interpreted language, originally developed at Dartmouth College. DG has several BASICs: MP/BASIC (MicroProcessor/BASIC), a modern BASIC shipped with AOS on desktop systems; Business BASIC, with special business and ISAM capabilities; and Extended BASIC, a traditional BASIC.

batch — the technique of processing in a continuous, noninteractive stream. You can tell AOS to run an operation in batch via the QBATCH command (described in the CLI manual). Batch jobs do not require a terminal and execute without human interaction (for example, overnight); they are suitable for big, well-defined tasks, like large sorts.

baud — the rate at which a line or modem can transfer data, in bits per second. Normally, each character requires 10 bits, so characters are transferred at 1/10 the baud rate. The standard (and default) baud rate for terminals is 9600 (960 characters per second). For modems it is 1200. For communication lines that are directly connected, the default is 4800 baud.

database — an information structure (usually kept in one or more files) that a program requires for proper operation. For example, CEO has a database for its Electronic Mail, and another database for Electronic Filing. Each of these databases involves a number of files that interrelate in a way that's meaningful to CEO.

data-sensitive record — A type of record delimited by a special, agreed-upon character. Some standard delimiters are NEW LINE (⤵) and form feed.

deadlock — a condition in which a system is frozen: unable to act, or respond. Usually you must break a deadlock manually (CMD and BREAK/ESC keys, then 14R to shut down).

debugger — a program that allows you to run another program, set breakpoints, stop execution at the breakpoints, and examine and change variables in the program. A debugger can help you find program errors and understand the details of program execution. There are several debuggers, including SWAT for high-level languages, and the assembly language debugger.

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.

delimiter — a special character that ends each record in a file. The system treats all characters up to the next special character as a record. A common delimiter is the ⤵ (NEW LINE) character. When you type a CLI command, the ⤵ you press at the end is the delimiter.

DFMTR — the filename of the Disk Formatter, an AOS utility program.

DG/GATE — a communications product that allows a desktop system to emulate a terminal produced by any of several major manufacturers — enabling you to to dial up and tap into another computer system (for example, an information pool with electronic bulletin boards or closing market prices). DG/GATE can record all intersystem dialog in a disk file for later review or printing.

DG/XAP — a communications product that allows you to transfer files between systems over a communications line.

directory — a file whose sole function is to contain other files. Directories can help you organize and keep track of your files; the system itself uses them for this purpose. (Also see *root, user, UTIL.*.)

disk (hard) — a fast mass storage device, with metal platters that rotate rapidly. The platters have a magnetic coating that is written to and read from. The operating system, all its directories and files, and all user directories and files are stored on hard disk. *

diskette — a flexible disk, with the magnetic coating on plastic, ranging in size from 3 to 8 inches. The standard diskette for desktop systems is 5-1/4 inches — a common size — and holds 368 Kbytes. It is a double-sided, double-density diskette. Other types of 5-1/4 inch diskettes can be read and written via CP/M or MS-DOS.

Disk Formatter — an AOS utility program that formats diskettes and hard disks, and checks disk surfaces for flaws (bad blocks).

document — In CEO, this is an entity, created via a Mail menu or the CEO Word Processor, that contains text. It can be edited, mailed, filed, formatted, printed, or deleted.

drawer — In CEO, each user has a "cabinet" in which he or she can create drawers. Within drawers, he or she can create "folders", and within folders, *documents.* Cabinets, drawers, and folders are office-oriented names for AOS directories, as documents are for AOS files.

dump — In data processing, dump means "to copy". Often the copy is done for safekeeping, as for backup. Sometimes a dump is done to help discover the source of a problem, as in core dump (which copies the computer's memory to diskettes, so that programmers can examine it).

echo — to confirm a character by displaying it. For example, when you type a character on the keyboard, AOS reads it and echoes it on the terminal screen (unless the character is part of a password, which AOS doesn't echo, to preserve privacy).

emulator — a program that enables a terminal or computer system to act like a specific (other) type of terminal.

ESD — a part of AOS that starts an Emergency Shutdown, closing the hard disk and allowing immediate restart of AOS. ESD is primarily useful after an AOS panic (fatal error). You can run it after a panic by typing I and 14R. (ESD is also acronym for ElectroStatic Discharge, which pertains to static electricity, like lightning, and its effect on computers.)

EXEC — an AOS utility program that manages printer queues and user terminals, if any.

fatal error — see *panic.*

file — a collection of information stored under a *filename*. Some device filenames are rigidly defined (for example, @DPM0 for diskette); but user filenames are flexible (also see *name*).

file transfer agent — see FTA.

filename — see *name.*

firmware — instructions that control some aspect of computer hardware. For example, in Model 10/SP computers, the terminal emulator program is a type of firmware. (The system console starts with limited ability (on power up); the emulator enables it for character and graphics handling.) A different type of firmware — which does not lose state when power is cut — is used in the computer itself.

FIXUP — an AOS utility program that closes a disk or diskette, when it has not been closed properly. For a hard disk, an AOS shutdown (normal or emergency) is required to properly close. For an open diskette, a CLOSE command (macro) is required to close properly.

floating-point unit (FPU) — a board that speeds up computations with floating-point numbers (numbers that have a decimal point). A hardware FPU is optional on some DESKTOP GENERATION systems, standard on others. Systems without an FPU use *firmware* for floating-point operations.

floppy — nickname for a diskette.

form feed — a character (CTRL-L) that tells the printer to stop printing on the current page and start at the top of the next page. Typed on the terminal screen, a form feed clears the screen.

Formatting — structures the surface of a diskette, allowing the operating system to read and write to it. Hardware formating creates disk blocks (sectors) on the diskette. All diskettes must be hardware formatted, no matter how you plan to use them. Data General diskettes are preformatted, so you do not need to format them yourself. Software formatting writes tables on a diskette that AOS needs to access the diskette as a directory. It also creates a bad block table that allows AOS to cope with bad blocks on the disk or diskette.

FORTRAN — contraction of Formula Translator, FORTRAN is one of the oldest and most popular programming languages. There are three different FORTRAN compilers available for DESKTOP GENERATION systems: FORTRAN 77 (the most modern FORTRAN, nicknamed F77), FORTRAN 5, and FORTRAN IV.

FTA — This is 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 terminal 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.

GLDP — is the General Language Development Package designed for people who want to write programs in a compiled language like FORTRAN or COBOL. GLDP includes a text editor, a link program, a debugger, and other programs.

Graphical Kernel System (GKS) — a collection of industry standard graphics-creating routines, which allow FORTRAN 77 or PL/I programs to draw pictures. GKS is available with desktop systems.

hang — see *deadlock.*

hardware formatting — see Formatting.

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).

The master CLI has special privileges, like Superuser (which allows it to bypass file access controls). Via the UP macro, the master CLI creates subordinate processes (like EXEC, INFOS and X25). EXEC, which is usually PID 3, creates user processes (the ones that run on user terminals when people log on). Most processes can create other proceses (called sons); if so, the creating process is called the father. The master CLI is the father of EXEC, which is the father of all user processes. When a process terminates, so do all of its sons; thus, if you terminate EXEC (via the DOWN macro), you terminate all user processes on the system.

host — a computer system that's connected to one or more other systems. The system you are on is called the *local* host; any of the other systems is called a *remote* host.

INFOS II — a file management system, supplied with AOS, that lets users create, maintain, and use large databases, via COBOL, PL/I, or FORTRAN 77 application programs. INFOS II is a superset of an ISAM file system. CEO (the full product) requires INFOS II.

Installer — a utility program that installs an AOS operating system on a hard disk. Once the AOS system is running, the INSTALL macro is used to load software.

INSTL — the Installer program filename.

I/O (Input/Output) — The process of reading information from a device into the computer's main memory (input) and/or writing information from memory (output). The input can come from, and the output go to: disk files, diskettes, a terminal, telephone lines, or microwave beams.

ISAM — Indexed Sequential Access Method. This is a file structure used by INFOS II.

Janitor (CEO) — a utility program that deletes documents and cleans up the mail directory.

Kbyte — 1,024 bytes (1,024 characters). A 368-Kbyte diskette holds 368,640 characters.

line (communications) — see asynchronous 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. For example, 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 DESKTOP GENERATION systems, with a large DG system as the central host (hub of the network).

local (item) — an *item* (like a terminal or CEO database) that is managed by your computer system without a communications line. The opposite of local is *remote*. For example, CEO can be configured with a local or remote Mail database. And a local terminal is attached directly to your desktop system, while a remote terminal is attached via a modem.

log in — an alternate term for the phrase "log on".

log on — to pass a recognition procedure and be accepted by a computer system. For example, to log on to a user terminal under AOS, you type your username and your password. The *log-on* concept is designed to provide security and privacy for files, and to prevent unauthorized people from using a computer system.

macro — a sequence of instructions or commands that can be called (accessed) by a single name; it may or may not require arguments. Macros are primarily timesavers, allowing people to write a series of commands only once, then execute them all by one name. CLI macros that simplify certain operations are shipped with AOS.

manager, system — the person who plans and administers an operating system, deciding (among other things) who will be allowed to use the system and what privileges they will have.

master CLI — See CLI.

Mbyte — Abbreviation for megabyte. In terms of computer memory, 1,024,576 characters. Two megabytes of main memory can hold 2,097,152 characters. In terms of disk storage, a megabyte means 1,000,000 bytes. A 15-megabyte disk can hold 15,000,000 characters. A 38.6-megabyte disk can hold 38,600,000 characters.

modem — a device that connects a remote terminal to a computer over a telephone line. One modem is needed at each site. From the remote site, you dial the destination computer's number, wait for a tone, then connect the modem (either via a switch or by inserting the phone receiver into the modem).

monitor — the system console display screen.

mouse — an input device that you move across a flat surface. Movements are translated as coordinates, which a program then uses to move a cursor or draw a picture.

MP/BASIC — see *BASIC.*

MS^{TM}-DOS — a popular operating system for microcomputers, developed by Microsoft Corporation. MS-DOS and some support software are available with AOS on some DG desktop model computers.

multiplexor — a board that allows a computer system to manage a communications line and user terminals. It sorts the incoming signals for the computer, and ensures that the computer's response goes to the right line. The multiplexor for desktop systems is called a *USAM.*

name — AOS filenames can be from 1 to 31 characters, including letters, numbers, underscore (_), period (.), $, and ?. BASIC and FORTRAN variable and array names can be from 1 to 32 characters including letters, numbers, and underscore, but must begin with a letter. COBOL names can be from 1 to 32 UPPERCASE letters, numbers, and dash (-).

network, — a group of computer systems that can communicate via a communications link. Broad-based networks can include different manufacturers' systems. DG's XODIAC network system, with X.25 interface and agents, allows large DG systems to participate in a general Public Data Network (PDN), or private or local area network with desktop systems.

NEW LINE — a character (produced by the NEW LINE key, shown in this book as ↵) that ends a line of text and starts the next line. It terminates commands to the CLI and other programs.

on line — in direct communication with the computer and under its control. For example, when a terminal is on line, the computer reads from the terminal keyboard and writes to its screen. When a terminal is off line, the computer ignores it.

operating system — a large program that manages and operates devices for users and user programs.

operator, system — the person who physically operates a computer system.

page-milliseconds — indicates a process's 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 this figure for processes.

panic, — what happens when an operating system hits a fatal error condition (an error so serious that the system cannot or dares not recover from it). The system console then prints a fatal error message; and you run a program called ESD, then restart.

parent directory — The directory immediately above a directory; for example, the parent directory of :UDD:OP is :UDD; and the parent of :UDD is : (the root).

password — a combination of filename characters that, used in conjunction with your username, allows you to log on to AOS from a user terminal or access a remote system.

pathname — a path, usually including directory names, to a file. For example, `:UDD:JACK:MYDIR:MYFILE` is a pathname.

peripherals directory (PER) — the system directory that holds entries for all devices. Its full pathname is :PER or the prefix @. The prefix @ that you use with devicenames and queuenames specifies the peripherals directory.

PID (Process ID) — the number that identifies a process. PID 1 is the peripheral manager (PMGR). PID 2 is the master (most powerful) CLI process. PID 3 is usually EXEC. Every process has a PID.

PL/I — Programming Language I, a compiled, high-level language.

pregen — a pregenerated (ready-made) version of a program. For example, PREGEN AOS is a version of AOS that has an operating system generated and ready to use. PREGEN is created to simplify using AOS.

PRESENT facility — an information retrieval program, available with the CEO Electronic Office and/or INFOS II, that can obtain stored information and — via TRENDVIEW — create pictures from it.

printer — a printing device. Several printers are available with desktop systems, including the model 4434 printer (up to 160 characters per line), model 4433 printer (up to 233 characters per line), and model 4518 letter-quality printer (up to 203 characters per line). You can have files printed via the CLI command QPRINT, and specify the printer as a listing file via the /L=@LPT switch.

process — an executing program.

profile — see user profile.

program — a series of instructions, translated into binary codes, that the computer can execute. Word processors, text editors, the CLI, and the operating system itself are programs.

protocol — a set of conventions between communicating programs that defines the format and sequence of messages to be exchanged.

pseudo-macro — a CLI construct designed to make macros more useful; returns a value. For example, [!DATE] returns the current date.

queue — a file designed to hold print and batch requests until the printer and system are ready to process them. AOS has a runtime queue directory and file, with pathname :QUEUE:Q. The AOS line printer queue is named LPT.

RDOS — a DG operating system that can run two programs simultaneously.

record — a series of one or more characters written to or read from a file. Further described in Chapter 11.

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.

remote (item) — an *item* (like a system, terminal, or CEO database) that is managed by another computer or by your computer over a communications line. The opposite of remote is *local*. For example, CEO can be configured to run with a remote Mail database. And a remote terminal is one attached to your system via a modem (instead of directly).

RMA — the XODIAC network Resource Management Agent. It allows people on one system to use devices on other computer systems, and to move files to another system. Also see network.

root directory (:) — the system master directory that contains and gives access to all other directories.

search list — a list of directories that the CLI will scan whenever it can't find the specified file in the working directory; established with the SEARCHLIST command. To help simplify things, the searchlist mechanism has been made automatic (and user-invisible) in AOS for desktop systems.

software formatting — see Formatting.

Software Trouble Report (STR) — a formal report, made by a customer to DG through a DG service area or 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, thus need as much information about the problem as possible. Or, instead of reporting errors, an STR can simply offer suggestions.

Software Subscription Service — a service that provides new revisions of AOS and support software as DG creates them. Membership is available with DESKTOP GENERATION systems.

Sort/Merge — a program, supplied with AOS, that allows users to edit records or reorder them in numeric or alphabetical order.

source file — the file that contains the source statements of a program. If the program is written in a compiled language, the source file must be compiled before the program can be run. In BASIC, you can usually just type and run the source file. In a compiled language, the source file is the most important file (more important than the .OB and .PR versions, which can easily be recreated by the compiler and Link program).

STR — see *Software Trouble Report.*

support organization — the DG group or person committed to supply help or support. For desktop systems, for a limited time after purchase, customers in North America can call a DG area resource center for help; the telephone numbers are given in Chapter 14.

SWAT — DG's high-level language debugger, which works with FORTRAN 77 and PL/I.

switch — aside from conventional meaning, a switch is a slash (/) followed by some characters that modify the execution of a command or macro. For example, /L=LPT selects the printer queue as a listing file (instead of the terminal).

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 — the name that identifies some procedure, variable, array name, or location. Often created by users, but sometimes defined by the language or system. For rules on names, see *names.*

synchronous line — a communications line that uses a synchronous protocol to transmit or receive data. Synchronous lines are frequently used in long distance communications between computer systems.

sysgen — is a version of a product that has all its components ready to generate according to your specifications. Sysgen versions of a product are not as easy to use as their pregen counterparts but offer somewhat more versatility because you can generate a system in different ways. There are pregen and sysgen versions of AOS, CEO, and XODIAC.

system console — the terminal connected directly to the computer. User terminals are not connected *directly,* but through a USAM that sorts incoming commands and sends responses to the correct terminal.

tape — a magnetic medium suitable for file backup and mass storage. For desktop systems, the tape unit name is @MTC0 or @TAPE. Tape *files* are numbered sequentially from 0. With a tape on unit MTC0 you could access the second file as @MTC0:1.

template — the word has two meanings. First, a template is 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 the topmost group of keys on the keyboard. These keys are called function keys and the template identifies them. There are different function keys for each product; for example, CEO and SED each have templates for their own key definitions.

terminal — an interactive device with a keyboard for input and a screen for display. The filename is @CONSOLE.

text editor — a computer program designed specifically to help people write and edit text. A text editor is closely related to a word processor, but has fewer automatic text-processing features. Both the CEO Word Processor and SED text editor are available with desktop systems.

TRENDVIEW — a charting package that can create text files and draw charts from these files, or can draw charts from program-created files. It can also produce color graphics on the Model 10/SP color monitor, or plot charts on the plotter. You can have TRENDVIEW draw a plot in one of two ways: from the CLI (after creating a chart file with either TRENDVIEW or the SED text editor); or from within TRENDVIEW.

USAM (Universal Synchronous Asynchronous Multiplexor) — an optional board, model 4463, that can manage one or four asynchronous lines. A USAM line can be used to communicate with another system, run a printer or plotter, or run a terminal (which can be connected directly to the computer or work remotely through a *modem*).

user (as in user, system) — anyone who — in any capacity — uses a computer system.

user directory — the directory created and maintained for each user. It usually becomes the working directory when you log on. Within it, you can create subordinate directories.

user directory directory (:UDD) — the system directory that contains user directories.

User ID — the term for *username* in CEO.

user profile — a disk file with information that allows a person to gain access to a system or program. User profiles that provide access to AOS contain a username, password, and other specifications; you can create them via the PROFILE macro. Profiles that provide access to CEO contain a User ID (which must be the same as the corresponding AOS username), and CEO privilege and other information. You can create a CEO profile from within CEO — path "Utilities", then "Office Manager Functions" from the Main Menu.

username — the name under which a user logs on. The username is also the name of the user directory.

UTIL — the utilities directory, contains many of the utility programs on the system. Its full pathname is `:UTIL`.

utility, utility program — a program supplied by DG to help you use the system; for example, the CLI and Disk Formatter. Some utilities are included with the operating system; others are optional extras.

Virtual Terminal Agent (VTA) — a XODIAC program that enables people to log onto a remote computer system.

warm start — a computer system startup in which computer power has stayed on since the operating system was shut down; usually faster than a cold start, since certain essential programs stay active and need not be reloaded.

Wastebasket (CEO) — a location where CEO places deleted documents. You can retrieve a deleted document from the Wastebasket until someone runs the CEO Janitor program, then the document is gone forever.

working directory — the directory where you currently are; the current directory.

X.25 — the XODIAC network management support process, that runs all other network operations. X.25 is the name of an international standard for intercomputer communications.

XLPT — name of the EXEC's process that manages the printer.

XODIAC — DG's networking system, which allows a large DG system to communicate with one or more desktop systems in a private or local area network. XODIAC also allows large DG systems to participate in a Public Data Network like TELENET.

Index

In this index, the letter "f" means "and the following page"; "ff" means "and the following pages".

For each topic, the primary page reference is listed first. All letters are lowercase, except for program names (like BASIC and CLI, commands switches and keys (like /AFTER), and programming language statements and calls.

For *definitions* of keywords, see the preceding glossary.

Documentation Comment Form

Manual Title ______________________________

Manual No. ______________________________

Your Name ______________________________

Your Title ______________________________

Company ______________________________

Street ______________________________

City ______________________ State ________ Zip ______________

Please help us improve our future publications by answering the questions below. Use the space provided for your comments. Thank you.

	Yes	No
Is this manual 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 wish to order manuals, contact your sales representative or dealer.

fold

Comments:

Date

Data General

Software Documentation
M.S. E-111
4400 Computer Drive
Westboro, MA 01581

artup

If power to system console is off, turn it on. The ON LINE light should glow.

If computer power is off, turn it on. Wait 15 seconds for the disk to become ready.

! **26H** (Type **26H** next to *!*)

WELCOME TO THE DG DESKTOP GENERATION

PROGRAM NAME? **AOS** ↵

(Depending on last program run, wait 10 seconds or more than a minute.)

AOS REV n

DATE (MM/DD/YY) **12 14 84**↵
(Type the current date.)

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

DO YOU WANT TO INSTALL AOS SOFTWARE [N] ?
(Say no by pressing ↵.)

AOS CLI REV n date time
)

If system has a printer, make sure it's turned on and on line. If it has multiple terminals, make sure each one is turned on and on line.

) UP ↵

If you have a communications line configured for a modem, it asks

To use the modem line for communications...
... When ready to continue, press NEW LINE.

Dial up, connect the modem, and press ↵.

The AOS up macro may execute other software up macros:

...(messages from CEO and other processes)

↵
)

12. **LOGON your-username** ↵
(For example, **LOGON SALLY**↵.)

AOS CLI REV n date time
)

The system is up — ready for you to run programs (like CEO) on the system console. People can log onto other terminals (if any), with username and password.

Shutdown

If no one is logged onto user terminals (other than the system console), start with step 3.

1. **)WHOS** ↵
. . (Check for user processes.)

) BROADCAST System coming down — log off. ↵

2. Use the WHOS and BROADCAST macros until all :UTIL:CEO_CP.PR, :UTIL:SED.PR, and :UTIL:MBASIC:MBASIC.PR users are out of their programs (all such PIDs are gone).

3. **) BYE** ↵

AOS CLI TERMINATING date time
You are now
PID 2 OP OP :CLI.PR (Back in the master CLI.)
)

4. **) DOWN** ↵
. (Messages about CEO, INFOS, XODIAC if you
. have them. If you see an error message,
. consult Chapter 14.)

↵

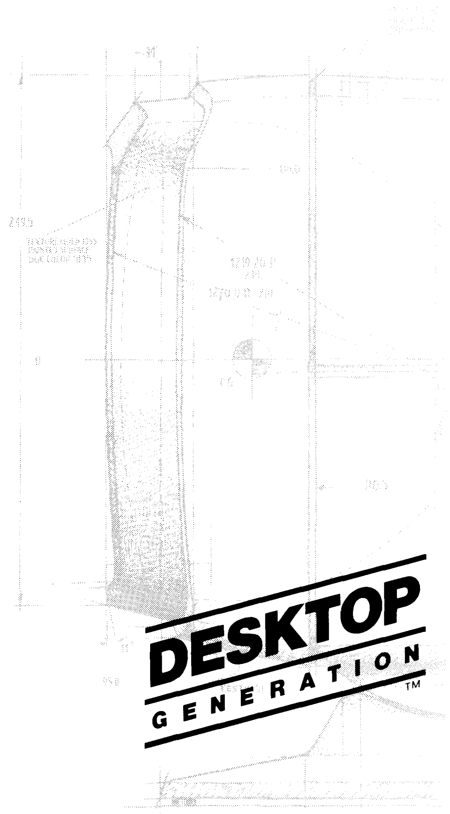
5. Check for processes other than 1 and 2 with **WHOS.** If any of your own site's programs are still running, terminate them. Then,

) BYE ↵

6. *DO YOU REALLY WANT TO SHUT THE SYSTEM DOWN?*
Y ↵
SYSTEM SHUTDOWN
!

Turn off computer power if you want. Disconnect modem and hang up if this applies.

Data General



DESKTOP GENERATION™

AOS
Summary Card

Commands, Macros, and Programs

BROADCAST message
Send a message to all system users.

BYE
Log off a user terminal or shut down the system.

CLOSE *[diskette-directory-name]*
Close an opened diskette before you remove it from its slot.

CONFIGURE
Identify system devices.

COPY destination-pathname source-pathname *[source-pathname] [...]*
Copy to a file one or more files.

CREATEDIR directory-pathname *[maxsize]*
Create a directory.

CREATETEXT pathname
Create a file and insert text in it.

DATE *[dd:mon:yy]*
Display or change the system date.

DELETE pathname *[pathname] [...]*
Delete one or more files.

XEQ DFMTR
Format a diskette for use as an AOS directory.

DIRECTORY *[directory-pathname]*
Display or change the working directory.

DISABLE7BIT
Restores 8-bit character set.

DOWN
Shut down the multiuser environment.

ENABLE7BIT
Enables 7-bit character set.

EXECUTE pathname *[argument]*
Execute another program.

FILESTATUS *[pathname] [pathname] [...]*
Describe filenames and statistics.

XEQ FIXUP.PR
Close a diskette that was open at abnormal shutdown.

FSTAT *[pathname] [pathname] [...]*
Describe filenames and statistics alphabetically.

FULL_BACKUP
Copy all files to (one or more) diskettes for backup.

HELP
Give help information on topics and CLI commands.

INC_BACKUP
Copy files to a diskette for backup, based on date.

INSTALL product
Start installing another DG software product.

LOGON username
Run a CLI user process on the system console.

XEQ MMOVE/DUMP*[/FROM]* diskette-unitname *[source-pathname] [...]*
Copy one or more files to diskette for safekeeping.

MOVE destination-pathname *[source-pathname] [source-pathname]*
Copy one or more files to a different directory.

```
      ┌                  ┐
      │@DPM0             │
      │@DISKETTE1        │
OPEN  │@RIGHT_DISKETTE   │  [diskette-dir-name]
      │@DPM1             │
      │@DISKETTE2        │
      │@LEFT_DISKETTE    │
      └                  ┘
```

Open a formatted diskette as a directory.

```
PERMANENCE pathname  [ON ]
                     [OFF]
```

Display or set permanence for one or more files.

PRINTER_ALIGN
Stop a printer so you can align paper, then restart it.

PRINTER_CONTINUE
Continue printing after a PRINTER_STOP command.

PRINTER_REDEFINE
Change number of characters per line and lines per page.

PRINTER_STOP
Stop a printer immediately.

PROFILE
Create, delete, or rename a user profile.

PROTECT pathname
Allows only you to access files that you own.

QCANCEL *[sequence-number]*
Cancel or abort a printing job or batch job.

QDISPLAY
Describe jobs in the batch and print queues.

QPRINT pathname *[pathname] [...]*
Place one or more files on the printer queue.

RENAME pathname new-filename
Rename a file.

RESTORE *[pathname]*
Restore files from backup media to the hard disk.

RUNTIME *[pid]*
Describe a process's life span and overhead.

SETUP
Set up the environment for each user at logon.

SPACE *[directory-pathname]*
Display the amount of disk space used and remaining

```
SUPERUSER  [ON ]
           [OFF]
```

Turn on superuser privilege to bypass file access co

TERMINATE pid
Kill a process.

TIME *[hh] [mm] [ss]*
Display or change the system time.

TYPE pathname *[pathname] [...]*
Type one or more files on the terminal screen.

UNPROTECT pathname
Allow any user access to a file that you own.

UP
Start up printers, other DG products, and the multiuse environment.

WHO *[pid]*
Describe a process.

WHOS
Describe all processes on the system.

WRITE *[argument] [argument] [...]*
Display arguments on the terminal or write them to a

XEQ pathname *[argument]*
Execute a program.