

Using OpenPC for AViiON™ Systems

Using OpenPC for AViiON™ Systems

069-100267-00

Ordering No. 069-100267-00
Printed in the United States of America
Rev. 00, January 1990

NOTICE

DATA GENERAL CORPORATION (DGC) HAS PREPARED AND/OR HAS DISTRIBUTED THIS DOCUMENT FOR USE BY DGC PERSONNEL, LICENSEES, CUSTOMERS, AND PROSPECTIVE CUSTOMERS. THE INFORMATION CONTAINED HEREIN IS THE PROPERTY OF THE COPYRIGHT HOLDER(S); AND THE CONTENTS OF THIS MANUAL SHALL NOT BE REPRODUCED IN WHOLE OR IN PART NOR USED OTHER THAN AS ALLOWED IN THE APPLICABLE LICENSE AGREEMENT.

The copyright holders reserve the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases 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, AND THE TERMS AND CONDITIONS GOVERNING THE LICENSING OF THIRD PARTY SOFTWARE CONSIST SOLELY OF THOSE SET FORTH IN THE APPLICABLE LICENSE AGREEMENT. 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.

All software is made available solely pursuant to the terms and conditions of the applicable license agreement which governs its use.

Autocad is a trademark of Autodesk, Incorporated.

Autosketch is a registered trademark of Autodesk, Incorporated.

AViiON, AVX-30, and Data General are trademarks of Data General Corporation.

CEOWrite is a trademark of the Data General Corporation.

Crosstalk is a registered trademark of Microstuf, Incorporated.

Data Ease is a trademark of Data Ease International, Incorporated.

dBaseIII is a trademark of Ashton-Tate Corporation.

DEC, VAX, and VT are trademarks of Digital Equipment Corporation.

Desqview is a trademark of Quarterdeck Office Systems.

DG/UX is a trademark of Data General Corporation.

D216+, D412+, and D462+ are trademarks of Data General Corporation.

EDIX is a registered trademark of Emerging Technology Consultant, Incorporated.

Ethernet is a trademark of Xerox Corporation.

Fastback is a trademark of Fifth Generation Systems, Incorporated.

Framework is a trademark of Ashton-Tate Corporation.

GraphWriter is a registered trademark of Lotus Development Corporation.
GrandView is a registered trademark of Symantec Corporation.
GW-BASIC is a registered trademark of Microsoft Corporation.
Harvard Graphics is a trademark of Software Publishing Corporation.
Harvard Total Project Manager is a trademark of Software Publishing Corporation.
Hercules is a registered trademark of Hercules Computer Technology, Incorporated.
IBM, PC AT, and CGA are registered trademarks of International Business Machines Corporation.
Intel is a registered trademark of the Intel Corporation.
Javelin PLUS is a trademark of Javelin Software Corporation.
Landmark Speed Test is a trademark of Landmark Software.
Leading Edge is a registered trademark of Leading Edge Products, Incorporated.
OpenPC is a trademark of Phoenix Technologies Ltd.
LaserJet is a trademark of Hewlett-Packard Company.
Lotus, Lotus 123 and Lotus Symphony are registered trademarks of Lotus Development Corporation.
Microsoft, Microsoft Windows, Microsoft Flight Simulator, MS-DOS, and Xenix are registered trademarks of Microsoft Corporation.
mips is a trademark of MIPS Computer Systems, Incorporated.
Mirror II is a registered trademark of SoftKlone, Incorporated.
Multimate is a registered trademark of Multimate International Corporation, an Ashton-Tate Company.
NFS is a trademark of Sun Microsystems, Incorporated.
Norton Utilities is a trademark of Peter Norton Computing, Incorporated.
Office Writer is a trademark of Office Solutions, Incorporated.
OSF/Motif is a trademark of Open Software Foundation.
Pagemaker is a registered trademark of Aldus Corporation.
PFS is a registered trademark of Software Publishing Corporation.
R:Base is a registered trademark of Microrim, Incorporated.
SAS/Graph and Base SAS are registered trademarks of SAS Institute, Incorporated.
Sidekick is a registered trademark of Borland International, Incorporated.
Softerm is a registered trademark of Softronics, Incorporated.
SuperCalc is a registered trademark of Computer Associates International, Incorporated.
Superkey is a registered trademark of Borland International, Incorporated.
Time Line is a registered trademark of Symantec Corporation.
TopView is a trademark of International Business Machines Corporation.
Turbo C is a registered trademark of Borland International, Incorporated.
Turbo Pascal is a registered trademark of Borland International, Incorporated.
20/20 is a trademark of Access Technology, Incorporated.
UNIX is a registered trademark of AT&T.
Ventura Publisher is a registered trademark of Xerox Corporation.
WordPerfect is a registered trademark of Word Perfect, Incorporated.
Wyse is a registered trademark and WY-60 is a trademark of Wyse Technology.
X Window System is a registered trademark of Massachusetts Institute of Technology.
XyWrite is a registered trademark of XYQUEST, Incorporated.
All other trademarks acknowledged.

Copyright © Phoenix Technologies Ltd., 1989

Copyright © Data General Corporation, 1989

Using OpenPC for AViiON Systems

All Rights Reserved
Printed in the United States of America

Certain portions of this document were prepared by Data General Corporation and the remaining portions were prepared by Phoenix Technologies Ltd.

Using OpenPC for AViiON Systems
069-100267-00
069-100268-00 (Japan only)
Effective with MS-DOS, Revision 4.01
Effective with DG/UX 4.20

WARNING: Multi-user capability may violate licensing requirements of DOS applications. Please review your DOS application license agreement before using your DOS application with OpenPC.

Preface

Purpose

This manual provides installation, administration, and usage information for the OpenPC software product.

Prerequisite Knowledge

Readers should have previous familiarity with the UNIX (or DG/UX) operating system and with DOS, as well as with the documentation that accompanies those operating systems. Readers should also have some familiarity with IBM PC AT-class computers.

How to Use This Manual

All users of OpenPC, including system administrators, should read Chapter 1 for an overview of what OpenPC is and how it works. Although OpenPC is ordinarily easy to use, it is a complex software product and involves the interaction of two different operating systems. You should become familiar with the new concepts and terms described in Chapter 1 so that you can use the information presented in later chapters.

System administrators should go on to read the rest of the book. Chapter 2 presents the first responsibility of the system administrator: installation. The remaining chapters present essential information for file management, configuration and general use.

If you are a general user of OpenPC, you can skip to Chapter 4 to learn how to use OpenPC to run your DOS applications. Chapter 5 provides further useful information on running DOS applications. If you have questions about using OpenPC, consult your System Administrator and texts on DOS and DG/UX or UNIX.

Applications Licensing

IBM PC ATs are single user machines. PC software applications are ordinarily licensed to a single user who then installs and runs them from the hard disk or diskette on his or her own machine. AViiON systems can be multiuser machines, but still each user must be licensed for each application. As a user of an application it is your responsibility to observe the licensing requirements issued with that application.

Notational Conventions

Angle Brackets Angle brackets enclosing a string of characters indicate a keyboard key. So <ALT> tells you to press and release the key labeled ALT. Likewise, <ALT> <SYS REQ> means hold down the <ALT> key while you press the <SYS REQ> key.

NOTE: On some computers, the <SYS REQ> key may have a different name.

Square brackets Square brackets indicate optional command arguments. For example:

`command [-n] filename`

indicates that the `-n` argument is optional. You do not type the square bracket characters.

<i>Italics</i>	Italics used within examples indicates a general name for what you specifically type. For example: <code>rm <i>filename</i></code> tells you to enter the name of the file that you want to remove. You do not type “filename.”
Bold	Boldface indicates literal strings, file names, and path names where contained in explanatory text, as in <code>cp</code> or <code>dir /w</code> or <code>/usr/opt/dosbin</code> . Note that spaces and/or upper and lowercase letters are significant in literal strings.
Monospace	Monospace (constant width “typewriter”) characters indicate literal strings you either read on your screen or enter, where the strings are spaced apart from the text, like this: <code>fdisk</code> <code>format /s</code>
Command names	Command names in the form <code>command(n)</code> , specify an entry within DG/UX manual reference pages. The <code>command</code> string is the name of the entry and <code>n</code> is the section number. So <code>rm(1)</code> refers you to the description of the <code>rm</code> program in section 1 (commands).

DOS

Throughout this manual, a DOS application indicates a program written for either MS-DOS or PC-DOS. (MS-DOS and PC-DOS are functionally equivalent.)

Mouse buttons



MB1, **MB2**, and **MB3** indicate which mouse button to press. **MB1** is mouse button 1, the button on the left. **MB2** is button 2, the button in the middle. **MB3** is button 3, the button on the right.

X Window System, X Windows, and X11 are used interchangeably in this manual.

Contents

1 Overview

Description of OpenPC	1-1
Features	1-2
Hardware Configurations	1-2
Concepts and Terminology	1-6
Virtual Devices	1-6
IBM PC AT Emulation	1-6
Virtual Disk	1-8
DOS Start-Up Files	1-10
OpenPC Start-Up Files	1-11
X and Serial Terminals	1-12
Editing — Cutting and Pasting	1-12
Idle Detection	1-13
CMOS RAM and ROM BIOS	1-13
Keyboard Mapping	1-14

2 Installing OpenPC

Installing the OpenPC Software	2-1
Checking Installation	2-2
Installing DOS Applications	2-4
Initial User Set-Up	2-5
Removing OpenPC	2-6

3	Customizing OpenPC	
	The openpc.dat File	3-2
	Writing your own openpc.dat file	3-3
	openpc.dat Keywords and Settings	3-6
	Customizing with the OPENPC Variable	3-10
	Example OPENPC	3-10
	DOS Start-Up Files	3-11
	Using the LIM EMS 4.0 Driver	3-12
4	Using OpenPC	
	Starting OpenPC	4-1
	Search Precedence	4-3
	The OpenPC Menu Bar	4-3
	Menu Bar Selections	4-4
	Using opencmd Instead of Menus	4-10
	Example Usage of opencmd in a DOS Batch File	4-12
	Accessing DG/UX Files from within DOS	4-12
	Accessing DG/UX Files Through Redirected Drives	4-14
	Accessing Virtual Disks	4-15
	Private Virtual Disks	4-16
	File Name Mapping Using Redirected Drives	4-16
	Mapping Algorithm	4-18
	Examples of File Name Mapping	4-18
	Printing from OpenPC	4-20
	Using the Mouse (X Windows only)	4-21
5	Using DOS applications	
	Where to Install Your Applications and Store Data	5-1
	Accessing Physical DG/UX Diskettes	5-2
	Installing DOS Applications	5-2
	Using the F10 Key in DOS Applications	5-3

6	Using DOSMOUNT and REDIR	
	Enabling redirected drives: REDIR	6-2
	Assigning Redirected Disks: DOSMOUNT	6-3
	Syntax	6-3
	Examples of DOSMOUNT Usage	6-5
	Example 1	6-5
	Example 2	6-5
	Example 3	6-6
	Example 4	6-6
	Example 5	6-7
7	Using Serial Terminals	
	Terminal Setup	7-2
	Limitations on Serial Terminals	7-2
	Troubleshooting	7-3
A	Unsupported Serial Terminals	
	Required Features	A-2
	Optional Features	A-2
	Option Descriptions	A-3
	Overriding Terminal Definitions	A-4
	Example	A-5
	Terminal Descriptions	A-6
	Necessary Terminal Information	A-6
	Terminal Capabilities	A-7
	OpenPC Capabilities	A-9
	Screen Attribute Definitions	A-9
	Special Keys	A-10
	pccompat	A-11
	ASCII Input to Scan Code Mappings	A-11
	Output Character Mappings	A-14

B Tested DOS Applications	B-1
C Troubleshooting and Error Messages	
Troubleshooting	C-2
Application Installed on a Redirected Drive Fails to Run	C-2
F10 Key in DOS Responds Improperly	C-2
Writing to the Virtual Disk Fails	C-3
Error messages	C-4
(3) Unable to open <FQF> as DOS drive <drive letter>	C-4
(4) Error while closing drive <drive letter>	C-5
(5) Creating file <FQF> because the requested file does not exist	C-5
(6) Insufficient memory available	C-6
(12) Unable to open <FQF> as DOS drive <drive letter>	C-7
(15) Unable to use DOS printer device <printer number>	C-8
(16) Unable to write to DOS printer <printer number>	C-9
(17) Error while closing destination printer <printer number> ..	C-9
(18) Cannot load ROM file: <FQF>	C-10
(19) Cannot open font 	C-11
(20) Insufficient memory available	C-12
(21) No mapping specified for the key <key number>	C-13
(23) Cannot open keyboard file: <FQF>	C-14
(24) Insufficient memory available0	C-14
(25) Error encountered using COM device	C-15
(26) Error encountered using COM device	C-16
(27) Insufficient memory available	C-17
(28) Unable to use DOS COM device <FQF>	C-18
(29) Error encountered using COM device	C-19
(30) Error encountered using COM device	C-20
(32) Insufficient memory available	C-21
(34) Error encountered using COM device	C-22
(35) Error encountered using COM device	C-23
(36) Error encountered using COM device	C-24

C Troubleshooting and Error Messages, Continued

Error messages, cont'd

- (37) Error encountered using COM device C-25
- (38) Error encountered using COM device C-26
- (57) Opening file <FQF> with Write access
even though lock request failed C-27

Glossary G-1

Index I-1

Overview

This chapter provides an overview of OpenPC™. The hardware and software environments and key concepts are briefly described.

To learn the details on how to install, configure and run OpenPC, refer to later chapters. The default information for OpenPC configurations is in Chapter 3.

Description of OpenPC

OpenPC is a software application that creates a virtual IBM® PC AT® environment on the AViiON™ DG/UX™ operating system as a DG/UX process. This environment supports MS-DOS® and PC-DOS® applications, allowing them to run concurrently under DG/UX.

DOS is the traditional operating system for an IBM PC or PC-compatible machine. OpenPC lets users on AViiON workstations access and run most commercial DOS and other PC AT operating system applications (for example Microsoft® Flight Simulator®, which runs on a PC but does not require DOS). With OpenPC, you can run UNIX® and DOS applications on the same machine without additional hardware.

Examples of some popular DOS applications which you can run using OpenPC are

- Lotus® 1-2-3®
- Microsoft® Word
- WordPerfect®

A sample list of applications tested under OpenPC is in Appendix B.

Features

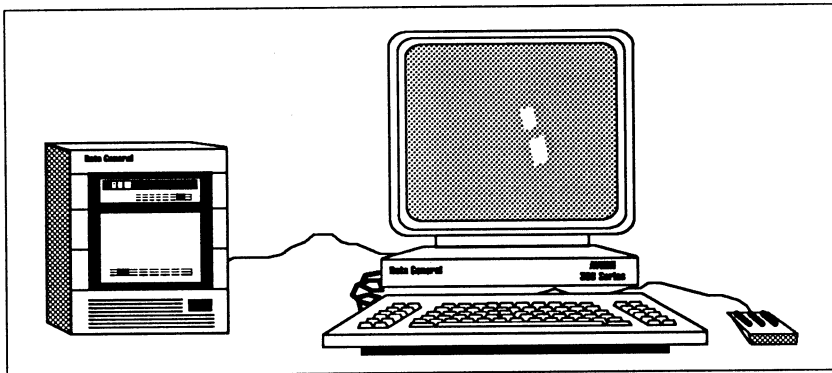
The OpenPC software provides:

- full PC AT compatibility for running DOS and applications under DG/UX.
- Phoenix AT-compatible ROM BIOS (industry standard for legal emulation).
- support for running under the X Window System®.
- wide or normal IBM PC AT fonts under the X Window System.
- Microsoft Bus Mouse emulation.
- LIM EMS (Lotus/Intel®/Microsoft Expanded Memory Specification) 4.0 support including full memory mapping.
- 80287 floating point coprocessor emulation.
- video emulation of CGA®, MDA, and Hercules® graphics.
- transparent file sharing between DOS and DG/UX via OpenPC's Redirector utilities.
- operation on Data General serial asynchronous terminals D216+™, D412+™, and D462+™.
- support for OSF/Motif™ user interface.
- diskette drive support.

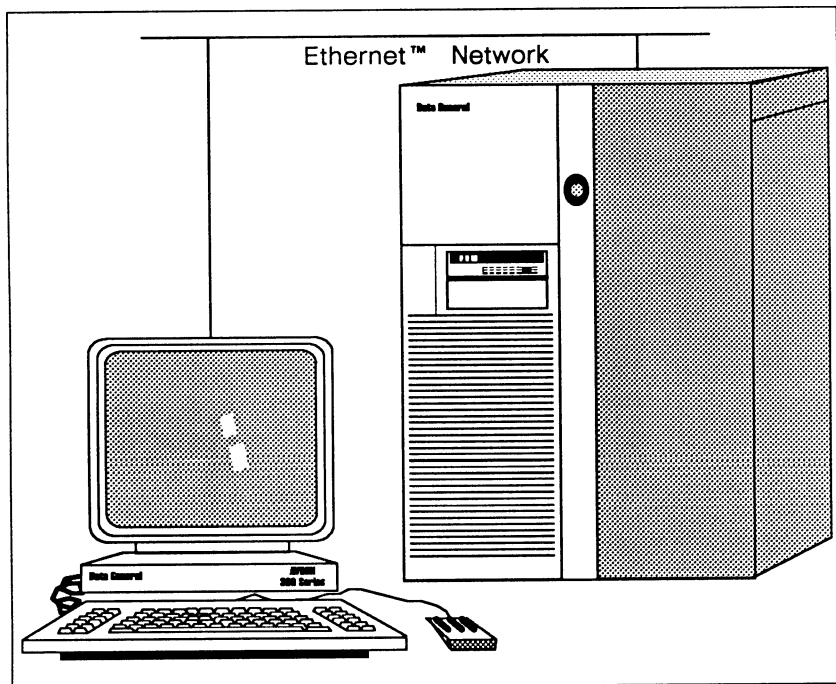
Hardware Configurations

The following diagrams represent possible hardware configurations of Data General hardware using OpenPC.

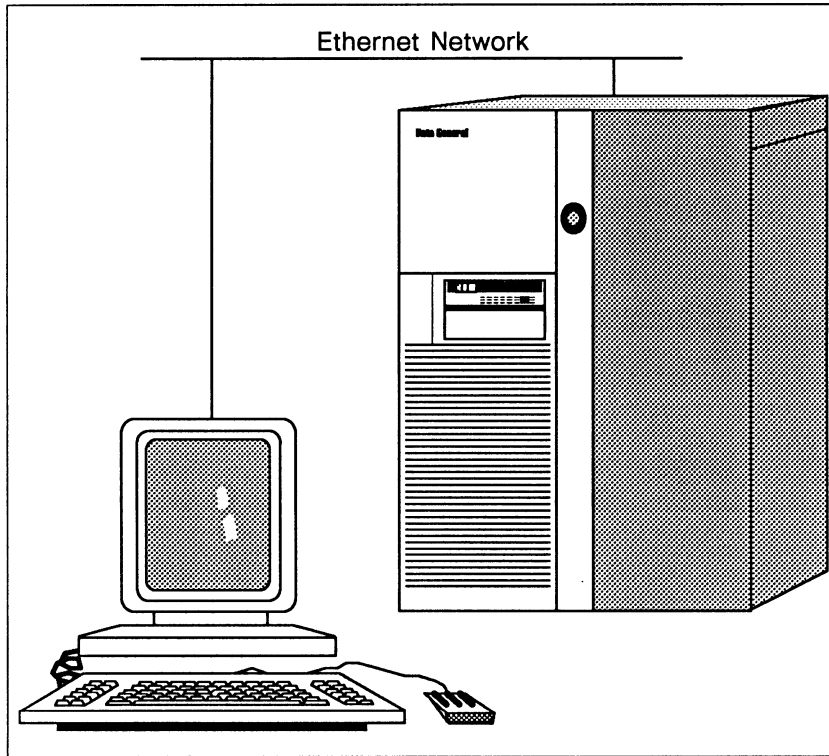
The following illustration shows a standalone (diskful) workstation, color or monochrome.



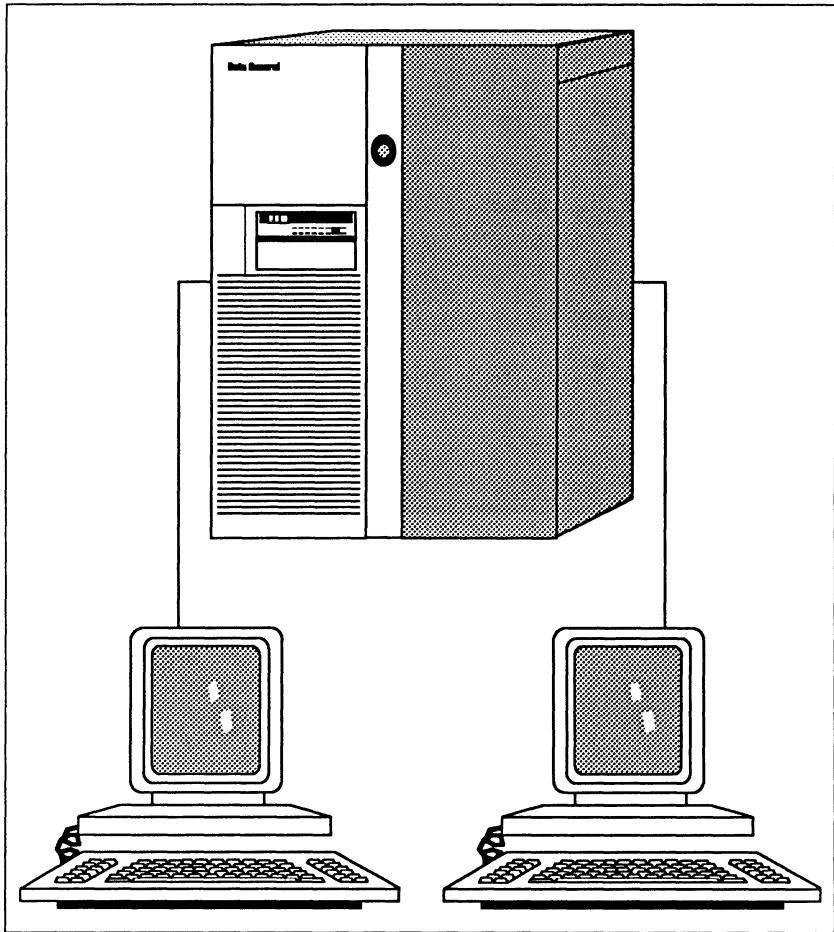
The illustration below shows a diskless workstation, color or monochrome, networked to a server.



The following illustration shows an X terminal and server.



The illustration below shows serial terminals and server.



Concepts and Terminology

OpenPC is complex software because it bridges two different machines and operating systems. Even if you are already familiar with DOS and DG/UX (or UNIX), OpenPC introduces new concepts and terminology which you must understand to get the maximum use out of this product. Once you learn these concepts, OpenPC is easy to use and extends your software base significantly.

Virtual Devices

OpenPC emulates an IBM PC AT configured with an extensive set of devices such as serial ports, parallel ports, and video cards. These devices are called virtual devices in the context of OpenPC. This simply means that although the physical PC AT devices are not present in the AViiON environment, the OpenPC software emulates them. Thus IBM PC AT programs act just as if they were using actual PC AT devices, even though the devices are really virtual devices using the services of the AViiON system.

IBM PC AT Emulation

OpenPC emulates an IBM PC AT with the following configuration:

- Intel 80286 CPU operating in real mode. This means that OS/2 and 80386-specific DOS programs will not run under OpenPC.
- Intel 80287 math coprocessor.
- Hercules adapter card installed. This video card allows graphics on a monochrome monitor (as well as maintaining text-mode monochrome functionality).
- Color Graphics Adapter (CGA) installed. This video adapter card drives the IBM Color Graphics Display. On OpenPC, color is translated into various patterns of black and white so that color information is preserved and displayed on a monochrome AViiON display or AVX-30™ display. OpenPC displays full color on a color AViiON workstation.

- Serial communications card installed. This card enables serial communications capability, for example, to use a modem, on the IBM PC AT. The serial port is called COM1. OpenPC uses a user-designated DG/UX tty port to provide this function. Both COM1 and COM2 ports are supported.
- Parallel printer support. OpenPC allows printing from the DOS environment to DG/UX files, devices, or printing facilities.
- Microsoft Bus Mouse installed. OpenPC lets you convert the AViiON mouse into a Microsoft Bus Mouse that DOS applications can then access.
- Expanded memory installed. On an IBM PC AT, DOS sets a 640 KB limit on application program size and data. This limit is increased by using the Lotus/Intel/Microsoft Expanded Memory Specification (LIM EMS). OpenPC lets you designate the amount of expanded memory that may be used by DOS applications.
- Two 20 MB hard disks installed. OpenPC emulates two IBM PC AT-type hard disks by using the DG/UX file system for data storage. These are called “virtual disks” and are described in the next section.
- Two 1.2 MB diskette drives installed. OpenPC can emulate IBM PC AT diskette drives using the DG/UX file system for data storage. Or you can use the physical AViiON disk drive to read, write, and format PC AT diskettes.
- 101-key keyboard installed. OpenPC uses the AViiON or AVX-30 keyboard in the same manner as an IBM PC AT.

In addition to the above functions, which emulate a fully-configured IBM PC AT, OpenPC adds the following features:

- Graphical interface to allow control of OpenPC as an X Window System client when run under X Windows. This includes cut and paste capabilities.
- File system redirection so that the DG/UX file system can be shared for DOS applications and data. This avoids the need for re-partitioning the DG/UX hard disk into DG/UX and DOS partitions.

- Idle detection to allow OpenPC to idle itself during periods when it is not being actively used. This lessens the load on the host CPU and allows more equitable distribution of the CPU time among all the processes running under DG/UX.

In summary, OpenPC allows you, as a DG/UX user, to run most IBM PC AT applications configured for a machine as described above.

Virtual Disk

OpenPC can store DOS files inside a single DG/UX file. This DG/UX file is called a **virtual disk**. OpenPC can also store DOS files directly on the DG/UX file system using the Redirector utilities which are fully described in Chapter 6.

To OpenPC, a virtual disk appears as a DOS disk or diskette. To DG/UX, a virtual disk appears as a data file. This data file emulates C: and/or D: hard disks for OpenPC. The virtual disk is a sector-by-sector image of a PC-type hard disk stored as a single DG/UX file. This file is designated as the C: drive and a copy of the file is designated as the D: drive. You can use the configuration file `openpc.dat` (default name) to change the built-in (default) DG/UX file names for virtual C: and D: drives, as discussed in Chapter 3. These names are `/usr/opt/OpenPC/defaults/dosc.dat` and `/usr/opt/OpenPC/defaults/dosd.dat`.

Because a virtual disk is an ordinary DG/UX file, you may use DG/UX commands to `cp`, `mv`, or `rm` it from DG/UX. You must exercise caution so you don't disrupt other users who may be sharing the virtual disk with you. The virtual disks contain files that DG/UX cannot directly manipulate.

CAUTION: Use care when executing DG/UX commands that might manipulate DOS files stored on virtual disks. Examples of DG/UX commands that might manipulate the contents of a file are `sed(1)` and `awk(1)`. Be careful of running shell scripts using these commands that could act upon wildcarded files.

The DOS drive specifications A: and B: are emulated using either **virtual diskettes** or physical (hardware) diskette drives. A virtual diskette is conceptually identical to a virtual disk, except that it is much smaller because it is used to emulate A: or B: diskettes.

Virtual disks start out very small and grow as DOS files are placed on the disk. For virtual diskettes, files will be at their full size (360 KB or 1.2 MB).

OpenPC has the ability to store DOS files in two places (virtual disks and the DG/UX file system) to give you storage flexibility. You can store shared DOS programs and data on the DG/UX file system and still maintain “private” applications and data on your own copies of the virtual disks. [You would then make an entry in the OpenPC configuration file `openpc.dat` to designate drive C: as this copy of the virtual disk file (see Chapter 3). Access programs stored on the DG/UX file system using the Redirector utilities (discussed in Chapter 6).]

WARNING: Multiuser capability may violate licensing requirements of DOS applications. Some application licensing allows more than one user, but not at the same time. You can store these applications and run them from the DG/UX file system.

DOS programs and files stored on the DG/UX file system are accessed using the drive letters E: through Z:. These drives are enabled and configured using the Redirector utilities. These drives are called **redirected disks** or **redirected drives**. The difference between virtual disks and redirected disks is that DOS files are not accessible to DG/UX when stored on a virtual disk but are visible to DG/UX when stored on a redirected disk.

For example, if you have two DOS files named `file1.jnk` and `file2.jnk` and if `file1.jnk` is stored on a C: virtual disk and `file2.jnk` is stored on redirected drive E:, then an `ls(1)` from DG/UX would display `file2.jnk` but not `file1.jnk`.

A few DOS programs do not run correctly if you store them on the DG/UX file system, because they expect the program files to reside on drives A: through F:. You must then store these DOS programs on virtual disks. Harvard Total Project Manager™ II is an example.

The following table summarizes drive designations.

DOS Drive Designation	Name Used in This Manual	DOS Files Accessible to DG/UX
A:	Virtual diskette/ Physical diskette	No
B:	Virtual diskette/ Physical diskette	No
C:	Virtual disk	No
D:	Virtual disk	No
E: - Z:	Redirected disk	Yes

DOS Start-Up Files

Whenever you start up DOS, DOS first looks for a file named **CONFIG.SYS**. This file supplies necessary configuration information to the DOS operating system. In general, **CONFIG.SYS** tells DOS how to go about managing the PC's resources.

DOS also looks for a file named **AUTOEXEC.BAT**, which contains DOS commands that automatically run whenever you boot up. In general, this file tells DOS what to do when starting up. Conceptually, **AUTOEXEC.BAT** is to DOS what the file **.profile** is to the Bourne shell or **.cshrc** is to the C shell.

As with an ordinary PC running under DOS, these files are available to you after starting OpenPC. These files are stored on the C: virtual disk.

OpenPC Start-Up Files

DOS and DOS applications look for DOS device names. At the DOS prompt, you issue DOS commands. DOS commands that access devices require DOS device names. Because the names that DG/UX uses to access devices are different from the names DOS uses, there must be a way to match DOS device names to corresponding DG/UX names. (Specific default values are listed in the table on page 3-6.)

OpenPC has a set of built-in default DG/UX names. OpenPC's configuration file in DG/UX (**openpc.dat**), however, can determine how DOS device specifications correspond to DG/UX specifications. **openpc.dat** thus maps DOS device names to DG/UX device names, overriding the built-in defaults. This file also sets other operating parameters.

See **openpc.dat** keywords and settings in Chapter 3 if you want to change the correspondence between DOS devices and DG/UX. The default settings are generally appropriate for common use.

OpenPC can use a DG/UX shell environment variable named **OPENPC**, which can specify the name of the OpenPC configuration file.

You would typically set this variable within a **\$HOME/.profile** or a **\$HOME/.login** file using a DG/UX editor. To set the variable, see Chapter 3.

If the OpenPC environment variable is not present, OpenPC searches your login directory, as specified by the **HOME** environment variable, for the file **openpc.dat**.

X and Serial Terminals

OpenPC attempts to determine its environment when it starts up. If it senses it is running under X Windows, it opens a window that acts as an IBM monochrome or CGA monitor would. If OpenPC senses it is running on one of the supported terminals, it disables CGA and Hercules emulation.

The X Windows environment is preferred because

- graphics modes/emulations are supported.
- cutting and pasting are supported.
- two sets of fonts, replicating the IBM PC AT character set, are available. These fonts display characters in two different sizes, one normal and one large. (These X fonts are used only when OpenPC is in text mode. Only one font is displayed at a time. The OpenPC window is sized automatically at startup. The window is fixed at either the normal or large size and cannot be re-sized.)
- OSF/Motif graphic user interface is used although the Motif Window Manager is not required. OpenPC runs with any X Window manager.

Serial terminals do not support the features listed above. The graphic user interface is replaced by a command line interface program called `opencmd`. This program is discussed in Chapter 4.

Editing — Cutting and Pasting

OpenPC supports cut/paste operations when running under X Windows. Cutting and pasting, however, works only for text modes. You may cut/paste from/to OpenPC windows only when a text-mode video emulation is in effect.

The mouse is used to cut and paste. See Chapter 4 for an explanation of mouse operation.

Idle Detection

If OpenPC detects no user input from the keyboard or mouse for 45 seconds, OpenPC “goes to sleep.” OpenPC suspends but does not kill the session, to give other DG/UX processes a larger share of CPU time when DOS is polling for user input (idling).

You can turn idle detection on or off using the **Session** menu (under X Windows only) or the `idledetection` function of the `openpcmd` command described in Chapter 4. You typically disable idle detection whenever you need to keep OpenPC running without keyboard or mouse input, as in lengthy batch file operations or spreadsheet recalculations.

If idle detection is enabled and OpenPC is sleeping, keyboard input will be delayed up to one second until OpenPC wakes up.

CMOS RAM and ROM BIOS

An IBM PC AT stores its configuration information in battery-backed non-volatile CMOS RAM. This RAM stores information such as types of disk drives, primary video adapter, and current date and time. OpenPC emulates the CMOS RAM by using a DG/UX file `/usr/opt/OpenPC/defaults/cmos.dat`. (Alternatively you can specify a different DG/UX CMOS RAM file name. To do so, you must change the configuration file `openpc.dat` in DG/UX. See Chapter 3.)

Similarly, an IBM PC AT stores its bootup code and Basic Input Output System (BIOS) in ROM chips. OpenPC emulates this ROM BIOS by storing the same code in a DG/UX file `/usr/opt/OpenPC/defaults/rom.dat` which is the built-in default value. (Alternatively you can specify a different DG/UX ROM BIOS file name. To do so, you must change the configuration file `openpc.dat` in DG/UX. See Chapter 3.)

WARNING: Do not change or put in the ROM entry in `openpc.dat` unless told to do so by your system administrator.

Keyboard Mapping

Since OpenPC runs under X Windows, it may be run across the network using any X server available on the network. This situation can present problems with keyboard input because the keyboards and/or the codes sent by the X server may be different from server to server, even when the keyboards appear to be the same physically.

OpenPC uses a keyboard mapping file to establish which server-keyboard combination it is getting its input from. The `openpc.dat` file can establish the mapping through the keyword `KEYBOARD_TRANSLATIONS`. The built-in default mapping is for an AViiON 300 workstation. The mapping file is `/usr/opt/OpenPC/term/aviion.key`. Keyboard mapping files for the AViiON 300 Workstation, AVX-30 X display and other popular X Windows workstations are provided in the directory `/usr/opt/OpenPC/term`.

You can change the keyboard mapping file after OpenPC is started from the **Load Keyboard** selection of the **Customize** menu. These menus are described in Chapter 4.

The Keyboard mapping is not used when OpenPC is running on a serial terminal.

Installing OpenPC

This chapter tells **system administrators** how to install and remove OpenPC software on the AViiON system. See Chapter 4 if you simply want to run OpenPC.

Installation is a two-step process consisting of

1. Loading the software.
2. Setting up the users.

Customizing the software for individual users is discussed in Chapter 3.

Installing the OpenPC Software

OpenPC must be loaded onto disk from the OpenPC tape. You can do this for your system after you have inserted the tape cartridge by following the steps listed below. **Please refer to your OpenPC release notice for detailed installation information.** This is both on tape and available as hard copy.

1. Log on as root.
2. Ensure that you are at run level 3. Use the DG/UX command **who -r** to determine the run level. Type **init 3** if you are not at this level. Mount the OpenPC file system under **/usr/opt**. Refer to your release notes for details on creating the OpenPC file system.
3. Type **sysadm**. This brings up the **sysadm** menu.
4. Choose **releasemgmt** from the menu. This brings up the **releasemgmt** menu.

5. Choose **loadpackage**. This will lead to a series of prompts.
6. Select the defaults which are appropriate for your system. Refer to the release notice for the step-by-step procedures. This procedure loads the tape onto the disk for your system.
7. Exit from **sysadm**.

If you are installing OpenPC on a diskless workstation, see the release notice.

Checking Installation

After loading the OpenPC software package as outlined previously, the system administrator must

- configure OpenPC to enable use by end users.
- install DOS applications end users want to run, or let end users install their own DOS applications.

Be sure to read Chapters 1 and 4 to help you with configuring and installing. As loaded from the tape, OpenPC is ready to run, but Chapters 1 and 4 explain how to use the software properly.

On a bitmapped graphics display, such as the AViiON 300 workstation or the AVX 30 display terminal, you can run OpenPC only if X Windows is running. If X Windows is not running, make sure `/usr/bin/X11` is in your path and then type **xstart**.

Check to see you have correctly installed OpenPC by starting it as described below.

For the Bourne shell, enter:

```
OPENPC=/usr/opt/OpenPC/openpc.dat; export OPENPC
```

For the C shell, enter:

```
setenv OPENPC /usr/opt/OpenPC/openpc.dat
```


If you see a window appear on your display similar to the one below, you are ready to install DOS applications.

```

Session  Edit  Customize  Print
-----
Phoenix 80286 ROM BIOS Version 3.10 V
Copyright (c) 1985-1988 Phoenix Technologies Ltd.
All Rights Reserved

C>prompt = $p$g

C:\>path c:\sys;c:\

C:\>redir
Redirector Installed: Version 1.14

C:\>dosmount h:-$HOME
Hostname is 'Virtual_PC'
Redirected drives:
    Drive H:\USR\PC\ROMEL
For help on DOSMOUNT, type DOSMOUNT HELP

C:\>
E:\>
C:\>_

```

If you do not see this window or if OpenPC exits after displaying the message

```
Cannot open /usr/opt/OpenPC/term/xterm.
TERM set correctly?
```

then the environment variable **DISPLAY** is not set. You must set **DISPLAY** or enter

```
openpc -display hostname:0
```

See the X Window manual if you do not know how to set **DISPLAY** or how to run X Window System clients. (OpenPC is an X Window client.)

See the X Window manual if you do not know how to set **DISPLAY** or how to run X Window System clients. (OpenPC is an X Window client.)

On a supported serial terminal, check installation by starting OpenPC. For the Bourne shell, enter:

```
OPENPC=/usr/opt/OpenPC/openpc.dat; export OPENPC
TERM=dg_pcterm; export TERM
```

For the C shell, enter:

```
setenv OPENPC /usr/opt/OpenPC/openpc.dat
setenv TERM dg_pcterm
```

Make sure **/usr/opt/OpenPC** is in your path as described at the top of page 2-3. Then enter:

```
openpc
```

You should see your terminal display the same messages as in the illustration on the previous page. You should also see the DOS prompt appear on your terminal screen.

Installing DOS Applications

Once you have OpenPC running, you may install DOS applications so that they will be available to the user.

It is recommended that you create a file system called “dosbin” to be used for DOS programs. Mount the “dosbin” directory in the DG/UX directory **/usr/opt**. Refer to your DG/UX documentation for more information on mounting file systems using diskman.

NOTE: Each subdirectory name should be less than 8 letters, should be lowercase and have only alpha-numeric characters. For example, Lotus 1-2-3 would be in **/usr/opt/dosbin/lotus123** but not in **/usr/opt/dosbin/LOTUS.123.V1**. You must follow the rules to prevent name translation from occurring when the subdirectory is accessed from DOS. See file name mapping in Chapter 4.

Initial User Set-Up

After checking the installation of OpenPC as outlined previously, the system administrator needs to set up individual users as follows:

1. Start OpenPC as described in the previous sections. Make sure you have the DOS C: prompt.
2. Decide on a directory location where DOS applications will reside. Load your desired applications in that directory using OpenPC. Recommended is

`/usr/opt/dosbin`

where `dosbin` is a mounted file system under `/usr/opt`.

If you have followed the recommendations on DOS application storage, then proceed to step 4.

3. Modify the `AUTOEXEC.BAT` file, using a DOS editor such as Edlin, in the default C: virtual disk so that you have access to your desired DOS applications. You must change the default `DOSMOUNT` command in the `AUTOEXEC.BAT` file to the subdirectory name you have chosen in step 2 above. If you are not familiar with Edlin, you can follow the DOS procedure listed below to change the `AUTOEXEC.BAT` file.
 - a. At the C: prompt, type `copy con temp`
 - b. Type `dosmount e:/usr/opt/dosbin`
 - c. While holding down the Control key, type `z`
 - d. Type `copy autoexec.bat + temp`
 - e. Type `type autoexec.bat` and then look for the new `DOSMOUNT` subdirectory command in the file to make sure it was entered.
 - f. Type `autoexec` to set the changes in DOS.

See Chapter 6 on the `DOSMOUNT` supporting utility for syntax if necessary.

4. Set up individual users by copying the default `openpc.dat` file and the default C: and D: virtual disks to the user's home directory. This procedure allows individual users to boot OpenPC from their own C: and D: virtual disks. It also allows individual users to store data files on their own virtual disk.

The default `openpc.dat` file is located in `/usr/opt/OpenPC/`. The default C: and D: virtual disks have the names:

```
/usr/opt/OpenPC/defaults/dosc.dat  
/usr/opt/OpenPC/defaults/dosd.dat
```

For more information on system administration when installing DOS applications, see Chapter 5.

Removing OpenPC

Follow this procedure if you want to take OpenPC off your AViiON hard disk. After you have completed the steps below, you will have to re-install OpenPC the next time you want to use it.

1. Remove OpenPC by using the DG/UX command

```
rm -r /usr/opt/OpenPC
```

2. Remove the installed DOS programs using the DG/UX command

```
rm -r /usr/opt/dosbin
```

Customizing OpenPC

This chapter provides configuration information for users who want to specify their own configuration parameters. These parameters are contained in several DG/UX and DOS configuration files.

Discussed in this chapter are OpenPC's configuration file **openpc.dat** (a DG/UX file), the DG/UX shell environment variable **OPENPC**, the DOS configuration file **CONFIG.SYS**, and the DOS command file **AUTOEXEC.BAT**.

This chapter also tells you how to configure the LIM EMS (Lotus/Intel/Microsoft Expanded Memory Specification) 4.0 memory driver.

The `openpc.dat` File

The `openpc.dat` file maps virtual DOS device names to DG/UX files and devices such as lineprinter and tty. `openpc.dat` also contains configuration options.

Data General supplies a master default `openpc.dat` file which establishes the command configuration for all OpenPC users. It is automatically loaded into the DG/UX directory `/usr/opt/OpenPC` when the OpenPC software package is installed. OpenPC, however, looks for a file named `openpc.dat` in your home directory before using the default file. You can copy `/usr/opt/OpenPC/openpc.dat` to your home directory where you can edit the copied file. Use the DG/UX `cp` command to copy both this default file `openpc.dat` and the C: virtual disk to your home directory if your system administrator has not done so already.

OpenPC will run without `openpc.dat` being present in your home directory. The built-in `openpc.dat` default values may not be the ones you want, however, especially because some C: and D: virtual disk files will be used by everyone running OpenPC. You can use a DG/UX editor such as `vi(1)` to edit your copy of `openpc.dat`. Edit so that, at the minimum, C: and D: virtual disk files are in your home directory.

Following this procedure enables you to keep private data separate from shared DOS applications and prevents device allocation conflicts. A private `openpc.dat` file for each OpenPC user is highly recommended.

Writing your own `openpc.dat` file

The format of each line of the `openpc.dat` file consists of entries containing the OpenPC configuration keywords and their desired settings, one entry per line. You can include comments, or turn off unwanted devices, by inserting any character at the beginning of the line. If OpenPC does not understand the first character in a line, the entire line is ignored.

On the following page is a sample `openpc.dat` configuration file. Yours should be similar.

Immediately following the sample file is a page of explanation. Each section in the sample file is matched by a numbered explanation.

For a complete listing of default values and further explanations, see the table of `openpc.dat` keywords and settings beginning on page 3–6.

```

# openpc.dat - OpenPC configuration file
# Format:
# -----
# Keyword                DG/UX device or file
#
# A "#" in column 1 denotes comment
#
# A: is virtual disk or physical disk. Un-comment to choose.
# DISKETTE_A              /usr/opt/OpenPC/disks/dosa.dat
DISKETTE_A                ① /dev/rpdsk/3
DISKETTE_B                /usr/opt/OpenPC/disks/dosb.dat
FIRST_HARD_DISK          /usr/opt/OpenPC/disks/dosc.dat
#FIRST_HARD_DISK        ② /usr/opt/OpenPC/disks/pdos401.dsk
SECOND_HARD_DISK        /usr/opt/OpenPC/disks/dosd.dat
CMOS_FILE                ③ /usr/opt/OpenPC/cmos.dat
ROM                       ④ /usr/opt/OpenPC/rom.dat
BOOT_DEVICE              ⑤ C
#
# Normal size PC fonts (Set 1)
N_FONT                   pc_font
U_FONT                   pc_fontu
B_FONT                   ⑥ pc_fontb
UB_FONT                  pc_fontub
40_FONT                  pc_font40
#
# Wider size PC fonts (Set 2)
# N_FONT                 pc_wfont
# U_FONT                 pc_wfontu
# B_FONT                 ⑥ pc_wfontb
# UB_FONT                pc_wfontub
# 40_FONT                pc_wfont40
#
USE_COLOR                 ⑦ FALSE
# DOS COM1 is mapped to asynch port
COM_PORT_1                ⑧ /dev/tty01
# Next line is for running on AViiON X Server
KEYBOARD_TRANSLATIONS    /usr/opt/OpenPC/term/aviion.key
# Next line is for running on AVX-30 X terminal
# KEYBOARD_TRANSLATIONS  ⑨ /usr/opt/OpenPC/term/avx30.key
# Next line is for running on MIPS® X Server
# KEYBOARD_TRANSLATIONS  /usr/opt/OpenPC/term/mips.key
# Next line is for running on VAX® X Server
# KEYBOARD_TRANSLATIONS  /usr/opt/OpenPC/term/lk301.key

```

-
- ① These entries in the configuration file map DOS disk drive A: to the AViiON diskette device and assign a virtual diskette to DOS disk drive B: . Thus the B: virtual device is mapped to a DG/UX file that emulates a diskette drive. The DG/UX files should have Read/Write permissions.
 - ② These entries in the configuration file are for the virtual disk drives. They configure the C: virtual disk to the DG/UX file `/usr/opt/OpenPC/disks/dosc.dat` and D: to be `/usr/opt/OpenPC/disks/dosd.dat`. Note that one entry is commented out. This illustrates how different virtual disk files may be configured to the C: or D: DOS drives. The DG/UX files should have Read/Write permissions.
 - ③ This entry defines the OpenPC CMOS image file. It defines the DG/UX specification for the PC's CMOS RAM file. This file should have Read/Write permission for the owner.
 - ④ This line specifies a file containing the ROM BIOS image. This entry defines a file image of the standard OpenPC ROM BIOS. It is a read-only file.

WARNING: Do not change this entry in your `openpc.dat` file unless told to do so by your system administrator. It will be changed only if your site receives a new `rom.dat` file from Data General.

- ⑤ This entry defines which drive to boot from. However, if the boot device is the A: drive, and the A: drive is not available or does not exist, then OpenPC automatically defaults to the C: drive (just as a PC would).
- ⑥ These lines configure the normal and wide (w) fonts provided with OpenPC. These fonts are used only in the OpenPC window to allow it to run under X11 with the full IBM PC AT character set. These fonts are not included with X11.

The wide font set has been commented out. OpenPC provides fonts of two widths. You should comment out one entire set or the other and not mix the two. These are X11 fonts and may be displayed without invoking OpenPC by using the `xfd` program.

Note that no path name for the font is specified since the fonts have to be in `/usr/lib/X11/fonts/misc` for the X server to access them.

These fonts are supplied with OpenPC and are installed in the standard place as part of the `sysadm(1m)` installation procedure.

- ⑦ This line allows the user to specify, on a color display, whether to display graphics in color or in monochrome. In the default configuration, it is turned off (set to FALSE).
- ⑧ This line specifies the DG/UX device file that is used to emulate the virtual AT's COM port 1.

Note: You should not have a `getty(1m)` running on `tty01`.

- ⑨ These lines specify keyboard mapping files. Four keyboard translation file specifications are here, and only AViiON 300 is not commented out.
-

openpc.dat Keywords and Settings

The following table is a listing of all keywords and some possible settings that you use in the **openpc.dat** file. These keywords and settings are optional, used according to your own configuration needs.

Keyword	Default Value	Example Value
DISKETTE_A	NONE	<code>/usr/opt/OpenPC/disks/dosa.dat</code> <code>\$HOME/dosa.disk</code> <code>/dev/rpdk/3</code>
In DOS, drive A:.. Maps A: to either a DG/UX file representing the virtual diskette or DG/UX physical diskette device. <code>\$HOME/dosa.disk</code> would be a copy of the supplied A: virtual diskette file. <code>/dev/rpdk/3</code> would be the physical diskette.		
DISKETTE_B	NONE	<code>/usr/opt/OpenPC/disks/dosb.dat</code> <code>\$HOME/dosb.disk</code> <code>/dev/rpdk/3</code>
In DOS, drive B:.. Maps B: to either a DG/UX file representing the virtual diskette or DG/UX physical diskette device. <code>\$HOME/dosb.disk</code> would be a copy of the supplied B: virtual diskette file. <code>/dev/rpdk/3</code> would be the physical diskette.		
FIRST_HARD_DISK	<code>/usr/opt/OpenPC/defaults/dosc.dat</code>	<code>\$HOME/my_C.dat</code> <code>/usr/opt/OpenPC/disks/test</code>
In DOS, drive C:.. Maps C: to a DG/UX file representing the virtual disk. <code>\$HOME/my_C.dat</code> would be a copy of the default file <code>/usr/opt/OpenPC/defaults/dosc.dat</code> . You would now store DOS programs and data in <code>\$HOME/my_C.dat</code> .		
SECOND_HARD_DISK	<code>/usr/opt/OpenPC/defaults/dosd.dat</code>	<code>\$HOME/my_D.dat</code> <code>/usr/opt/OpenPC/disks/test</code>
In DOS, drive D:.. Maps D: to a DG/UX file representing the virtual disk. <code>\$HOME/my_D.dat</code> would be a copy of the default file <code>/usr/opt/OpenPC/defaults/dosd.dat</code> . You would now store DOS programs and data in <code>\$HOME/my_D.dat</code> . Also, C: and D: virtual disk files can be copies of each other, or even the same DG/UX file.		

Keyword	Default Value	Example Value
USE_COLOR	FALSE	TRUE
<p>Allows you to specify, on a color display, whether to display graphics in color (TRUE) or in shades of gray (FALSE). Setting to FALSE on a color system enhances speed of visual updating. It has no effect on a monochrome system. Only the values TRUE or FALSE are allowed for the USE_COLOR keyword. Anything else sets the value to FALSE.</p>		
CMOS_FILE	/usr/opt/OpenPC/ defaults/cmos.dat	\$HOME/my_cmos.dat
<p>Defines DG/UX file representing OpenPC's battery-backed RAM. \$HOME my_cmos.dat would be your private copy of /usr/opt/OpenPC/defaults/cmos.dat. Private copy is necessary because the file is updated when you exit OpenPC, so if two users exit at the same time the file could be corrupted. Two users could also have different memory configurations (expanded and not expanded). You should, therefore, have your own cmos.dat in your \$HOME directory, just as you have virtual disks.</p>		
ROM	/usr/opt/OpenPC/ defaults/rom.dat	NONE
<p>Contains the ROM BIOS image for OpenPC. There is usually no need to change the default value of this keyword.</p>		
BOOT_DEVICE	C	A
<p>Specifies the drive to boot from at startup. If the boot device is the A: drive, and the A: drive does not exist or is not available, or the entry is not present, then OpenPC automatically defaults to the C: drive (just as a PC would). Only the values C or A are allowed for this keyword.</p>		
N_FONT	pc_font	pc_wfont fixed
<p>Specifies an X11 font used as OpenPC normal font. pc_wfont specifies a wide font. Specify fixed if the particular X server you want OpenPC to connect to does not have the OpenPC font sets installed. (Use the xlsfonts -fn "pc*" command to query the X server.) See also the sample openpc.dat file explanation on page 3-5.</p>		

Keyword	Default Value	Example Value
U_FONT	pc_fontu	pc_wfontu fixed
Specifies an X11 underline font. Specify fixed if the particular X server you want OpenPC to connect to does not have the OpenPC font sets installed. (Use the <code>xlsfonts -fn "pc*"</code> command to query the X server.) See also the sample <code>openpc.dat</code> file explanation on page 3-5.		
B_FONT	pc_fontb	pc_wfontb fixed
Specifies an X11 bold font. Specify fixed if the particular X server you want OpenPC to connect to does not have the OpenPC font sets installed. (Use the <code>xlsfonts -fn "pc*"</code> command to query the X server.) See also page 3-5.		
UB_FONT	pc_fontub	pc_wfontub fixed
Specifies an X11 underline bold font. Specify fixed if the particular X server you want OpenPC to connect to does not have the OpenPC font sets installed. (Use the <code>xlsfonts -fn "pc*"</code> command to query the X server.) See also page 3-5.		
40_FONT	pc_font40	pc_wfont40 fixed
Specifies an X11 40-column font. Specify fixed if the particular X server you want OpenPC to connect to does not have the OpenPC font sets installed. (Use the <code>xlsfonts -fn "pc*"</code> command to query the X server.) See also page 3-5.		
KEYBOARD_TRANSLATIONS	/usr/opt/OpenPC/ term/aviion.key	/usr/opt/OpenPC/term/ avx30.key
Specifies the keyboard translation file for OpenPC to use. If you run OpenPC under X Windows, a new keyboard translation file may be loaded using the Load Keyboard option from the Customize menu. The example value sets the keyboard translation file for the AVX-30 X display station.		

Keyword	Default Value	Example Value
LINE_PRINTER_1	NONE	<pre>pr lp lp -dlj1 cat >filename cat >/dev/lp</pre>
<p>Specifies which DG/UX printing facility is to be used for OpenPC printing. The examples show four possible entries:</p> <p>pr lp — Specifies OpenPC printing to LPT1 be sent to DG/UX's pr utility which pipes to the line printer. See the pr(1) and lp(1) man pages.</p> <p>lp -dlj1 — Specifies OpenPC printing to LPT1 be sent to the line printer whose destination is specified to be printer lj1. See the lp(1m) man pages.</p> <p>cat >filename — Specifies OpenPC printing to LPT1 be sent to a DG/UX file via the cat(1) utility. See the cat(1) man page.</p> <p>cat >/dev/lp — Specifies OpenPC printing to LPT1 be sent directly to the line printer using cat(1).</p> <p>NOTE: Only one of the above examples should be entered in the openpc.dat file.</p>		
LINE_PRINTER_2	NONE	Same as for LINE_PRINTER_1
Same as for LINE_PRINTER_1. Only DOS printer name changes from LPT1 to LPT2.		
LINE_PRINTER_3	NONE	Same as for LINE_PRINTER_1
Same as for LINE_PRINTER_1. Only DOS printer name changes from LPT1 to LPT3.		
COM_PORT_1	NONE	/dev/tty01
<p>COM1 is the DOS name for the first serial port, usually connected to a modem. This entry in openpc.dat specifies that the DG/UX serial port tty01 is assigned to COM1.</p> <p>Find out which DG/UX TTY port to use from your system administrator, or else you may conflict with login users. /dev/tty01 is only an example.</p> <p>NOTE: You should not have a getty(1m) running on tty01 if this entry is in openpc.dat.</p>		
COM_PORT_2	NONE	/dev/tty02
<p>COM2 is the DOS name for the second serial port, usually connected to a modem. This entry in openpc.dat specifies that the DG/UX serial port tty02 is assigned to COM2.</p> <p>Find out which DG/UX TTY port to use from your system administrator, or else you may conflict with login users. /dev/tty02 is only an example.</p> <p>NOTE: You should not have a getty(1m) running on tty02 if this entry is in openpc.dat.</p>		

Customizing with the OPENPC Variable

To customize your environment, you can use the **OPENPC** variable. It specifies the file name for the OpenPC configuration file. This variable lets you use different configuration files. **OPENPC** is not the same as the executable **openpc**.

Example OPENPC

OpenPC first looks at the value of the **OPENPC** shell environment variable. For example, for the Bourne shell (`/bin/sh`), after the `$` prompt, enter

```
OPENPC=$HOME/myconfig
export OPENPC
```

For the C shell (`/bin/csh`), after the `%` prompt, enter

```
setenv OPENPC ~/myconfig
```

When you enter these environment variables at the DG/UX shell prompt, you establish the file **myconfig** in the path pointed to by the **\$HOME** expansion string as the configuration file OpenPC is to use upon startup. If **OPENPC** is not in the environment, then OpenPC looks for **\$HOME/openpc.dat**. If **\$HOME/openpc.dat** is missing, OpenPC uses the built-in defaults listed in the table in the preceding section.

Note that **OPENPC** establishes a configuration file name (including its path) for OpenPC. Otherwise, the configuration file is established as **\$HOME/openpc.dat**.

DOS Start-Up Files

When DOS starts up (boots), it first looks for a file named **CONFIG.SYS**. This file supplies configuration information to the DOS operating system. In general, this file tells DOS how to go about managing the machine's resources.

Below is the default **CONFIG.SYS** file.

```
buffers=20
files=20
fcbs=25,25
stacks=0,0
lastdrive=z
```

DOS also looks for a file named **AUTOEXEC.BAT**, which contains DOS commands that automatically run whenever you boot. In general, this file tells DOS what to do when starting up.

Below is the default **AUTOEXEC.BAT** file.

```
prompt = $p$g
path c:\sys;c:\
redir -l
dosmount h:=$HOME
dosmount e:/usr/opt/dosbin
```

These files must reside on virtual disks. **AUTOEXEC.BAT** is searched for on the disk OpenPC boots from (C: or A:) and must be in the disk's root directory.

As with an ordinary PC running DOS, these files are available to you after starting OpenPC. (Refer to your DOS user's manual for more information on DOS.) These files can be edited as needed using customary DOS procedures.

Using the LIM EMS 4.0 Driver

You will need the LIM EMS 4.0 (Lotus/Intel/Microsoft Expanded Memory Specification) memory driver only when applications require it or when you get an “out of memory” error message from DOS or a DOS application.

The LIM EMS driver supports up to 32 MB of expanded memory. The Phoenix EMS driver, PEMM.EXE, is a software interface between OpenPC and DOS programs such as Lotus 1–2–3. You must enter `pemm.exe` in the `CONFIG.SYS` DOS file to use the LIM EMS driver. Use a DOS editor such as Edlin or other DOS procedures to make the following entry in the `CONFIG.SYS` file:

```
device=pemm.exe [Sx]
```

where `Sx` is the size of expanded memory allocated on startup. The `x` is the number of pages, each containing 16 KB.

For example, to configure an 8 MB space, insert:

```
device=pemm.exe S512
```

into your `CONFIG.SYS` file. (512 x 16 KB = 8 MB.)

Specifying the number of 16 KB pages is optional. The default is 64, which gives 1 MB of expanded memory (64 x 16 KB = 1 MB).

OpenPC supports a maximum of 2048 of these 16 KB pages.

EMS driver installation increases OpenPC’s memory demand on DG/UX. Install only when necessary. Expand memory only by increments of half a megabyte until your DOS application runs properly.

Remember to delete the `device=pemm.exe` line from the `CONFIG.SYS` file as soon as you no longer need the expanded memory.

Using OpenPC

This chapter describes

- starting OpenPC
- the OpenPC menu bar
- using `opencmd`
- access to DG/UX files from DOS, including
 - virtual disk access
 - file name mapping
- printing from OpenPC
- using the mouse

Starting OpenPC

If you are at an AViiON workstation console, and if X11 is not already running, start X11 by

1. Making `/usr/bin/X11` part of your path.
2. Entering `xstart&`

OpenPC can run with any X11 window manager. Its user interface, however, follows OSF/Motif graphic conventions. Therefore, running OSF Motif Window Manager (MWM) before you run OpenPC provides the most consistent user interface.

You usually start the Motif Window Manager by entering `mwm&` at the shell prompt. See the X11 documentation if you are unsure how to start and use the X Window System or the Motif Window Manager.

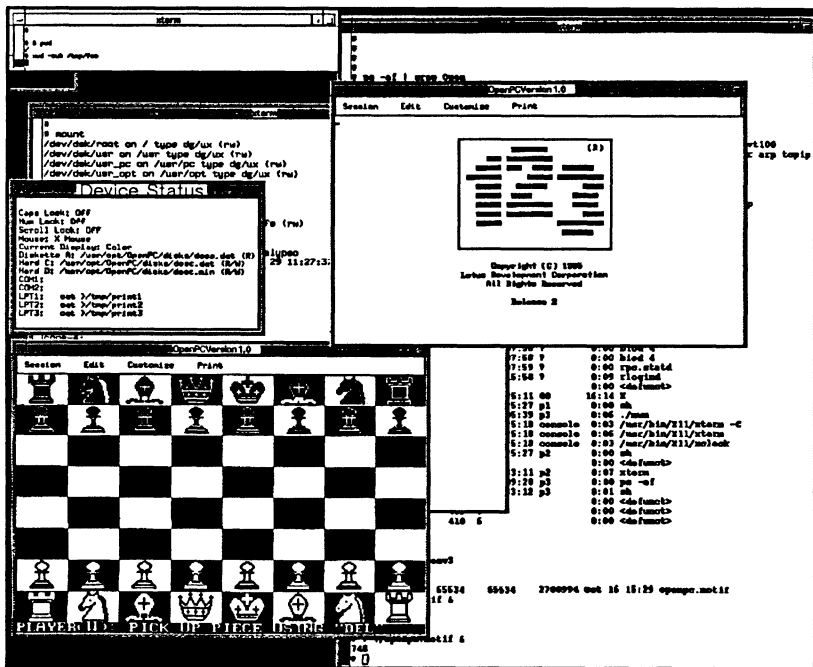
To begin running OpenPC, type

```
openpc
```

at the DG/UX shell prompt. You can also run OpenPC as a background process, by typing `openpc&` instead. Make sure the directory `/usr/opt/OpenPC` is in your path first. See page 2-3.

If you do not start OpenPC from an X terminal, it will attempt to run in serial terminal mode, using the `TERM` environment variable as the name of the asynchronous serial terminal it is to run on.

See the screen below for a picture of two OpenPC sessions running on an AViiON workstation under X11 (Lotus 1-2-3 and a chess game). The window manager you see running is Motif Window Manager.



Search Precedence

Upon invocation, OpenPC searches for its configuration file in the sequence below. The search stops as soon as the file is found.

1. File specified by the **OPENPC** environment variable.
2. File **openpc.dat** in the path specified by the **HOME** environment variable. This should be your home directory.
3. File **/usr/opt/OpenPC/openpc.dat**.

If none of these files is found, OpenPC continues using built-in default values as outlined in Chapter 3. Chapter 3 tells you how to modify default values in the **openpc.dat** file.

The OpenPC Menu Bar

The OpenPC window displaying the menu bar and DOS boot information in the work area is shown below.

Session	Edit	Customize	Print
Phoenix 80286 ROM BIOS Version 3.10 V			
Copyright (c) 1985-1988 Phoenix Technologies Ltd.			
All Rights Reserved			
C>prompt = \$p\$g			
C:\>path c:\sys;c:\			
C:\>redir			
Redirector Installed: Version 1.14			
C:\>dosmount h:=\$HOME			
Hostname is 'Virtual_PC'			
Redirected drives:			
Drive H:\USR\PC\ROMEL			
For help on DOSMOUNT, type DOSMOUNT HELP			
C:\>			
C:\>			
C:\>_			

NOTE: The OpenPC window is fixed at either normal or large size, depending on your choice of normal or large font width. You cannot create other sizes.

You can open a menu of options from each word in the menu bar. To do so, click **[MB1]** after positioning the mouse cursor on the word. To select an option, click **[MB1]** while pointing to the option.

Some of the options have graphic “toggle buttons.” The toggle button is a shaded square inside a box (see Pause in the menu below for an example). The shaded square is an indicator showing when the option is on. When off, the option has no indicator inside the box.

Some of the options incorporate submenus. This means that options with arrows at the right of the option name have submenus of their own. To bring up the submenu, select the option, then select the suboption desired.

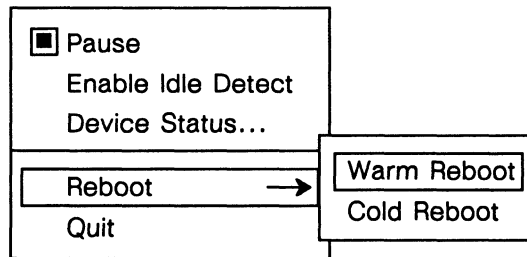
Other options pop up dialog boxes which request your input such as a file name or acknowledgment.

Below the menu bar is the work area, which behaves in the same way as a screen on a standalone PC.

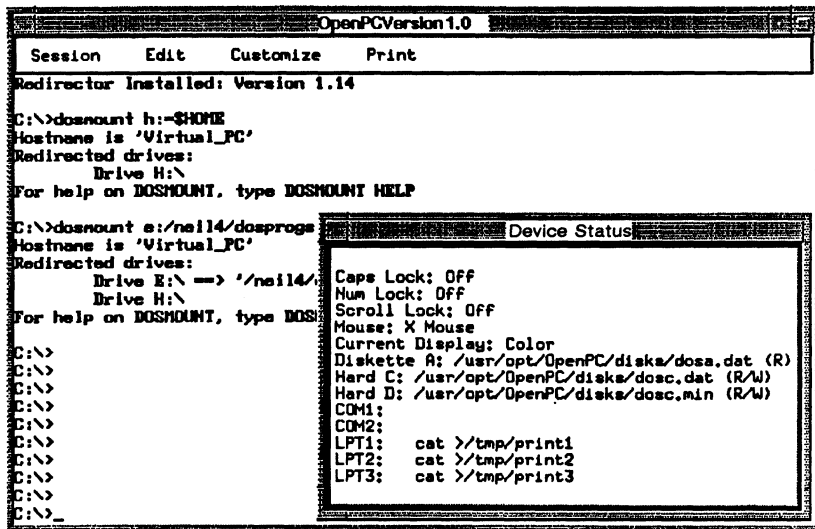
Menu Bar Selections

The following describes the selections available from the menu bar of the OpenPC window.

Choosing **Session** displays the following menu. Selecting **Reboot** brings up the submenu as shown:



The Device status... option opens an information popup box which requires no response.

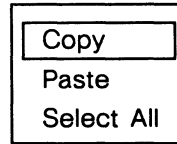


The following table describes the Session menu selections.

Selection	Description
Pause	suspends or restarts the execution of OpenPC. You can use the switch to toggle between pause and restart. Pause is on if the indicator is filled in, off if the button is not present.

Selection	Description
Enable Idle Detect	<p>If DOS receives no user input from the keyboard or mouse (remains idle) for 45 seconds, OpenPC sleeps. OpenPC operation is suspended so that other DG/UX processes may have full attention of the AViiON processor.</p> <p>You can switch off this feature using the toggle button. When on, the indicator is filled in; when off, the indicator is not present. The default state is on.</p> <p>Use this selection for long batch processing.</p>
Device Status ...	<p>displays a message window listing the status of all devices. See the picture of the popup window preceding this Session menu.</p>
Reboot →	<p>Cold Reboot — restarts the OpenPC program. It is equivalent to turning a PC off and then back on.</p> <p>Warm Reboot — resets the OpenPC program. The <code>openpc.dat</code> file is not read, and any reassignments of disk drives will remain. Equivalent to pressing Control–Alt–Delete keys.</p>
Quit	<p>asks Do you really want to quit? Selecting Yes closes the OpenPC window. Selecting No returns you to the OpenPC window.</p> <p>NOTE: DOS applications should be exited normally before quitting or files may be corrupted.</p>

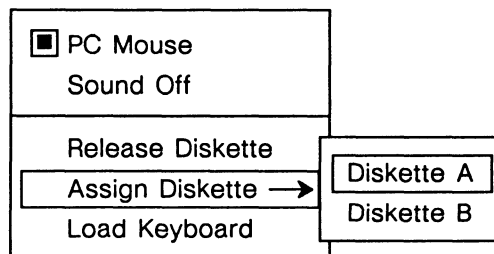
Choosing **Edit** displays the following menu:



Following is a table describing all the **Edit** menu selections, to be used only in text mode (no graphics). You can also cut/paste simply by using mouse buttons without using this menu at all, following X Window usage. **MB1** cuts, **MB3** pastes.

Selection	Description
Copy	places selected text into a clipboard buffer. Select text by pressing MB1 while dragging mouse cursor to highlight the text. The clipboard will hold this text until you Copy again.
Paste	places contents of the clipboard into the currently active OpenPC window.
Select All	selects the entire currently displayed screen (text) image. You can Copy this image into the clipboard, and Paste it from the clipboard.

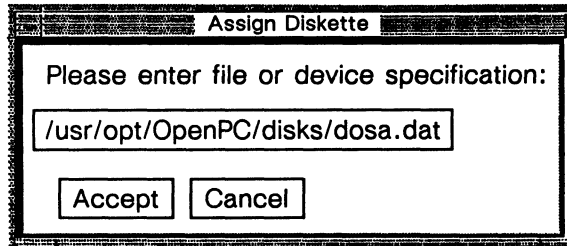
Choosing **Customize** displays the following menu and submenus:



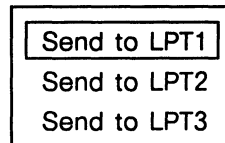
Following is a table describing the selections in the **Customize** menu.

Selection	Description
PC Mouse	enables a PC Mouse configuration. Toggles between X Mouse and PC Mouse. PC Mouse is on if the indicator is filled in, X Mouse is on if the indicator is not present.
Sound Off	toggles Sound Off/On using the toggle switch. Sound is off if the indicator is filled in, sound is on if the indicator is not present.
Release Diskette →	<p>allows you to release a diskette you have used, so that others may use it. Simply select the device you have finished with, and click the [MB1] button.</p> <p>NOTE: When this OpenPC session terminates, any devices attached to it are automatically released.</p>
Assign Diskette →	allows you to change the device specification for Diskette A or Diskette B. Select the device you wish to see the specification for, and its current status is shown. A dialog box appears which allows you to enter a new file specification to assign a virtual diskette or physical drive. See the drawing of this window following. Position cursor (now in the form of an “I beam”) to edit in the box.
Load Keyboard	allows you to change the keyboard translation file without exiting OpenPC, editing openpc.dat , and re-starting OpenPC. A dialog box appears which allows you to enter a new file specification. Position cursor (now in the form of an “I beam”) to edit in the box.

You can see below the popup window with its dialog box where you can edit or enter a file specification to assign a virtual diskette or physical drive.



Choosing **Print** displays the following menu:



Following is a table describing the selections in the **Print** menu.

Selection	Description
Send to LPT1	forces anything in the printer buffer to the LPT1 printer.
Send to LPT2	forces anything in the printer buffer to the LPT2 printer.
Send to LPT3	forces anything in the printer buffer to the LPT3 printer.

The printer buffer is automatically flushed as it fills up. You may, however, request only a small amount of print activity that does not fill the buffer. If the buffer is not filled, it will not be forced to flush automatically. To flush when it is not filled, you must use the Print menu.

You must flush the print buffer before you quit to prevent your print job from becoming interspersed with print jobs other users have queued up.

It is also a good idea to flush the printer before ending an OpenPC session if you have done any printing during the session. For more information on printing, see the section on printing at the end of this chapter.

Chapter 3 tells you how to associate DOS LPT devices to DG/UX printing facilities.

Using `opencmd` Instead of Menus

The `opencmd` utility is a DOS program that allows you to do many of the same functions as the menus. This includes rebooting, assigning and releasing diskettes, etc.

If you are running OpenPC on a serial terminal and, therefore, lack a windowing system, the `opencmd` utility gives you the functionality of the OpenPC menus. It may also be used instead of the OpenPC menus when OpenPC is running under X Windows.

This utility also can be useful for interacting with OpenPC via DOS batch files.

`opencmd` resides on the default C: virtual disk. Its syntax is

```
opencmd function [parameters]
```

The following table contains a brief summary of **opencmd** functions. Functions are listed in the same order as they appear in the **opencmd** command summary. You can see this summary in your DOS window after you type **opencmd** at the DOS prompt.

Function	Allowed Parameters	Description
assigndiskette	a, b; <disk name>	assigns drive or virtual diskette
devstatus	(NONE)	displays status of PC devices
flushprinter	1, 2, 3	flushes PC printer buffer
idledetection	enabled, disabled	controls PC idle detection
quit	(NONE)	exits PC environment
reboot	warm, cold	reboots PC
releasediskette	a, b	releases drive or virtual diskette
sound	on, off	controls sound (beeping)
mouse	PC, host	controls mouse mode (this has no effect on a serial terminal, but only on a terminal running X Windows)

Commands and parameters do not have to be spelled out fully. You only need to type enough characters to make the command or parameter unambiguous. For example, typing **opencmd s of** is sufficient to turn the sound off.

Note the following menu functions are not available through **opencmd**:

- Pause
- Edit (copy, paste, select all)

Example Usage of `opencmd` in a DOS Batch File

You would use `opencmd`, for example, when you have a batch file which automatically emulates the insertion of a physical diskette before running a program. Running Lotus 123 with a financial data file might be implemented with the following batch file:

```
opencmd assign a financial.dat
opencmd idle disable
cd 123
123
```

This file assigns virtual diskette `financial.dat` to drive A:, turns off idle detection (in this case to do long recalculations), goes to the `lotus` directory, and starts `123`. `idle disable` in the batch file above is an example of typing a recognizable abbreviation of the full `idle-detection disabled` command to make it unambiguous.

You must turn off idle detection to do long recalculations because OpenPC “sleeps” if it detects no user input from keyboard or mouse for 45 seconds. See `idle detection` in the concepts section of Chapter 1 for more information.

See the section on batch processing in your DOS manual if you need more information on how to create batch files.

Accessing DG/UX Files from within DOS

OpenPC can map DG/UX files to DOS files on a one-to-one basis. This permits transparent access to DG/UX files from any DOS application without modifications to the DOS program.

Two DOS programs are supplied with OpenPC to map DG/UX files to DOS files:

- `DOSMOUNT.EXE`
- `REDIR.EXE`

Together these constitute the Redirector utilities. The Redirector utilities map DG/UX directories to DOS disk drives above D:

(drives C: and D: are reserved for virtual disks). Thus drives E: up to Z: can be redirected to point to different directories within the DG/UX file system. Chapter 6, *Using DOSMOUNT and REDIR*, describes these programs in detail. This section explains supporting concepts, which are

- the DG/UX redirected drive (for example, Z:)
- the \$HOME redirected drive (for example, H:)
- file name mapping

The \$HOME and DG/UX redirected drives can be any letter otherwise unused for disk drive names. By default OpenPC uses H: or Y: for the \$HOME DG/UX directory and Z: or U: for another DG/UX directory. These defaults are controlled by the DOS AUTOEXEC.BAT file. Drive names A: through D: are reserved for DOS drives.

You can change these drive letter designations by including the DOSMOUNT command in the AUTOEXEC.BAT file, or by issuing the DOSMOUNT command from the DOS prompt.

Below is an example of an AUTOEXEC.BAT file that assigns H: to the user's home directory and a DG/UX directory /usr/opt/dosbin/mc/bin to Z:.

```
prompt = $p$g
path c:\;c:\dos;c:\utl;z:\usr\opt\dosbin\mc\bin
redir
dosmount h:=$HOME
dosmount Z:/usr/opt/dosbin/mc/bin
```

NOTE: You can use either slash (/) or backslash (\) with the DOSMOUNT command.

Accessing DG/UX Files Through Redirected Drives

You can access all DG/UX files through redirected drives. Any drive letter except A:, B:, C:, and D: may be used as the redirected drive. The redirected drives are available when you activate the Redirector (with the **REDIR** and **DOSMOUNT** commands).

DG/UX access privileges apply to redirected drive files.

You may set up more than one redirected drive so that different areas of the DG/UX file system may be pointed to. NFS™ (Network File System) mounted directories may also be used as the mount point for redirected drives by their local name. (The expression “mount point” means the DG/UX directory the root of the redirected drive points to.)

If your **AUTOEXEC.BAT** file contained the following:

```
prompt $p$g
path c:\;c:\dos;c:\utl
redir
dosmount h:=$HOME    e:/usr/opt/dosbin
dosmount f:/         g:/mav4/dosbin
```

and if your directory **/mav4/dosbin** is the NFS mount for **mav4:/usr/opt/dosbin** where **mav4** is a remote host, then redirected drives and what they point to on the DG/UX file system are as follows:

E:	/usr/opt/dosbin	[local host]
F:	/	[root of local host]
G:	/mav4/dosbin	[NFS mounted directory]
H:	Your home directory on the local host	

Accessing Virtual Disks

OpenPC attempts to open the DG/UX file representing a virtual disk with Read/Write permissions. If another OpenPC user already has the file open, the file is opened to the second user with read permission only.

This limitation prevents multiple sessions of OpenPC from accessing the same file and colliding when writing to the virtual disk. This limitation can occasionally cause a problem for a program running on a virtual disk, as applications assume write access.

If such a problem in attempting to write occurs, you will get a DOS error message,

```
Write protect error writing drive C
Abort, Retry, Ignore?
```

You can select A to abort, and then specify to the program to use a redirected drive, or you can exit OpenPC. If you choose to exit OpenPC, you may copy a new virtual disk to your local directory, edit `openpc.dat` to use the new virtual disk, and restart OpenPC.

When a diskette drive is released (either a virtual diskette or a physical diskette), the file is closed (or the device is deallocated if it is a physical device). If the drive is accessed again before re-booting or assigning another diskette, an attempt is automatically made to regain access to the previously released diskette. For example, if you release drive B:, thereby closing the file and then go to drive B: under DOS and do a `dir`, the virtual disk file is reopened automatically. This virtual disk file is reopened automatically as long as other users have not already assigned B:.

If another user opens a virtual diskette before you do, you get only read access. If a physical diskette is being used, only one user can have access at a time. The other user cannot write to the diskette.

Private Virtual Disks

By placing the appropriate entries in the `openpc.dat` file, you can make the virtual disk a private virtual disk (exactly like using C: drive on any independent PC). To create a private virtual disk, place the DG/UX file that represents the virtual drive in your home directory.

The `openpc.dat` file entry might look like this:

```
FIRST_HARD_DISK $HOME/dosc.dat
```

It is highly recommended that you place copies of the virtual disk files in your home directory for security and data privacy. Also set file permissions to reflect this, using the DG/UX `chmod(1)` command to set 0644 (Read/Write for you, the owner, and Read only for the group and others).

File Name Mapping Using Redirected Drives

OpenPC maps all DG/UX file names to DOS file names when you access the DG/UX files through a redirected drive.

DOS file names have the format:

```
name.[ext]
```

where *name* is 1 to 8 characters long. The extension (*ext*) is optional, and if present, is from 1 to 3 characters long. The dot (.) is always implied at the end of the name and *must* be explicit if an extension is specified. All file names in DOS are mapped to uppercase. Thus, `test.doc`, `TEST.DOC`, and `TesT.Doc` all map to `TEST.DOC`. In DG/UX these are 3 separate files but they are only one file in DOS.

DOS has problems with DG/UX file names because:

- DG/UX file names can be longer than DOS file names.
- A dot (.) character can appear anywhere in a DG/UX file name, and can appear multiple times.
- DG/UX file names can mix uppercase and lowercase characters.
- DG/UX file names can have characters which are not allowed in DOS file names.
- Unlike DOS, DG/UX uses file names to access physical devices.

To overcome these restrictions, OpenPC maps all DG/UX file names which are not allowed in DOS to acceptable DOS names.

File name mapping is performed dynamically. It generates a unique name based on the order in which mapped names are requested. Consequently, a DG/UX file name can map to different DOS file names at different times.

When naming files under DG/UX which you may later wish to access from OpenPC, try to

1. use DOS conventions for length of name.
2. avoid characters which are not allowed under DOS. Examples: non-printing characters, * (asterisk).
3. use lowercase for file names.

A description of the mapping algorithm follows.

NOTE: DG/UX directory names that you would wish to access from OpenPC (via a redirected drive) should also follow these rules.

Mapping Algorithm

- File name mapping uses the first five characters of the original file name, if possible.
- All letters are forced to uppercase.
- Any characters not allowed in DOS are replaced by a tilde (~) character.
- Any allowable DOS extension is appended.
- When a file name undergoes a mapping other than lower to uppercase, OpenPC assigns a two-character index immediately before the explicit or implied dot character.
- When a file name is shortened, OpenPC precedes the index characters with a tilde (~).
- Any files with uppercase letters will be mapped.

NOTE: This algorithm always yields a name with a tilde (~) character replacing any illegal character. This algorithm lets you do wildcard searches. See examples in the following table.

Examples of File Name Mapping

The following table shows examples of name mapping. In the table, XX represents a two-character map table index. The XX can be any two capital letters; OpenPC selects these while generating the map. Whenever a name is shortened, OpenPC precedes the index (XX) with a tilde character.

DG/UX File Name	Mapped to DOS File Name under OpenPC	Explanation
abc123.def	ABC123.DEF	Lower to uppercase.
a	A	Lower to uppercase.
A	A~~~~XX	Uppercase requires mapping. Notice the two-character index at the end of the mapped name.
a_long_name	A_LON~XX	Mapped due to length. OpenPC inserts the tilde before the two-character index to indicate the shortened name.
AB.c	AB~~~~XX.C	Mapped using available legal characters.
Ab.C	AB~~~~XX.C	The value of the two-character index will differ from the immediately preceding example. OpenPC avoids overwriting files by changing the index value during mapping.
a.b.c	A~B~~~XX.C	Too many dots. Notice that the suffix maps to the .C extension.
abcd.efgh	ABCD~~XX	Requires mapping because the suffix is greater than three characters.
.login	~LOGI~XX	A leading period (.) always makes a name illegal.

Printing from OpenPC

OpenPC supports the maximum of three printer ports allowed under DOS. These are named **LPT1**, **LPT2**, and **LPT3** under DOS.

The **openpc.dat** file is used to configure the three printer ports so that printing activity within DOS is sent to DG/UX print queues, files, or filters. This allows great flexibility in printing.

For example, suppose you run OpenPC on an AViiON workstation that has a printer capable of printing IBM PC AT-generated graphics attached to its parallel printer port. If your **openpc.dat** file has the following entries,

```
LINE_PRINTER_1    /dev/lp
LINE_PRINTER_2    cat >/tmp/dosprint.out
LINE_PRINTER_3    pr|lp -dlj2
```

then DOS print activity to

1. LPT1 will be sent directly to the printer.
2. LPT2 will be sent to **cat(1)**, as **cat**'s **stdin**. Then **cat(1)** will redirect this input to the file **/tmp/dosprint.out**. You could later use this file for DG/UX editing, etc.
3. LPT3 will be sent to **pr(1)**, as **pr**'s **stdin**. Then **pr(1)** will pipe the **pr**'d file to the DG/UX **lp** print queue facility which has printer **lj2** specified as the device to print on. Thus in this case, DOS printing activity is sent to a filter, **pr(1)**, which pipes the filtered file to the DG/UX print spooler.

Using the Mouse (X Windows only)

OpenPC has to share the mouse with X Windows. That is, the mouse serves either OpenPC or X Windows at any given instant. The mouse serves X Windows by default and is called an X mouse. To change this default, however, and attach the mouse to OpenPC, you can either

- click `[MB2]`, or
- use `opencmd` or the mouse toggle button selection in the **Customize** menu.

To re-attach the mouse to X Windows, use `[MB2]` or the command `opencmd mouse host`.

Also note that when the mouse is attached to OpenPC,

1. the X Windows mouse cursor disappears.
2. the DOS mouse cursor is confined to the OpenPC window. To make a selection from the menu bar, you have to re-attach the X mouse by clicking `[MB2]` again. (The mouse cursor may be invisible. See last paragraph of this chapter.)

DOS applications require that a DOS mouse driver be loaded so that the applications can use the mouse. If your application asks you to identify the type of mouse you are using so that the appropriate mouse driver can be loaded, select the Microsoft Bus Mouse. OpenPC emulates only this type of mouse. If your application doesn't ask you to identify the mouse, use the Microsoft Bus Mouse driver. The driver is usually loaded by adding a line similar to the following:

```
device=mouse.sys
```

in the DOS `CONFIG.SYS` file.

NOTE: To avoid licensing violation issues, you must be a licensed user of the Microsoft Bus Mouse.

See your DOS manual for information on loadable device drivers and **CONFIG.SYS**. Also see your Microsoft Bus Mouse documentation for information on loading a mouse driver.

Also note that each individual DOS application draws its own mouse cursor. Thus, if you attach the mouse to OpenPC and only DOS is running, the mouse cursor disappears. Start your application to produce the cursor.

Using DOS Applications

This chapter explains how to

- manage disk space within OpenPC.
- install DOS applications on OpenPC.
- run DOS applications within OpenPC.

See your product manual for specific information on running a particular DOS application.

Where to Install Your Applications and Store Data

Because OpenPC emulates the hard disk of the PC AT by using a DG/UX file, you might install the application on your own C: or D: virtual disk. This wastes DG/UX disk space when there is more than one user of an application. When you want to share program files, you should use OpenPC's capability to access the DG/UX file system (through the Redirector utilities described in Chapter 6) and install DOS applications in a subdirectory.

Each OpenPC user then **DOSMOUNTS** (see Chapter 6) this subdirectory and runs the DOS application from the subdirectory. Most DOS applications create and manipulate data. The best way to store your data is to put it in your home directory on a virtual disk or through the Redirector. Storing it this way provides privacy for your data. Thus the stored DOS programs reside in a DG/UX subdirectory and the data they create or manipulate resides on your private virtual disk.

NOTE: You should check with your system administrator to see if a licensed network version of each application exists. You have the responsibility to determine if this scheme for storing the DOS application is in accordance with your license. Store DOS applications on private virtual disks if you are unsure.

Accessing Physical DG/UX Diskettes

Unlike the PC AT's diskette drive, the OpenPC diskette drive is shared among all users. Assignment of ownership is therefore necessary. Use the **Assign Diskette** selection from the **Customize** menu, described in Chapter 4 or make the appropriate entry in `openpc.dat` as described in Chapter 3.

If a diskette is assigned in your `openpc.dat` file, it will be opened for your exclusive use when you invoke OpenPC. The diskette drive will be locked to other users. Therefore, be sure to release the diskette for use by other DG/UX or OpenPC users.

Installing and running copy-protected applications is not supported.

Installing DOS Applications

DOS applications typically are supplied on 360 KB 5.25-inch diskettes. To copy these to a redirected drive, edit `openpc.dat` to ensure that DOS drive A: is mapped to use the DG/UX physical diskette drive (for example, `/dev/rpdsk/3`). Then start OpenPC, and make sure your DG/UX DOS application storage directory is **DOSMOUNTed**.

Refer to installation instructions that come with your DOS application.

Many PC applications have an **INSTALL** utility for placing the application on a hard disk. You can use this utility under OpenPC.

Be sure to release the diskette once the DOS application is installed. Use the **Release Diskette** selection from the **Customize** menu, described in Chapter 4.

Some older DOS applications cannot be installed and/or run from a redirected drive. These have to be installed on the C: or D: virtual disks. Make sure that you can write to the C: drive before installing such applications.

The **Device Status...** selection from the menu bar (OpenPC running under X11) or the `opencmd devstatus` command will display the DG/UX access permissions on a virtual disk.

Using the F10 Key in DOS Applications

The OSF/Motif environment uses the F10 key differently from the way it is used on a PC. DOS applications therefore will not respond properly to the normal F10 keystroke in OpenPC.

You must activate the Caps Lock key first, and then you can use the F10 key for DOS applications in OpenPC.

Using DOSMOUNT and REDIR

This chapter describes:

- enabling redirected drives with **REDIR**.
- assigning DG/UX directories to redirected drives with **DOSMOUNT**.

The two DOS programs **REDIR** and **DOSMOUNT** make up the Redirector. Both are necessary to access the DG/UX file system from within DOS.

REDIR enables the Redirector, so that you can access drive letters greater than D:.

DOSMOUNT provides access to the DG/UX file system from DOS by assigning drive letters to directories in the DG/UX hierarchy.

NOTE: Under DOS, the terms “networked drive” (used by DOS) and “redirected drive” (used for OpenPC) are equivalent.

Enabling redirected drives: REDIR

To initialize the Redirector (to enable redirected drives):

1. Make sure the **DOS CONFIG.SYS** file includes the following lines (supplied as defaults). If it doesn't, use an editor to write in:

```
buffers=30
files=50
fcbs=25,25
lastdrive=z
stacks=0,0
```

The statement **LASTDRIVE=Z** must be in the **CONFIG.SYS** file to access all possible redirected drives (H: through Z:) from within DOS.

These are optimal values to enable redirected drives.

You can start the Redirector only after you reboot DOS.

2. Run the DOS utility **REDIR.EXE** from DOS. You can do this by either:
 - entering the line **REDIR** in your **AUTOEXEC.BAT** file, or
 - entering the **REDIR** command at any DOS prompt.

After initializing the Redirector, use the **DOSMOUNT** utility to assign DOS drive letters to the DG/UX directory hierarchy.

Assigning Redirected Disks: DOSMOUNT

REDIR enables the Redirector, so that you can access drive letters greater than D:. **DOSMOUNT** provides access to the DG/UX file system from DOS by assigning drive letters to directories in the DG/UX hierarchy.

You can use **DOSMOUNT** any time the DOS prompt is present and a **REDIR** command has been run. Before executing the **DOSMOUNT** command, the Redirector will not assign DOS volumes to the DG/UX file system.

Syntax

Entering **DOSMOUNT HELP** displays the message (examples use e: as the DOS drive letter):

```
usage: dosmount e: [-] [path-prefix]
```

NOTE: Spaces are not allowed between the colon (:) and a path prefix.

Use **DOSMOUNT** to select a DOS drive letter through which you can access the DG/UX file system.

dosmount e:	makes drive e: a networked drive
dosmount e:\usr\bin	makes e: a networked drive whose root is \usr\bin
dosmount e:\	makes e: a networked drive with root at / and eliminates the path-prefix for e:
dosmount e:-	disables drive e: as an available redirected drive
dosmount +r	enables the Redirector (default)
dosmount -r	disables the Redirector

dosmount	shows the Redirector status and lists all networked drives
dosmount e:=\$MY_PATH	makes drive e: a networked drive and sets the current directory of e: to the value of the \$MY_PATH environment variable (The root of e: is the root of DG/UX. If you omit the =, the root of e: will be the value of \$MY_PATH environment variable.)

You can use DG/UX shell environment variables as values for paths. So

```
dosmount e:=$HOME
```

sets mount points so that e: points to your home directory.

NOTE: You *do not* have to execute the **dosmount e:-** command before reassigning a drive letter to another path.

Also note that entering **dosmount** without parameters displays a listing with the characters ==> indicating mount points. (The expression “mount point” means the DG/UX directory the root of the redirected drive points to.) For example:

```
Hostname is 'hostname'  
Redirected drives:  
Drive E: \  
Drive F: \ ==> '\usr\romel\ect'  
Drive H: \OpenPC ==> '\usr\romel'  
Drive U: \USR\ROMEL  
Drive Z: \USR\ROMEL\REDIR  
For help on DOSMOUNT, type DOSMOUNT HELP
```

The root of F: is **/usr/romel/ect**. The current directory of U: is **/usr/romel**. You could do a DOS **cd ** on the U: drive and then the DG/UX directory pointed to by U:\ will be ****. If you do a **cd ** on F:, however, you would be in **\usr\romel\ect**.

Examples of DOSMOUNT Usage

The following examples assume you are running OpenPC under X11. If you are not, substitute “terminal screen” for “window” in the explanations.

Example 1

The command

```
DOSMOUNT F:
```

enables Drive F: as a valid DOS drive designation, makes F: available to the Redirector, and points the root directory of F: to the DG/UX root directory.

In the OpenPC DOS window you will see

```
Hostname is 'hostname'  
Redirected drives:  
  Drive F:\  
For help on DOSMOUNT, type DOSMOUNT HELP
```

Example 2

The command

```
DOSMOUNT F:\usr\tmp
```

enables Drive F: as a valid DOS drive designation, makes F: available to the Redirector, and points the root directory of F: to the DG/UX `/usr/tmp` directory.

In the OpenPC DOS window you will see

```
Hostname is 'hostname'  
Redirected drives:  
  Drive F:\ ==> '\usr\tmp'  
For help on DOSMOUNT, type DOSMOUNT HELP
```

Example 3

If you started OpenPC from a C shell (csh), the command

```
DOSMOUNT F:~
```

enables Drive F: as a valid DOS drive designation, makes F: available to the Redirector, and points the root directory of F: to the directory specified in the user's **\$HOME** environment variable. (~ is a synonym for your home directory under csh.)

In the OpenPC DOS window you will see

```
Hostname is 'hostname'  
Redirected drives:  
  Drive F:\ ==> '~'  
For help on DOSMOUNT, type DOSMOUNT HELP
```

Example 4

The command

```
DOSMOUNT F:-
```

disables Redirection to Drive F. Further access to this drive letter will not be possible unless you use **DOSMOUNT** to enable the drive.

In the OpenPC DOS window you will see

```
Hostname is 'hostname'  
For help on DOSMOUNT, type DOSMOUNT HELP
```


Example 5

The command

```
DOSMOUNT -r
```

disables the Redirector. No access to the DG/UX file system will be possible. The **DOSMOUNT +r** command must then be used to reactivate the Redirector.

In the OpenPC DOS window you will see, for **DOSMOUNT -r**,

```
REDIR is disabled
```

and for **DOSMOUNT +r**,

```
REDIR is enabled
```

Using Serial Terminals

This chapter describes:

- terminal setup
- limitations on OpenPC running on a serial terminal
- serial terminal personality modes
- troubleshooting

You can run OpenPC on the Data General D412+, D216+, and D462+ or later serial (asynchronous) terminals. These terminals have a PC personality mode that OpenPC enables when it is invoked from one of these terminals.

The PC personality mode enables:

- 25-line display
- PC AT extended character set display
- PC AT keyboard scan code input

Terminal Setup

If you can log onto the AViiON system from one of the DG serial terminals listed above, you are ready to run OpenPC on that terminal (provided you have OpenPC environment variables already set). Make sure the terminal is connected to the AViiON workstation at the highest baud rate allowed. Check with your system administrator or use the `stty` command to set baud rates.

You also *must* set the `TERM` environment variable to the terminal type `dg_pcterm`. This is because OpenPC uses the `TERM` variable to search the terminfo database and `dg_pcterm` is the name that `terminfo(4)` uses for these terminals in PC personality mode. If you can use a DG/UX editor like `vi` without problems, `TERM` is probably set correctly. Refer to your terminal documentation for more information on setting the `dg_pcterm` terminal type.

Make sure the terminal is in VT100™ or VT320™ mode when starting up OpenPC.

Limitations on Serial Terminals

Because a serial terminal is incapable of running the X Window System, the user interface menus described in Chapter 4 do not apply. Instead, you must use the DOS program `opencmd` to achieve the same results as with the menu-driven interface. `opencmd` is explained in Chapter 4.

OpenPC running on a serial terminal is limited to emulating an IBM monochrome video display. In some cases a DOS application requiring a CGA or Hercules adapter may not run or will have limited functionality. This means, for example, Lotus 1-2-3 will display spreadsheets, but cannot display graphs generated by the data contained in a spreadsheet.

Troubleshooting

There are three reasons why OpenPC may not display correctly when you are using a serial terminal:

1. Your **TERM** variable is set incorrectly. That is, your **TERM** environment variable does not match the terminal type you are using.
2. The **TERM** variable you are using does not have a terminal specification file with exactly the same name in **/usr/opt/OpenPC/term**.

As an example, if **echo \$TERM** displays **dg_pcterm**, then the terminal specification file for this terminal *must* be named **dg_pcterm** and should be in the **/usr/opt/OpenPC/term** directory. Carefully observe upper/lowercase when making your entries.

3. The terminfo database is not installed on the system.

The serial terminal can sometimes be in the wrong personality mode. For example, if OpenPC aborts (because of an error or because it is killed from another terminal—see the **kill(1)** command), it will leave the terminal in the PC personality mode. DG/UX, however, expects it to be in its native personality mode.

Restore the terminal to its default state by using the terminal set up menus.

Unsupported Serial Terminals

This appendix describes how you can install, configure, and run OpenPC on unsupported serial (asynchronous) terminals. You must do more work to create the necessary file than is required for any other aspect of OpenPC. Be prepared to study this chapter closely. “Unsupported” in the title of this chapter means Data General provides no more technical assistance than is presented in the following pages.

Although Data General supports only its D216+, D412+, and D462+ (or later) asynchronous terminals for running with OpenPC, you can run OpenPC on virtually any terminal provided the terminal has the required features. You must, however, be ready to describe the terminal to OpenPC using an ASCII text file called a *terminal specification file*. The name of the file is the same as the **TERM** variable for the particular unsupported terminal you wish to configure to run OpenPC.

NOTE: You can run OpenPC also on the Data General D412™, D216™, and D462™ terminals. These terminals do not have a PC personality mode, but do have the features required to run OpenPC in serial terminal mode. These features are listed below. Terminals D412/216/462 will run OpenPC in VT100 mode, using a terminal specification file.

Examples of other serial terminals OpenPC can run on are the Wyse® WY-60™ and the DEC™ VT™ 220. OpenPC includes the terminal specification file only for the Wyse WY-60.

Required Features

These terminal features are required in order to use OpenPC in serial terminal mode.

- A display of at least 24 rows by 80 columns.
- Direct cursor addressing (an escape sequence that allows cursor movement to any row and column).
- No autowrap (a mode in which writing a character in column 80 does not automatically advance the cursor to column 1 of the next row).
- An escape sequence for clearing the screen.
- Escape sequences for line insertion and deletion, or setting the scrolling region and reverse scrolling (like a VT220).

Optional Features

The following terminal features are highly desirable but not required for using OpenPC with a serial terminal. Their presence, however, yields better video emulation.

- A PC-compatible mode (or mode that can be entered and exited through escape sequences).
- Keys outside the basic ASCII set, such as function and arrow keys.
- A 25-line display (although a 24-line display is supported).
- Screen attributes:
 - bold intensity
 - blinking
 - reverse
 - underline
 - invisible

Option Descriptions

PC-Compatible Mode	A PC-compatible mode (also known as PC personality mode) that can be entered and exited through escape sequences. This feature is now only available on a few terminals. It is important because it allows the entire PC character set to be displayed, making the terminal close to a “real” PC console. It also generates keyboard scan codes which are used by PC applications instead of the ASCII codes. This is important because the translation between ASCII and scan codes needs to be done in software.
Extra (Non-ASCII) Keys	These keys are convenient in that they can frequently be used to represent PC function keys and arrow keys.
A 25-Line Display	Because typical PCs have 25-line displays, a 25-line terminal eliminates the need to use “scrolling” to show the top 24 or bottom 24 lines on a terminal that does not have this feature.
Screen Attributes	This option allows applications that use different screen attributes to display correctly. Examples of screen attributes are blinking, bolding, or underlining characters.

Overriding Terminal Definitions

To support the capability of running DOS applications on a serial terminal, OpenPC needs certain information about your terminal.

OpenPC uses the standard shell environment variable **TERM** to find out your terminal type. OpenPC then searches two different areas for specific information about how to run PC applications on that type of terminal. OpenPC overrides information found in an earlier source with that from a subsequent source. The sources are:

1. the terminfo database, where OpenPC gathers information on escape sequences sent to the terminal.
2. the OpenPC terminal-specific specification file. The name for this file is the same name as that assigned to the **TERM** user environment variable. See page 7-3. This file also must be located in the directory `/usr/opt/OpenPC/term`. For example, if **TERM** is set to `vt220`, then the terminal-specific file that OpenPC will use will be `/usr/opt/OpenPC/term/vt220`.

OpenPC uses this file's data to determine the way the non-ASCII keys of the terminal are to be translated.

Example

OpenPC will scan the following files, in the order listed.

`/usr/lib/terminfo/v/vt220`

Standard UNIX terminfo data base for a vt220 terminal.

`/usr/opt/OpenPC/term/vt220`

OpenPC terminal-specific file with the name that was set in the environment variable **TERM** (this file is required, but is not supplied — you must write this file).

See the file `/usr/opt/OpenPC/term/dg_pcterm` for examples of format. This file provides the support for the D412+, D216+, and D462+ terminals.

Terminal Descriptions

OpenPC requires a terminal definition in the **terminfo** database for any terminal type from which an OpenPC process is to be run. OpenPC automatically scans the definition when activated.

Necessary Terminal Information

OpenPC requires three types of information about a terminal:

- Escape key sequences for terminal capabilities.
- Mappings from ASCII sequences to scan code sequences.
- Mappings from scan code sequences to ASCII sequences.

You can specify all the information OpenPC needs in the OpenPC terminal configuration file format. The supported terminals have terminal specification files in the `/usr/opt/OpenPC/term` directory that you may use as examples.

Terminal Capabilities

You can specify any capability that the terminal physically has to increase the display efficiency or the accuracy of the PC emulation. The following are typical **terminfo** capabilities that may already be defined for your terminal. To find out if they are already defined in the **terminfo** database, use the DG/UX **tput(1)** command. If they are not defined but are available, you **must** add them to the terminal specification file for that terminal.

Capability	Description
clear	Clear screen and home cursor
ed	Clear to end of screen
el	Clear to end of line
csr	Change scrolling region (vt100)
ri	Scroll screen down one line
indn	Scroll screen up <i>n</i> (variable number of) lines
il1	Insert one line
il	Insert lines
dl1	Delete one line
dl	Delete lines
ich1	Insert one character
ich	Insert characters
dch1	Delete one character
dch	Delete characters
cup	Position cursor
home	Move cursor home
cub1	Move cursor left
cuf1	Move cursor right
cuu1	Move cursor up
cul1	Move cursor down

The table is continued below. Note that the first six commands have the same name as a corresponding **terminfo** capability. Only the semantics are slightly different. See “Screen Attribute Definitions” on the following page.

Capability	Description
sgr0	Normal mode
blink	Blink mode
bold	Bold mode
rev	Reverse video mode
invis	Invisible mode
smul	Underline mode
blbo	Blink and bold mode
blr	Blink and reverse mode
blu	Blink and underline mode
bor	Bold and reverse mode
bou	Bold and underline mode
blbor	Blink, bold and reverse mode
blbou	Blink, bold and underline mode
sopenpc	OpenPC starting string
fopenpc	OpenPC shutdown string
k_up	Up “rocker” sequence for 24-line terminals displaying 25-line screens
k_down	Down “rocker” sequence for 24-line terminals displaying 25-line screens
k_refr	Refresh sequence
pcompat	PC-compatible terminal

If you want any of these capabilities, you must add them to your **terminfo** file. To add a capability, type

```
capability<SPACE or TAB>escape sequence<SPACE or TAB>comment description
```

As an example from the Wyse WY-60 terminal specification file, type

```
blink \EGr * blink mode
```

NOTE: To determine the escape sequence, you must see the manual for your particular terminal.

OpenPC Capabilities

The following sections describe the OpenPC capabilities.

Subsection	Describes...
Screen Attribute Definitions	sgr0, blink, bold, rev, invis, smul, blbo, blr, blu, bor, bou, blbor, and blbou
Special Keys	k_up, k_down, and k_refr
pccompat	pccompat

Screen Attribute Definitions

The capabilities **sgr0, blink, bold, rev, invis, smul, blbo, blr, blu, bor, bou, blbor, and blbou** all define screen attributes that you can toggle (turn off and on). While the first six of these have the same name as a generic **terminfo** capability, the semantics are different. The other capabilities are new to **terminfo** definitions. If the terminal can support these attributes, all should be defined in one of the OpenPC-specific definition files.

For example, the differences between the ordinary terminfo **blink** and the OpenPC **blink** is that the former means “turn on blinking without changing any existing attributes.” Thus, if reverse video were already on, it would remain on. However, the OpenPC “blink” means “turn on blinking and turn off all other attributes.” The same is true of the other twelve definitions. On an ANSI terminal, the terminfo **blink** is defined as `\E[5m`, while the OpenPC **blink** is `\E[0;5m`.

Special Keys

You can define three special ASCII sequences for non-PC-compatible terminals. These sequences correspond to pressing a single key on an incompatible terminal.

The capabilities **k_up** and **k_down** support 24-line terminals. Defining either of these capabilities:

- tells OpenPC that the terminal has only 24 lines.
- defines a sequence that “rocks” the screen up or down, showing either the first or last 24 lines of the true 25-line display.

You can use identical definitions for **k_up** and **k_down**. If you define only one of these, the same key rocks the screen alternately up and down.

On a 24-line terminal within OpenPC, you should scroll with the screen “rocked up” (with the “25th” line showing). If the screen is rocked down, scrolling will be much slower, and the “25th” line will be invisible.

You can clear messages from the screen with a refresh key. The **k_refr** capability does so.

pccompat

OpenPC must know whether your terminal is PC-compatible or not, so that it knows whether to expect scan codes or ASCII characters from the keyboard.

Give **pccompat** the value **y** on PC-compatible terminals. Assign the value **n** for incompatible terminals in the terminal specification file.

ASCII Input to Scan Code Mappings

PC-compatible terminals transmit numerical values called scan codes with each keystroke. These scan codes correspond to the position of the key on the keyboard. Separate scan codes indicate when a key is pressed and when it is released (“make” and “break” codes).

On incompatible terminals, key sequences result in ASCII values. There must be a mapping between ASCII input characters and the sequence of scan codes that a PC application expects to receive.

For example, the ASCII **a** key, when shifted, transmits (octal) 101, the ASCII code for **A**. However, a PC application expects to see the (hex) scan codes 2A, 1E, 9E, and AA in that order, corresponding to the make code for the **Shift** key, the make and break codes for the **a** key, and the break code for the **Shift** key.

If your terminal is PC-compatible, these mapping specifications have no meaning. However, having them specified will not hurt anything. The reason nothing will be hurt is because the **pccompat** capability in the terminal-dependent specification file causes OpenPC to ignore these mappings.

There are a number of keys on a PC keyboard that do not correspond to any ASCII character:

- the ten function keys,
- the three toggle keys: <CAPS LOCK>, <NUM LOCK >, <SCROLL LOCK >, and
- the nine-key numerical keypad, <INS>, , <SYS REQ>, <PRT SC>, - and + on the numerical keypad, <ALT>, <CTRL>, <LEFT SHIFT>, and <RIGHT SHIFT>.

These last four, while present on ASCII keyboards, do not transmit a character, but rather modify the character transmitted by other keys. To communicate with all PC applications, it must be possible to “tell” OpenPC that each of these keys has been pressed. In fact, in the case of <ALT>, <CTRL>, <LEFT SHIFT>, and <RIGHT SHIFT>, it is necessary for some applications to know separately when they have been pressed (make code) and when they have been released (break code).

The terminal configuration file has lines beginning with the word **input:** that allows for arbitrary mappings between sequences of input ASCII characters and sequences of scan codes. For example, if your terminal is extremely simple and has no “extra” keys whatsoever, you may decide to map each of the above PC keys to ASCII sequences that start with a <CTRL><Z> and have a second character that is some mnemonic for the PC key.

Part of this mapping might look like:

```
input:  \032-1   3B-BB           ^Z1=F1
input:  \032-2   3C-BC           ^Z2=F2
input:  \032-3   3D-BD           ^Z3=F3
input:  \032->   4B-CB           ^Z>=->
input:  \032-32  1D-2C-AC-9D     ^Z^Z=ctrl make,Z,ctrl break
```

NOTE: The last “^Z” column contains only comment.

The first of these lines says that when a user types a `<CTRL><Z>` (octal 032) followed by a `<1>` on an ASCII terminal, it will have precisely the same effect as if the user had pressed and released the `<F1>` key on a PC keyboard, since 3B and BB are the make and break codes for the `<F1>` key.

Similarly, the next two lines allow a user to emulate `<F2>` and `<F3>` respectively. The fourth line maps the `<->` key on the PC keyboard (the `<6>` on the numeric keypad) to `<CTRL><Z>` followed by the greater-than sign key on the ASCII keyboard.

Since the first character of any such multiple-character ASCII sequence is now special in that it no longer actually transmits its own, “natural” scan code sequence, we must also provide a sequence that does transmit its natural scan codes, which is what the fifth line in the preceding example does. It says that typing two `<CTRL><Z>`s will send the scan codes for the PC `Ctrl` make code, the `Z` make and break codes, and the `Ctrl` break code.

Note that the above ASCII sequences are completely arbitrary, whereas the scan codes correspond specifically to some key that we wish to emulate on the PC keyboard. Therefore, if your ASCII terminal has a number of “extra” keys that naturally transmit some sequence of ASCII characters (typically starting with `Esc`, octal 033), you could conveniently map that ASCII sequence to a corresponding key on the PC keyboard. Again, remember that if you follow this procedure, you must also provide a sequence for actually typing an `Esc` character (which is used by many PC applications).

The syntax of an input mapping line is:

```
input: ASCII_sequence scan_code_sequence
```

NOTE: You can put either a space or a tab between `sequence` and `scan`.

The ASCII sequence is a series of ASCII characters separated by dashes (-). An ASCII character is either a single printable character, or an octal representation of a character (a backslash followed by 3 octal digits). Since the actual characters **dash**, **backslash** and **asterisk** are special characters in a terminal configuration file, you must enter them using the octal representation just described if they are actual characters in an ASCII input sequence. The scan code sequence is a series of two-digit hex scan codes separated by dashes.

A number of sample mappings may be found in the various distributed OpenPC terminal configuration files located in `/usr/opt/OpenPC/term`.

Output Character Mappings

To support serial terminals, it is necessary to allow mappings from each of the 256 characters in the PC character set to other characters, since many of these 256 are not present on ASCII terminals. Moreover, even PC-compatible terminals require that control characters, namely octal 000 through 037 and 177 through 377, be somehow escaped if you want to display the actual PC character for that code.

In an OpenPC terminal configuration file, use **output:** lines to perform this mapping. The default output mappings simply map every character to itself, except the control characters mentioned above. It maps each of these to **esc** (octal 033), followed by itself (except for 000, which is mapped to a space, since 0 and space are identical in the PC character set). This default mapping will work on most PC-compatible terminals.

For non-PC-compatible terminals, you must decide what to map to each of the characters 001 through 037 and 177 through 377, since they are typically not present in an ASCII terminal character set. As with **input:** lines, **output:** lines allow total flexibility. For example, you can map all of the above characters to a period (.), or you can find the visually closest ASCII character to the corresponding PC character (see a PC Reference Manual for pictures of the PC character set). Moreover, if the ASCII terminal has a graphics mode that supplies some of these characters, you can map them to the escape sequence to get into graphics mode, followed by the graphics character, followed by the escape sequence to get out of graphics mode.

While this procedure sounds costly, most of the characters output by a PC application will be regular ASCII characters and the overhead may be acceptable.

The syntax of an output mapping line is:

```
output: character character_sequence * comment
```

The character sequence has the same syntax as an ASCII sequence in an input mapping line, as described above (except that the characters \200 through \377 make sense in this context, but not in the other context). The character is also the same, but is a single character rather than a sequence of characters.

Tested DOS Applications

The applications listed below are a representative sample of popular DOS programs that have been tested for compatibility with OpenPC. Earlier or later revisions of these applications may have different features that would result in incompatibilities with OpenPC.

Major features but not every possible option in the software have been tested.

In general, any well-behaved DOS application can be expected to run correctly under OpenPC.

AutoSketch® 1.01	Harvard™ Total Project Manager II 1.0
Autocad™ 9.0	Javelin PLUS™ 2.01
Base SAS® 6.03	Landmark Speed Test™
CEOWrite™ 2.10	Leading Edge® Word Processor 1.5
Crosstalk® Mark IV 1.01	Lotus® HAL 1.0
Data Ease™ 2.53	Lotus® 1-2-3® 2.01
dBase III® Plus 1.1	Lotus® Symphony® 2.0
Desqview™ 2.2	Microsoft C Compiler™ 5.1
EDIX®	Microsoft Excel™ 2.0
Framework II™ 1.10	Microsoft Flight Simulator
GW-BASIC® 3.2	Microsoft Windows™ 2.03
GrandView® 1.0	Microsoft Word™ 4.0
GraphWriter® II 1.0	Microsoft Works™ 1.05
Harvard™ Graphics 2.10	Mirror II® 3.6.12

Multimate® 1.0	Softterm® PC 2.10
Norton Utilities™ 4.5	SuperCalc® 4 1.1
Office Writer™ 6.0	Superkey® 1.16a
Pagemaker® 3.0	Time Line® 3.0
PC Labs Benchmark® Series 4.0, 2.0	TopView™ 1.12
PFS®: First Publisher 1.0	Turbo C® 2.0
Phoenix PC AT Diagnostic Tools	Turbo Pascal® 4.0
R:Base® for DOS 2.0	20/20™ 2.2
SAS/Graph® 6.03	Ventura Publisher® 2.0 with Professional Extension
Sidekick® 1.56a	WordPerfect® 4.2, 5.0
Sidekick® Plus 1.01	XyWrite® III Plus 3.5

Applications which don't run on OpenPC include those running in protected mode, such as Lotus 1-2-3 version 3.0. In addition, applications do not run if they access the DMA directly, such as Fastback Plus™.

Troubleshooting and Error Messages

This appendix describes some problems you may encounter when you run OpenPC. You can attempt the solutions suggested here, as well as consulting your system administrator.

This appendix also lists error messages, their level of severity, causes, and suggested remedies.

For other problems not covered in this appendix, review the appropriate section elsewhere in this manual and try the procedures described there again.

Because this manual does not discuss DOS and DG/UX in any comprehensive way, your best sources may be found among more experienced users, your system administrator, and texts on DOS, DG/UX, and UNIX.

Troubleshooting

This section lists problems with OpenPC you may encounter and their solutions.

Application Installed on a Redirected Drive Fails to Run

Sometimes an application program cannot find files on a redirected drive, such as H: or Z:.

The Z: drive is a valid drive under DOS. However, some applications are designed to use only disk drives C: and/or D:. If a problem occurs when running an application on drive Z: or H:, you can

- try using a “lower letter,” such as E:.
- try installing, then running, the application on virtual disks C: or D:.

F10 Key in DOS Responds Improperly

The OSF/Motif environment uses the F10 key differently from the way it is used on a PC. DOS applications therefore will not respond properly to the normal F10 keystroke in OpenPC.

You must activate the Caps Lock key first (when you are in your application). Then you can use the F10 key for DOS applications in OpenPC.

Writing to the Virtual Disk Fails

OpenPC attempts to open the DG/UX file representing a virtual disk with Read/Write permissions. If another OpenPC user already has the file open, you can open the file with read permission only.

If such a problem in attempting to write occurs, you will get a DOS error message,

```
Write protect error writing drive C
Abort, Retry, Ignore?
```

This limitation prevents multiple sessions of OpenPC from accessing the same file and colliding when writing to the virtual disk. This limitation can occasionally cause a problem for a program running on a virtual disk, as applications assume write access.

If such a problem occurs, either specify to the program to use a redirected drive or exit OpenPC. Then copy a new virtual disk to your local directory, edit `openpc.dat` to use the new virtual disk, and restart OpenPC.

Error messages

This section lists the error messages, ranks them as either **WARNINGS** or **FATAL** errors, and tells you what action you can take to fix the problem.

The messages have the format of a number in parentheses, followed by text. Messages with the same name but with different numbers each have a different cause, as explained in the text for each message. (**Insufficient memory available** messages and **Error encountered using COM device** messages are examples.) Different internal operations may lead to the same error message. The numbers are for technical support and for easier debugging. The numbers are intentionally non-sequential.

NOTE: **FQF** is an acronym for Fully Qualified File name. It is the full name for a file, including path.

(3) Unable to open <FQF> as DOS drive <drive letter>

[<drive letter> is the DOS disk (A or B)]

Severity: **WARNING**

Cause: Physical diskette open failed. Possible causes are bad name in **openpc.dat** for the diskette device or no diskette in drive.

User Action: Enter correct diskette device node name in **openpc.dat** (See your system administrator if you don't know the name). Put diskette in drive.

(4) Error while closing drive <drive letter>

[<drive letter> is the DOS disk (A or B)]

Severity: WARNING

Cause: Physical diskette close failed. Possible causes are bad name in **openpc.dat** for the diskette device or no diskette in drive.

User Action: Enter correct diskette device node name in **openpc.dat** (See your system administrator if you don't know the name). Put diskette in drive.

(5) Creating file <FQF> because the requested file does not exist

Severity: WARNING

Cause: Virtual disk or diskette doesn't exist. Possible cause is bad name in **openpc.dat** or someone has deleted or changed access to the named file.

User Action: This message will appear in a dialog box that asks for a Yes/No response to create. If you answer "Yes" to the create dialog, you will have to run the DOS commands:

```
fdisk
format /s
```

to make the just created virtual disk/diskette a bootable disk.

*Read **fdisk** and **format /s** discussions in your DOS manual to avoid potential data destruction.*

(6) Insufficient memory available

Severity: FATAL

Cause: OpenPC requested memory allocation for an internal buffer from DG/UX, but was unsuccessful. This may indicate insufficient process space.

User Action: This is a **SERIOUS** error and should result in a Software Trouble Report (STR) if it occurs. See your OpenPC release notes for instructions on filing an STR to Data General.

Sufficient memory for all practical purposes should always be available on a virtual memory system like DG/UX. See your system administrator first to determine the maximum process space for your system. Have your system administrator reconfigure your system to expand memory in small steps if necessary and try to reproduce the problem with each expansion. This way you can determine if insufficient memory is really the problem.

(12) Unable to open <FQF> as DOS drive <drive letter>

[<drive letter> is the DOS disk (C or D)]

Severity: WARNING

Cause: Virtual disk open failed. Possible causes are:

1. Virtual disk open failed and then you answered Yes to the “create a virtual disk” dialog box, but the create failed.
2. Virtual disk open failed and system failure code returned indicating no hope of creating virtual disk. Either the file system is full or you do not have the correct permissions to create the virtual disk.

User Action: Check permissions and free space on the DG/UX file system.

(15) Unable to use DOS printer device <printer number>

[<printer number> is 1 for LPT1, 2 for LPT2, 3 for LPT3]

Severity: WARNING

Cause: Could not create a DG/UX pipe to send printing activity to. Possible causes are

1. OpenPC has too many files open.
2. The DG/UX system file table is full.

User Action: 1. Reduce the number of redirected drives you have assigned.

 2. Use the **df(1m)** from the DG/UX command line to check free space and i-nodes. Remove some files if the number of i-nodes is zero. Note that file systems which are remotely mounted will always show the number of i-nodes as zero. To get the correct number of i-nodes for a remotely mounted filesystem, **rlogin** to that system host and then check i-nodes with **df**.

Do not do any printing from OpenPC until the problem is resolved.

(16) Unable to write to DOS printer <printer number>

[<printer number> is 1 for LPT1, 2 for LPT2, 3 for LPT3]

Severity: WARNING

Cause: OpenPC could not write to DG/UX pipe created for printing. Possible cause is the DG/UX file system is full.

User Action: Check free space on the DG/UX file system using the **df(1m)** command. Do not do any printing from OpenPC until the problem is resolved.

(17) Error while closing destination printer <printer number>

[<printer number> is 1 for LPT1, 2 for LPT2, 3 for LPT3]

Severity: WARNING

Cause: OpenPC could not close the DG/UX pipe created for printing. Possible causes are:

1. **close(2)** call was interrupted.
2. An I/O error occurred while reading from or writing to the file system.

User Action: Retry the operation that caused the error. File an STR if the error persists. See your OpenPC release notes for instructions on filing an STR to Data General.

(18) Cannot load ROM file: <FQF>

[<FQF> is the name of the ROM BIOS image file as found in `openpc.dat` or will be the built-in default name which is `/usr/opt/OpenPC/defaults/rom.dat`.]

Severity: FATAL

Cause: OpenPC could not load the ROM BIOS image file.
Possible causes are:

1. Incorrect path or file name in `openpc.dat`. Path name should be absolute path of where `rom.dat` is located.
2. ROM BIOS image file has been truncated. It should always be 32768 bytes long.

User Action: Restore the ROM BIOS image file. Ask your system administrator for the backup.

(19) Cannot open font

[is the name of X11 font as specified in **openpc.dat**.]

Severity: **WARNING**

Cause: OpenPC is unable to access either or the default fonts, fixed and 6x13. At least one should be available.

User Action: 1. Have the X11 server re-read the font's directories by issuing the command:

```
                  /usr/bin/X11/xset fp rehash
```

 2. Make sure is correct in **openpc.dat**.

 3. Make sure , fixed or 6x13, is installed on the system that is running the X11 server you are using. See the system administrator if you do not know how to determine this.

(20) Insufficient memory available

Severity: FATAL

Cause: OpenPC requested memory allocation for an internal buffer from DG/UX, but was unsuccessful. This may indicate insufficient process space. The memory was to be used for OpenPC's video image buffer.

User Action: This is a **SERIOUS** error and should result in a Software Trouble Report (STR) if it occurs. See your OpenPC release notes for instructions on filing an STR to Data General.

Sufficient memory for all practical purposes should always be available on a virtual memory system like DG/UX. See your system administrator first to determine the maximum process space for your system. Reconfigure your system if necessary and try to reproduce the problem.

(21) No mapping specified for the key <key number>

[<key number> is the decimal number returned by the X11 server to OpenPC.]

Severity: **WARNING**

Cause: **There is no translation code for the input key indicated by < key number >. Either you are not using the correct keyboard translation file for the X11 server OpenPC is connected to or there is an error in the keyboard translation file.**

User Action: 1. Check the **KEYBOARD_TRANSLATIONS** entry in **openpc.dat** to make sure the keyboard translation file corresponding to the X11 server OpenPC is connected to is correctly specified.

 2. Check with your system administrator to ensure that a keyboard translation file exists (and is specified in **openpc.dat**) for the particular X11 server OpenPC is connected to.

(23) Cannot open keyboard file: <FQF>

Severity: WARNING

Cause: OpenPC could not open the keyboard translations file named <FQF> and specified by the **KEYBOARD_TRANSLATIONS** keyword in **openpc.dat**. Possible causes are:

1. Incorrect file/path name in **openpc.dat**.
2. You don't have read access permission for the file.
3. File does not exist or was erased on your system.

User Action: 1. Check file and path name in **openpc.dat**.
 2. Check permissions and existence.

(24) Insufficient memory available

Severity: FATAL

Cause: OpenPC requested memory allocation for an internal buffer from DG/UX, but was unsuccessful. This may indicate insufficient process space. The memory was to be used for OpenPC's shadow video image buffer.

User Action: This is a **SERIOUS** error and should result in a Software Trouble Report (STR) if it occurs. See OpenPC release notes on filing an STR. Sufficient memory for all practical purposes should always be available on a virtual memory system like DG/UX. See your system administrator first to determine the maximum process space for your system. Reconfigure your system if necessary and try to reproduce the problem.

(25) Error encountered using COM device

Severity: FATAL

Cause: An **ioctl(2)** system call to the tty line device node (as specified by the **COM_PORT1** or **COM_PORT2** keyword in **openpc.dat**) failed. The **ioctl(2)** call was to set line parameters.

User Action:

1. Ensure that a **getty(1m)** is NOT running on the tty line.
2. Ensure that you have read and write permissions on the device.
3. Comment out the **COM_PORT[1,2]** specification in **openpc.dat** until the problem is resolved. You will not be able to use the DOS COM1 or COM2 device until the problem is resolved.

(26) Error encountered using COM device

Severity: FATAL

Cause: An **ioctl(2)** system call to the tty line device node (as specified by the **COM_PORT1** or **COM_PORT2** keyword in **openpc.dat**) failed. The **ioctl(2)** call was to set modem parameters.

User Action:

1. Ensure that a **getty(1m)** is NOT running on the tty line.
2. Ensure that you have read and write permissions on the device.
3. Ensure that you have a modem (not a terminal) connected to the tty line using a modem cable.
4. Comment out the **COM_PORT[1,2]** specification in **openpc.dat** until the problem is resolved. You will not be able to use the DOS **COM1** or **COM2** device until the problem is resolved.

(27) Insufficient memory available

Severity: FATAL

Cause: OpenPC requested memory allocation for an internal buffer from DG/UX, but was unsuccessful. This may indicate insufficient process space. The memory was to be used for a communications control structure as part of the DOS COM device emulation.

User Action: This is a **SERIOUS** error and should result in a Software Trouble Report (STR) if it occurs. See your OpenPC release notes for instructions on filing an STR to Data General.

Sufficient memory for all practical purposes should always be available on a virtual memory system like DG/UX. See your system administrator first to determine the maximum process space for your system. Reconfigure your system if necessary and try to reproduce the problem.

(28) Unable to use DOS COM device <FQF>

Severity: WARNING

Cause: OpenPC could not open the device node specified for DOS COM[1,2] emulation. Possible causes are:

1. Incorrect specification of the file name as specified by the keyword COM_PORT_[1,2] in **openpc.dat**.
2. Access permissions on <FQF>.

User Action: 1. Check the file name in **openpc.dat**.
 2. Check access permissions for the file.
 3. Comment out the COM_PORT[1,2] specification in **openpc.dat** until the problem is resolved. You will not be able to use the DOS COM1 or COM2 device until the problem is resolved.

NOTE: This error will ask if you want to continue. If you answer affirmatively, you will not be able to use the DOS COM device until the problem is resolved.

(29) Error encountered using COM device

Severity: WARNING

Cause: An **ioctl(2)** system call to the tty line device node (as specified by the **COM_PORT1** or **COM_PORT2** keyword in **openpc.dat**) failed. The **ioctl(2)** call was to get line parameters.

User Action: 1. Ensure that a **getty(1m)** is NOT running on the tty line.

 2. Ensure that you have read and write permissions on the device.

 3. Comment out the **COM_PORT[1,2]** specification in **openpc.dat** until the problem is resolved. You will not be able to use the DOS **COM1** or **COM2** device until the problem is resolved.

(30) Error encountered using COM device

Severity: FATAL

Cause: An **ioctl(2)** system call to the tty line device node (as specified by the **COM_PORT1** or **COM_PORT2** keyword in **openpc.dat**) failed. The **ioctl(2)** call was to get modem parameters.

User Action:

1. Ensure that a **getty(1m)** is NOT running on the tty line.
2. Ensure that you have read and write permissions on the device.
3. Ensure that you have a modem (not a terminal) connected to the tty line using a modem cable.
4. Comment out the **COM_PORT[1,2]** specification in **openpc.dat** until the problem is resolved. You will not be able to use the DOS COM1 or COM2 device until the problem is resolved.

(32) Insufficient memory available

Severity: FATAL

Cause: OpenPC requested memory allocation for an internal buffer from DG/UX, but was unsuccessful. This may indicate insufficient process space. The memory was to be used for a communications buffer as part of the DOS COM device emulation.

User Action: This is a **SERIOUS** error and should result in a Software Trouble Report (STR) if it occurs. See your OpenPC release notes for instructions on filing an STR to Data General.

Sufficient memory for all practical purposes should always be available on a virtual memory system like DG/UX. See your system administrator first to determine the maximum process space for your system. Reconfigure your system if necessary and try to reproduce the problem.

(34) Error encountered using COM device

Severity: WARNING

Cause: An **fcntl(2)** system call to the tty line device node (as specified by the **COM_PORT1** or **COM_PORT2** keyword in **openpc.dat**) failed. The **fcntl(2)** call was to set line parameters.

User Action:

1. Ensure that a **getty(1m)** is NOT running on the tty line.
2. Ensure that you have read and write permissions on the device.
3. Comment out the **COM_PORT[1,2]** specification in **openpc.dat** until the problem is resolved. You will not be able to use the DOS **COM1** or **COM2** device until the problem is resolved.

(35) Error encountered using COM device

Severity: FATAL

Cause: An `ioctl(2)` system call to the tty line device node (as specified by the `COM_PORT1` or `COM_PORT2` keyword in `openpc.dat`) failed. The `ioctl(2)` call was to get modem parameters.

User Action:

1. Ensure that a `getty(1m)` is NOT running on the tty line.
2. Ensure that you have read and write permissions on the device.
3. Ensure that you have a modem (not a terminal) connected to the tty line using a modem cable.
4. Comment out the `COM_PORT[1,2]` specification in `openpc.dat` until the problem is resolved. You will not be able to use the DOS `COM1` or `COM2` device until the problem is resolved.

(36) Error encountered using COM device

Severity: FATAL

Cause: An **ioctl(2)** system call to the tty line device node (as specified by the **COM_PORT1** or **COM_PORT2** keyword in **openpc.dat**) failed. The **ioctl(2)** call was to set line parameters.

User Action:

1. Ensure that a **getty(1m)** is NOT running on the tty line.
2. Ensure that you have read and write permissions on the device.
3. Comment out the **COM_PORT[1,2]** specification in **openpc.dat** until the problem is resolved. You will not be able to use the DOS **COM1** or **COM2** device until the problem is resolved.

(37) Error encountered using COM device

Severity: WARNING

Cause: An **ioctl(2)** system call to the tty line device node (as specified by the **COM_PORT1** or **COM_PORT2** keyword in **openpc.dat**) failed. The **ioctl(2)** call was to set the **BREAK** line parameter.

User Action:

1. Ensure that a **getty(1m)** is NOT running on the tty line.
2. Ensure that you have read and write permissions on the device.
3. Comment out the **COM_PORT[1,2]** specification in **openpc.dat** until the problem is resolved. You will not be able to use the DOS **COM1** or **COM2** device until the problem is resolved.

(38) Error encountered using COM device

Severity: WARNING

Cause: OpenPC could not write to the device node specified for DOS COM[1,2] emulation. Possible causes are:

1. Access permissions.
2. Another process has locked the device node.

User Action: 1. Check access permissions for the file.
 2. Comment out the COM_PORT[1,2] specification in **openpc.dat** until the problem is resolved. You will not be able to use the DOS COM1 or COM2 device until the problem is resolved.

NOTE: This error will ask if you want to continue. If you answer affirmatively, you will not be able to use the DOS COM device until the problem is resolved.

(57) Opening file <FQF> with Write access even though lock request failed

Severity: WARNING

Cause: OpenPC attempts to lock virtual disks/diskettes when they are first opened to forestall the possibility of two different OpenPC sessions attempting to write to the same virtual disk/diskette simultaneously, possibly corrupting the virtual disk/diskette. If the lock fails, this message appears, and you are asked if you want to continue despite the failure. Possible causes of lock failure are:

1. Access permissions: you may not have write permissions for the DG/UX file representing the virtual disk in question.
2. The virtual disk file exists on an NFS-mounted file system. Locks cannot be set on remote file systems.

User Action:

1. Change the virtual disk file in **openpc.dat**.
2. Make a local copy of the virtual disk file, if it exists on the NFS-mounted file system.

NOTE: If you decide to continue despite the lock failure, the possibility exists that if another OpenPC session uses the same virtual disk file, a write collision could occur. Ensure that other OpenPC sessions access different virtual disk files than yours.

Glossary

AT-Compatible Software

Software designed to run on the IBM PC AT or clone.

Background

Background and foreground have special meaning under OpenPC. Specifically, “foreground” denotes “displayed on the screen.” Conversely, “background” relates to non-viewed applications when they are running. Under DG/UX, background is a type of program execution in which you request the shell to execute a command away from what you are presently doing, that is to say, “in the background.” See *foreground*.

BIOS

Acronym for Basic Input/Output System. In PCs, the BIOS is information contained in ROM (Read Only Memory). The BIOS is used by DOS to provide access and control of the computer’s hardware.

Console

A console is a combination of a display device and a keyboard device. A serial terminal is treated as a console.

Environment

A set of strings available to applications running under an operating system such as UNIX or DOS. For example, the shell environment includes such components as your shell prompt string, specifics for backspace and erase characters, commands for sending output from your terminal to the computer, and search path. Both DG/UX and DOS have environments, but the two are unrelated and cannot be shared.

NOTE: In the special case where drives may be redirected to **\$HOME**, it may appear as if DOS and DG/UX can share. Even in this case, DOS applications don't pick up the DG/UX environment. For example, if DOS programs are in the DG/UX files in **/usr/opt/dosbin**, and this directory is in a DG/UX path, you will not see this path in OpenPC.

Foreground

Background and foreground have special meaning under OpenPC. Specifically, "foreground" relates to viewed and "background" relates to non-viewed applications when they are running. Foreground is the normal type of program execution. Under DG/UX, the shell waits for a command to end before prompting you for another command when run in foreground mode. See *background*.

Ill-Behaved Applications

IBM PC AT application programs that sometimes bypass the BIOS and access the hardware directly. To improve performance, ill-behaved applications take advantage of the unprotected PC environment to manipulate hardware control registers. On larger machines, these registers are accessible only to the operating system, but on PCs they can be bypassed.

Multitasking

The ability of an operating system to execute more than one task or process at the same time. DG/UX and UNIX are operating systems which allow multitasking. DOS does not allow multitasking.

Multiuser

The ability of an operating system to support several users at the same time. DG/UX is a multiuser operating system. DOS is not a multiuser operating system.

Path

The trail from one point in the hierarchy of directories to another position within the hierarchy.

Piping

A method of redirecting the output of any command to be the input of another command in DG/UX and UNIX. Piping is named for the character (|) that redirects the output. For example, the shell command `who | wc -l` pipes output from the `who(1)` command to the `wc(1)` command, telling you the total number of people logged into your multiuser system.

Pipeline

In DG/UX and UNIX, a series of filters separated by the pipe character (|). The output of each filter becomes the input of the next filter in the line. The last filter in the pipeline writes to standard output. (A filter is a program acting on an input stream so as to transform the stream in some way. The transformed stream is the output of the filter.) Generally the streams are `stdin` and `stdout`.

Printer Daemon

A program routine that stays in memory and intercepts and processes printer I/O for printer spooling.

Process

Generally a program that is at some stage of execution. In the DG/UX system, a process also refers to the execution of a computer environment, including contents of memory, register values, name of the current directory, status of files, and information recorded at login time.

Virtual Disk

An exact sector-by-sector image of a PC-type hard disk stored as a file on the DG/UX file system. It appears as data to DG/UX.

Scan Code Mapping

Scan codes are the values that an IBM PC sends when a key is pressed on the keyboard. OpenPC has the ability to translate or “map” keyboard input into scan codes.

System Administrator

The person who monitors and controls the computer on which your UNIX system operates.

Virtual Memory

A data storage method that, through an elaborate software and hardware memory addressing scheme, allows a computer (but not a PC) to act as if it has more memory available on board than is physically present.

Virtual memory schemes allow an application such as OpenPC to use as much memory as required without concern for other processes that also may be resident in memory at the same time.

Virtual memory implementation is usually accompanied by the operating system writing processes out to disk. This is called swapping. If too many processes are running concurrently, swapping activity increases, resulting in a noticeable degradation of performance in individual applications.

Virtual PC

An IBM PC simulated by software. In the context of the OpenPC product, virtual PC actually means virtual AT-compatible PC because OpenPC runs AT-compatible software in a protected multi-tasking environment.

The name given to the ability within OpenPC to have more than one task assigned to the screen at the same time.

Well-Behaved

Application programs that correctly use the BIOS to access hardware.

Index

Symbols

\$HOME redirected drive, 4-13

~ (tilde), 4-18

A

access, to virtual diskette, 4-15

access permissions, via Status, 5-3

applications

DOS, 5-1

installing, 5-2

network licensing, 5-2

non-DOS, 1-1

where to install, 5-1

Assign Diskette, 4-7

for ownership among all users,
5-2

AUTOEXEC.BAT, 1-10, 2-5,
4-13

defaults, 3-11

entering REDIR, 6-2

B

backslash character in
DOSMOUNT, 4-13

batch file

disabling idle detection to
process, 1-13

example, 4-12

using opencmd in, 4-10

C

CGA, 1-6

chmod, 4-16

CMOS RAM

emulation, 1-13

image file, 3-5

color video, 1-6

command line interface, 1-12

communications, serial, 1-7

CONFIG.SYS, 1-10

defaults, 3-11

enabling Redirector, 6-2

mouse driver in, 4-21

configuration file, openpc.dat, 3-4

configuration file, openpc.dat,
1-11

coprocessor, floating point, 80287,
1-2

cursor, for X Windows mouse,
4-21

C, cont'd

Customize menu, 4-7

cut/paste, 1-12

D

data

privacy, 3-2, 4-16

where to store, 5-1

Device status... information popup,
4-5

dg_pcterm, 7-2

naming terminal specification
file, 7-3

dialog box for assigning diskette or
drive, 4-9

diskette

assigning ownership, 5-2

regaining access, 4-15

diskman, 2-4

DISPLAY, 2-3

DOS

applications, 5-1

file names, 4-16

start-up files, 1-10

dosbin, 2-4

DOSMOUNT, 2-5, 4-13,

6-1, 6-3

examples, 6-5

DOSMOUNT.EXE, 4-12

drive, redirected, 4-14

drives, mapping A:-Z:, 4-13

E

Edit menu, 4-7

emulation

CMOS RAM, 1-13

hard disks, 1-7

IBM PC AT, 1-6

ROM BIOS, 1-13

video, CGA, MDA, Hercules,
1-2

environment variable

HOME, 1-11

OPENPC, 1-11

shell, 3-10

TERM, 7-2

Expanded Memory Specification,
3-12

F

F-10 key setting, 5-3

file name mapping, example, 4-18

file permissions, setting, 4-16

file sharing, 1-2

filter, 4-20

FIRST_HARD_DISK, required
keyword, 3-6

fonts

normal, 3-5

wide, 3-5

G

getty, 3-5
 graphical interface, 1-7

H

Hercules adapter, 1-6
 home directory, 4-16
 HOME environment variable, 1-11

I

idle detection, 1-13, 4-12
 information popup box under
 Device status, 4-5
 INSTALL utility, 5-2

K

keyboard, 1-7
 keyboard mapping, 1-14
 keywords in openpc.dat, 3-6

L

licensing, multiusers, 1-9
 LIM EMS, 3-12
 Lotus 1-2-3, 3-12, 4-12
 lowercase characters, in DG/UX
 file names, 4-17
 LPT, 4-20

M

memory
 expanded, 1-7
 limits, 1-7
 Menu Bar, 4-3
 modem, 1-7, 3-9
 monochrome display, 1-6
 Motif Window Manager, 1-12, 4-1
 mount point
 DOSMOUNT, 6-4
 NFS mounted directories, 4-14
 mouse
 cursor disappears, 4-21, 4-22
 default, 4-21
 licensing, 4-21
 Microsoft Bus, 1-7
 OpenPC emulation of, 4-21
 PC, 4-8
 re-attaching, 4-21
 serving OpenPC, 4-21
 serving X Windows, 4-21
 using, 4-21
 X, 4-8
 MS-DOS, 1-1
 multiple sessions, 4-15
 mwm, 4-2
 myconfig, 3-10

N

network, 1-14
 networked drive, 6-1
 NFS, 4-14
 NFS mount, 4-14

O

opencmd, 1-12, 4-10
 example, 4-12
 on serial terminals, 7-2

OPENPC, 3-10

OPENPC environment variable,
 1-11

OpenPC start-up files, 1-11

openpc.dat, 1-9, 1-11, 2-5, 3-2,
 4-16
 keywords, 3-6
 mapping drive A:, 5-2
 printer specifications, 4-20
 sample file, 3-4
 settings, 3-6

OS/2, 1-6

OSF/Motif, 4-1

out of memory error message,
 3-12

P

PC personality mode, 7-1

PC-DOS, 1-1

PEMM.EXE, 3-12

Phoenix, EMS driver, 3-12

physical diskette, emulating
 inserting, 4-12

physical drive, 4-9

pipe, to print spooler, 4-20

Print menu, 4-9

printer buffer, 4-10
 flushed, 4-10

printer, parallel, 1-7

printing, 4-20

privacy
 data, 4-16
 via home directory, 5-1
 via Redirector, 5-1

R

Read/Write permissions, 4-15

REDIR, 6-1

REDIR.EXE, 4-12, 6-2

redirected disks, 1-9, 1-10
 assigning via DOSMOUNT, 6-3
 DG/UX access to, 1-9

redirected drives
 enabling, 6-2
 file name mapping, 4-16

redirection, file system, 1-7

Redirector
 initializing, 6-2
 utilities, 4-12

ROM BIOS
 emulation, 1-13
 image, 3-5

run level 3, when installing, 2-1

S

serial communications, 1-7

serial terminal, supported, 2-4

S, cont'd

serial terminal mode, 4-2

serial terminals, 1-12, 7-1
 D412+, D216+, D462+, 7-1
 limitations on, 7-2
 opencmd, 4-10, 7-2
 opencmd on, 4-10
 personality modes, 7-3
 restoring serial mode, 7-3
 setup, 7-2
 troubleshooting, 7-3

Session menu, 4-4

slash character in DOSMOUNT,
 4-13

Status, displaying access
 permissions, 5-3

submenus, 4-4
 Reboot, 4-4

T

tape, 2-1

TERM environment variable,
 4-2, 7-2
 incorrectly set, 7-3

text mode, 1-12

toggle button, 4-4

U

uppercase filenames, 4-16

user interface, choosing, 4-1

V

video
 color, 1-6
 graphics, 1-6
 monochrome, 1-6

virtual, devices, 1-6

virtual diskettes
 assigning, 4-9
 defined, 1-9

virtual disks
 accessing, 4-15
 capacity, 1-7
 defined, 1-8
 DOS start-up files on, 3-11
 private, 4-16

virtual IBM PC AT, 1-1

W

window size, changing, 1-12, 4-3

workstation, diskless, 2-2

X

X terminals, 1-12

X Window, client, 2-3

X Windows, 2-2

X11, 4-1

xfd program, 3-5

xlsfonts, 3-7

xterm, 4-2

TIPS ORDERING PROCEDURES

TO ORDER

1. An order can be placed with the TIPS group in two ways:
 - a) MAIL ORDER – Use the order form on the opposite page and fill in all requested information. Be sure to include shipping charges and local sales tax. If applicable, write in your tax exempt number in the space provided on the order form.

Send your order form with payment to: Data General Corporation
 ATTN: Educational Services/TIPS G155
 4400 Computer Drive
 Westboro, MA 01581-9973

- b) TELEPHONE – Call TIPS at (508) 870-1600 for all orders that will be charged by credit card or paid for by purchase orders over \$50.00. Operators are available from 8:30 AM to 5:00 PM EST.

METHOD OF PAYMENT

2. As a customer, you have several payment options:
 - a) Purchase Order – Minimum of \$50. If ordering by mail, a hard copy of the purchase order must accompany order.
 - b) Check or Money Order – Make payable to Data General Corporation.
 - c) Credit Card – A minimum order of \$20 is required for Mastercard or Visa orders.

SHIPPING

3. To determine the charge for UPS shipping and handling, check the total quantity of units in your order and refer to the following chart:

Total Quantity	Shipping & Handling Charge
1-4 Units	\$5.00
5-10 Units	\$8.00
11-40 Units	\$10.00
41-200 Units	\$30.00
Over 200 Units	\$100.00

If overnight or second day shipment is desired, this information should be indicated on the order form. A separate charge will be determined at time of shipment and added to your bill.

VOLUME DISCOUNTS

4. The TIPS discount schedule is based upon the total value of the order.

Order Amount	Discount
\$1-\$149.99	0%
\$150-\$499.99	10%
Over \$500	20%

TERMS AND CONDITIONS

5. Read the TIPS terms and conditions on the reverse side of the order form carefully. These must be adhered to at all times.

DELIVERY

6. Allow at least two weeks for delivery.

RETURNS

Items ordered through the TIPS catalog may not be returned for credit.

7. Order discrepancies must be reported within 15 days of shipment date. Contact your TIPS Administrator at (508) 870-1600 to notify the TIPS department of any problems.

INTERNATIONAL ORDERS

9. Customers outside of the United States must obtain documentation from their local Data General Subsidiary or Representative. Any TIPS orders received by Data General U.S. Headquarters will be forwarded to the appropriate DG Subsidiary or Representative for processing.

DATA GENERAL CORPORATION TECHNICAL INFORMATION AND PUBLICATIONS SERVICE TERMS AND CONDITIONS

Data General Corporation ("DGC") provides its Technical Information and Publications Service (TIPS) solely in accordance with the following terms and conditions and more specifically to the Customer signing the Educational Services TIPS Order Form. These terms and conditions apply to all orders, telephone, telex, or mail. By accepting these products the Customer accepts and agrees to be bound by these terms and conditions.

1. CUSTOMER CERTIFICATION

Customer hereby certifies that it is the owner or lessee of the DGC equipment and/or licensee/sub-licensee of the software which is the subject matter of the publication(s) ordered hereunder.

2. TAXES

Customer shall be responsible for all taxes, including taxes paid or payable by DGC for products or services supplied under this Agreement, exclusive of taxes based on DGC's net income, unless Customer provides written proof of exemption.

3. DATA AND PROPRIETARY RIGHTS

Portions of the publications and materials supplied under this Agreement are proprietary and will be so marked. Customer shall abide by such markings. DGC retains for itself exclusively all proprietary rights (including manufacturing rights) in and to all designs, engineering details and other data pertaining to the products described in such publication. Licensed software materials are provided pursuant to the terms and conditions of the Program License Agreement (PLA) between the Customer and DGC and such PLA is made a part of and incorporated into this Agreement by reference. A copyright notice on any data by itself does not constitute or evidence a publication or public disclosure.

4. LIMITED MEDIA WARRANTY

DGC warrants the CLI Macros media, provided by DGC to the Customer under this Agreement, against physical defects for a period of ninety (90) days from the date of shipment by DGC. DGC will replace defective media at no charge to you, provided it is returned postage prepaid to DGC within the ninety (90) day warranty period. This shall be your exclusive remedy and DGC's sole obligation and liability for defective media. This limited media warranty does not apply if the media has been damaged by accident, abuse or misuse.

5. DISCLAIMER OF WARRANTY

EXCEPT FOR THE LIMITED MEDIA WARRANTY NOTED ABOVE, DGC MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE ON ANY OF THE PUBLICATIONS, CLI MACROS OR MATERIALS SUPPLIED HEREUNDER.

6. LIMITATION OF LIABILITY

A. CUSTOMER AGREES THAT DGC'S LIABILITY, IF ANY, FOR DAMAGES, INCLUDING BUT NOT LIMITED TO LIABILITY ARISING OUT OF CONTRACT, NEGLIGENCE, STRICT LIABILITY IN TORT OR WARRANTY SHALL NOT EXCEED THE CHARGES PAID BY CUSTOMER FOR THE PARTICULAR PUBLICATION OR CLI MACRO INVOLVED. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO CLAIMS FOR PERSONAL INJURY CAUSED SOLELY BY DGC'S NEGLIGENCE. OTHER THAN THE CHARGES REFERENCED HEREIN, IN NO EVENT SHALL DGC BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES WHATSOEVER, INCLUDING BUT NOT LIMITED TO LOST PROFITS AND DAMAGES RESULTING FROM LOSS OF USE, OR LOST DATA, OR DELIVERY DELAYS, EVEN IF DGC HAS BEEN ADVISED, KNEW OR SHOULD HAVE KNOWN OF THE POSSIBILITY THEREOF; OR FOR ANY CLAIM BY ANY THIRD PARTY.

B. ANY ACTION AGAINST DGC MUST BE COMMENCED WITHIN ONE (1) YEAR AFTER THE CAUSE OF ACTION ACCRUES.

7. GENERAL

A valid contract binding upon DGC will come into being only at the time of DGC's acceptance of the referenced Educational Services Order Form. Such contract is governed by the laws of the Commonwealth of Massachusetts, excluding its conflict of law rules. Such contract is not assignable. These terms and conditions constitute the entire agreement between the parties with respect to the subject matter hereof and supersedes all prior oral or written communications, agreements and understandings. These terms and conditions shall prevail notwithstanding any different, conflicting or additional terms and conditions which may appear on any order submitted by Customer. DGC hereby rejects all such different, conflicting, or additional terms.

8. IMPORTANT NOTICE REGARDING AOS/VS INTERNALS SERIES (ORDER #1865 & #1875)

Customer understands that information and material presented in the AOS/VS Internals Series documents may be specific to a particular revision of the product. Consequently user programs or systems based on this information and material may be revision-locked and may not function properly with prior or future revisions of the product. Therefore, Data General makes no representations as to the utility of this information and material beyond the current revision level which is the subject of the manual. Any use thereof by you or your company is at your own risk. Data General disclaims any liability arising from any such use and I and my company (Customer) hold Data General completely harmless therefrom.

CUSTOMER DOCUMENTATION COMMENT FORM

Your Name _____ Your Title _____

Company _____ Phone _____

Street _____

City _____ State _____ Zip _____

We wrote this book for you, and we made certain assumptions about who you are and how you would use it. Your comments will help us correct our assumptions and improve the manual. Please take a few minutes to respond. Thank you.

Manual Title _____ Manual No. _____

Who are you? EDP/MIS Manager Analyst/Programmer Other _____
 Senior Systems Analyst Operator
 Engineer End User

How do you use this manual? (*List in order: 1 = Primary Use*)

___ Introduction to the product ___ Tutorial Text ___ Other
 ___ Reference ___ Operating Guide _____

About the manual:		Yes	No
Is it easy to read?		<input type="checkbox"/>	<input type="checkbox"/>
Is it easy to understand?		<input type="checkbox"/>	<input type="checkbox"/>
Are the topics logically organized?		<input type="checkbox"/>	<input type="checkbox"/>
Is the technical information accurate?		<input type="checkbox"/>	<input type="checkbox"/>
Can you easily find what you want?		<input type="checkbox"/>	<input type="checkbox"/>
Does it tell you everything you need to know?		<input type="checkbox"/>	<input type="checkbox"/>
Do the illustrations help you?		<input type="checkbox"/>	<input type="checkbox"/>

If you wish to order manuals, contact your sales representative or dealer.

Comments:

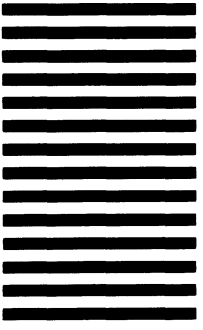
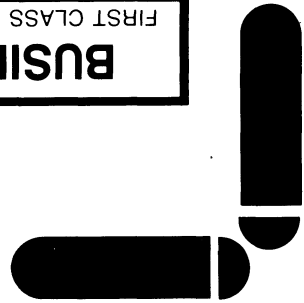
134-1/02-02

Customer Documentation
MS E-111
4400 Computer Drive
P.O. Box 4400
Westboro, MA 01581-9890

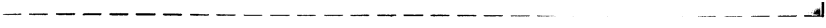


POSTAGE WILL BE PAID BY ADDRESSEE

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 26 WESTBORO, MA 01581



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



Cut here and insert in binder spine pocket

