

VP/ix* Environment
Guide

INTERACTIVE

product family

INTERACTIVE
.....
A Kodak Company

First printing (March 1990)

No part of this manual may be reproduced in any form or by any means without written permission of:

INTERACTIVE Systems Corporation
2401 Colorado Avenue
Santa Monica, California 90404

© Copyright INTERACTIVE Systems Corporation 1987-1990

© Copyright Phoenix Technologies Ltd. and INTERACTIVE Systems Corporation 1984, 1985, 1986, 1987, 1988

© Copyright AT&T Corporation 1987

© Copyright Microsoft Corporation 1981, 1982, 1983, 1984, 1985, 1986, 1987

RESTRICTED RIGHTS:

For non-U.S. Government use:

These programs are supplied under a license. They may be used, disclosed, and/or copied only as permitted under such license agreement. Any copy must contain the above copyright notice and this restricted rights notice. Use, copying, and/or disclosure of the programs is strictly prohibited unless otherwise provided in the license agreement.

For U.S. Government use:

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR Section 52.227-14 (Alternate III) or subparagraph (c)(1)(ii) of the clause at DFAR 252.227-7013, Rights in Technical Data and Computer Software.

All rights reserved. Printed in the U.S.A.

The *termcap* code and documentation were developed at the Electrical Engineering and Computer Sciences Department at the Berkeley Campus of the University of California under the auspices of the Regents of the University of California.

The following trademarks shown as registered are registered in the United States and other countries:

VP/ix is a trademark of INTERACTIVE Systems Corporation and Phoenix Technologies Ltd.

UNIX is a registered trademark of AT&T.

dBASE is a trademark of Ashton-Tate.

Paradox is a registered trademark of Borland International, Inc.

DEC, VT52, and VT100 are trademarks of Digital Equipment Corporation.

Fastback is a trademark of Fifth Generation Systems, Inc.

Hercules is a trademark of Hercules Computer Technology.

Intel is a registered trademark of Intel Corporation.

386, 80386, and MULTIBUS are trademarks of Intel Corporation.

AT, IBM, and IBM PC are registered trademarks of International Business Machines Corporation.

KT-70/PC is a registered trademark of Kimtron Corporation.

LOGITECH is a trademark of LOGITECH, Inc.

Lotus and 1-2-3 are registered trademarks of Lotus Development Corporation.

WordStar is a trademark of MicroPro International, Inc.

Microsoft and MS-DOS are registered trademarks of Microsoft Corporation.

MS Windows and WORD are trademarks of Microsoft Corporation.

CrossTalk is a trademark of MicroStuf, Inc.

Ptel is a trademark of Phoenix Technologies Ltd.

WordPerfect is a registered trademark of WordPerfect Corporation.

Wyse and Wyse 60 are registered trademarks of Wyse Technology.

Programs described in this manual are copyrighted and their copyright notices may be found in heralds, by using the UNIX System *what* program, and by reading files whose names start with "coprisc".

VP/ix Environment Guide

CONTENTS

Introduction to the VP/ix Environment

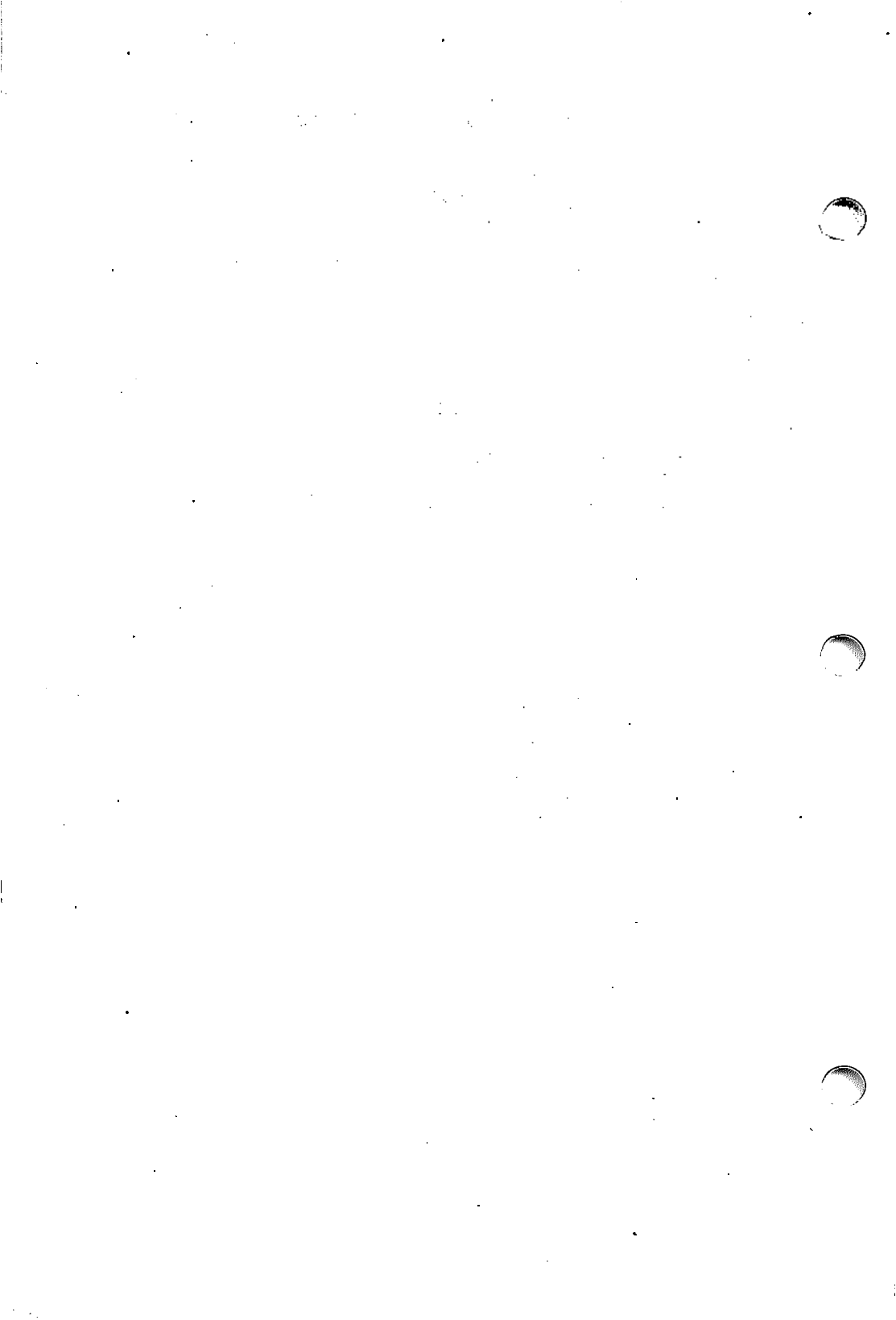
VP/ix Environment Release Notes

VP/ix and MS-DOS Primer

VP/ix Environment Installation Instructions

VP/ix Environment Maintenance Procedures

VP/ix Environment Manual Entries



Introduction to the VP/ix* Environment

Welcome to the *VP/ix Environment Guide*. This guide contains the information you need to install, maintain, and use the VP/ix Environment, Release 1.2. Whether you are an experienced VP/ix user or brand new to the VP/ix Environment, be sure to read the next few pages of this document. They will tell you what is contained in this guide and how to use the guide to your best advantage.

WHAT'S INCLUDED

The *VP/ix Environment Guide* includes:

- **VP/ix Environment Release Notes**
Describes the enhancements made to Release 1.2 of the VP/ix Environment and outlines important installation instructions for updating your system.
- **VP/ix and MS-DOS Primer**
Provides an introduction to the basic capabilities and commands available with VP/ix and MS-DOS* (DOS). This primer is a step-by-step guide to using DOS and the UNIX* System in the VP/ix Environment.
- **VP/ix Environment Installation Instructions**
Provides step-by-step instructions on how to install and configure the VP/ix Environment.
- **VP/ix Environment Maintenance Procedures**
Provides a detailed description of the basic procedures required to keep the VP/ix Environment running smoothly.
- **VP/ix Environment Manual Entries**
INTERACTIVE's proprietary manual entries, which supplement those in the *INTERACTIVE UNIX System V/386 Release 3.2 User's/System Administrator's Reference Manual*.
- **Reader's Comment Form**
Provides you with a way to tell us what you like or dislike about this guide and to send us your ideas for making it even better.

WHERE TO BEGIN

The *VP/ix Environment Guide* includes a variety of documents for users at varying levels of experience. Depending on your level of experience, you may want to use this guide in a number of different ways. The outline below provides some suggested ways to use this guide:

- **If you are a beginner . . .**
Read the “VP/ix and MS-DOS Primer” to learn how VP/ix works in the DOS environment.
- **If you are an experienced DOS user . . .**
Read the following sections in the “VP/ix and MS-DOS Primer”: “GETTING STARTED WITH VP/ix,” “DIFFERENCES BETWEEN DOS AND UNIX SYSTEM FILE SYSTEMS,” and “USING DOS AND UNIX SYSTEM COMMANDS TOGETHER.” Refer to Appendix B of the “VP/ix and MS-DOS Primer” for important information about the differences between running native DOS and DOS under VP/ix.
- **If you are installing the system . . .**
Read and follow the steps outlined in the “VP/ix Environment Installation Instructions.” Once you have completed the basic system installation, go to the “VP/ix Environment Maintenance Procedures” for information about how to perform system maintenance tasks and tailor the system to match your requirements.
- **If you want the latest system information . . .**
Read the “VP/ix Environment Release Notes.” These notes provide you with up-to-the-minute information on what’s new in Release 1.2 of the VP/ix Environment.
- **If you want supplemental documentation . . .**
Refer to the *MS-DOS Reference Guide* for detailed descriptions of DOS commands, program interface information, and other technical details. Refer to the *VP/ix Technical Guide* for a detailed, technical description of the internals of the VP/ix Environment. These documents are intended for experienced DOS and VP/ix users. For a complete listing of all documentation that relates to INTERACTIVE UNIX System V/386 Release 3.2, refer to the “Documentation Roadmap” included in the *INTERACTIVE UNIX System V/386 Release 3.2 Operating System Guide*.

OVERVIEW OF THE VP/ix ENVIRONMENT

The VP/ix Environment is an enhancement to the UNIX Operating System that allows DOS and the UNIX System, as well as their applications, to execute on the same computer. The VP/ix Environment provides the DOS user with a multi-tasking, multi-user operating environment. For the UNIX System user, the VP/ix Environment provides the DOS interface, which supports thousands of “off-the-shelf” applications. In addition, the VP/ix Environment bridges the gap between the UNIX System and DOS by enabling users to:

- Run UNIX System commands from DOS.
- Run DOS commands from the UNIX System.
- Access UNIX System files from DOS.
- Run DOS commands from remote terminals.
- Perform many other functions impossible with a DOS or UNIX System alone.

The term “DOS” refers to MS-DOS version 3.3 running in the VP/ix Environment. The term “native DOS” refers to MS-DOS running as the native operating system on a PC.

HOW DOES THE VP/ix ENVIRONMENT WORK?

In past years, hardware and software vendors attempted to develop systems that would capitalize on the key advantages of both DOS and the UNIX System. However, the development of a truly integrated DOS/UNIX environment was limited by the available hardware technology. Most solutions required that a computer run either DOS or UNIX – the two systems could not run simultaneously. In addition, data used by the two systems was stored in two distinct areas on the fixed disk of the computer.

The introduction of the 80386* microprocessor from Intel* has finally made the integration of the UNIX System and DOS possible. The architecture of the 80386 chip is a technological advancement that permits the execution of DOS and DOS applications under the UNIX System.

In the VP/ix Environment, the UNIX Operating System acts as the primary (native) operating system, providing the multi-tasking, multi-user foundation for the computer. To access the VP/ix

Environment, the user establishes a UNIX System session by logging into the computer and starting DOS with the `vpix` command. Once the VP/ix Environment is active, DOS programs and commands can be run just as if DOS were the native operating system on your computer.

The combination of the INTERACTIVE UNIX Operating System and the VP/ix Environment provides a powerful, fully integrated solution for users who want to retain their investment in DOS applications while acquiring the versatility of the UNIX System. The INTERACTIVE UNIX Operating System gives you the power of the UNIX System, and the VP/ix Environment gives you the convenience of DOS.

WHO CAN USE THE VP/ix ENVIRONMENT?

The VP/ix Environment can be used by executives, writers, secretaries, clerks, programmers, and accountants – anyone who uses a computer. Users can purchase off-the-shelf applications designed to run on either the UNIX System or DOS. Because the VP/ix Environment, together with the UNIX System, provides two operating system interfaces, users select the interface with which they are most comfortable.

For application builders, VP/ix offers a powerful development environment. Developers can use both DOS and UNIX System utilities, commands, and program development tools to create new applications or customize existing applications quickly and easily.

VP/ix ENVIRONMENT FEATURES

With the VP/ix Environment, multiple DOS and UNIX System applications can be run simultaneously. Other features include a file system that fully integrates DOS and the UNIX System and the ability to run UNIX System commands under DOS and DOS commands under the UNIX System. Major features of this environment are described below.

A Multi-User, Multi-Tasking DOS Environment

The VP/ix Environment is a fully integrated combination of MS-DOS and the UNIX Operating System. With VP/ix, multiple users can simultaneously run DOS applications on the same system. The UNIX System kernel is extended to provide the same efficient, interrupt-driven scheduling services to both DOS and UNIX System applications. Under VP/ix, multiple DOS applications can run concurrently with conventional UNIX System processes in a paged, virtual memory environment. Each application runs in its own secure virtual address space.

Support for DOS Applications

VP/ix supports popular PC applications and DOS facilities that run on IBM* PC AT*-compatible systems. It includes support for direct hardware accesses to standard devices required by ill-behaved programs. VP/ix also provides support for up to 2 MB of virtual EMS (Expanded Memory Specification) for DOS programs, regardless of a system's actual physical memory configuration.

Integrated DOS File System

VP/ix provides a transparent integration of the DOS and UNIX System file systems. Any program, regardless of whether it is a DOS or a UNIX System program, can access and share the same files.

Convenient User Interface

Users can run DOS applications under VP/ix in one of two ways: by explicitly invoking the MS-DOS under VP/ix Environment or by running the application directly from the UNIX System prompt. As

a result, users can run DOS applications from the environment in which they are most comfortable: the UNIX System or DOS. VP/ix also provides an optional pop-up menu interface, complete with help screens.

Virtual Terminal Integration

The VP/ix Environment is fully integrated with the INTERACTIVE UNIX Virtual Terminal Manager. This enables a user to run multiple full-screen DOS and UNIX System applications concurrently on the system console or other terminals with direct-write screens. The INTERACTIVE UNIX Virtual Terminal Manager allows a user to switch the physical screen from one virtual screen to another with a “hot-key” sequence.

Transparent Execution of DOS and UNIX System Commands

The VP/ix Environment allows users to run DOS programs directly from the UNIX System and UNIX System programs directly from DOS. VP/ix allows users to combine DOS and UNIX System commands with pipes and I/O redirection on a command line.

Customized Version of DOS

An enhanced version of MS-DOS is licensed and packaged with the VP/ix Environment to improve system performance.

Graphic Display Support

On the system console, DOS graphics are fully supported on the EGA, CGA, and Hercules* graphics adapters. Multiple concurrent graphics applications are supported via the INTERACTIVE UNIX Virtual Terminal Manager.

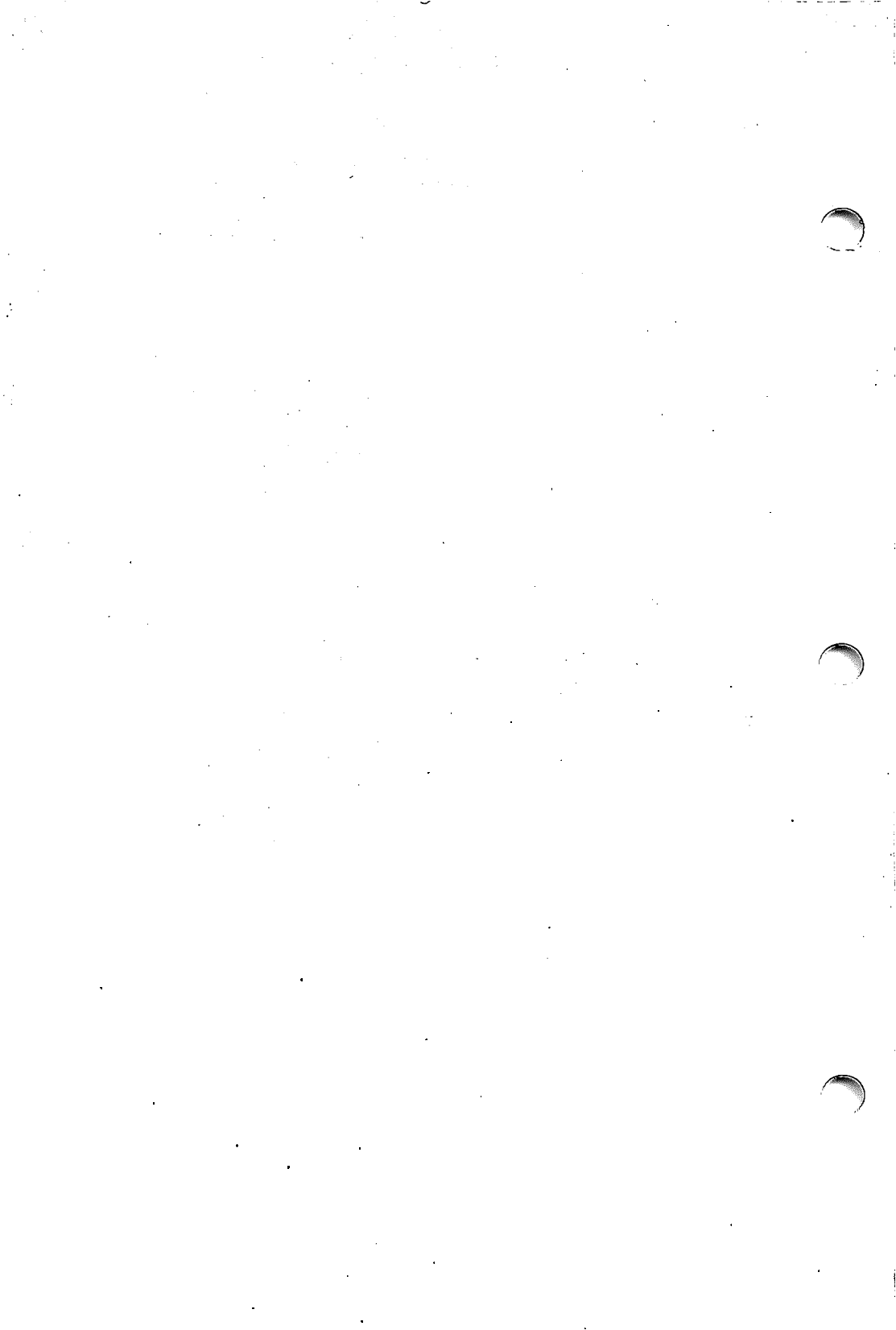
DOCUMENTATION REFERENCES

Throughout this guide, the following documentation titles will be referenced in shortened versions as follows:

<i>Full Title</i>	<i>Shortened Version</i>
INTERACTIVE UNIX System V/386 Release 3.2 Operating System Guide	INTERACTIVE UNIX Operating System Guide
INTERACTIVE UNIX System V/386 Release 3.2 Guide for New Users	INTERACTIVE UNIX System Guide for New Users
INTERACTIVE Software Development System Guide and Programmer's Reference Manual	INTERACTIVE SDS Guide and Programmer's Reference Manual

FOR MORE INFORMATION

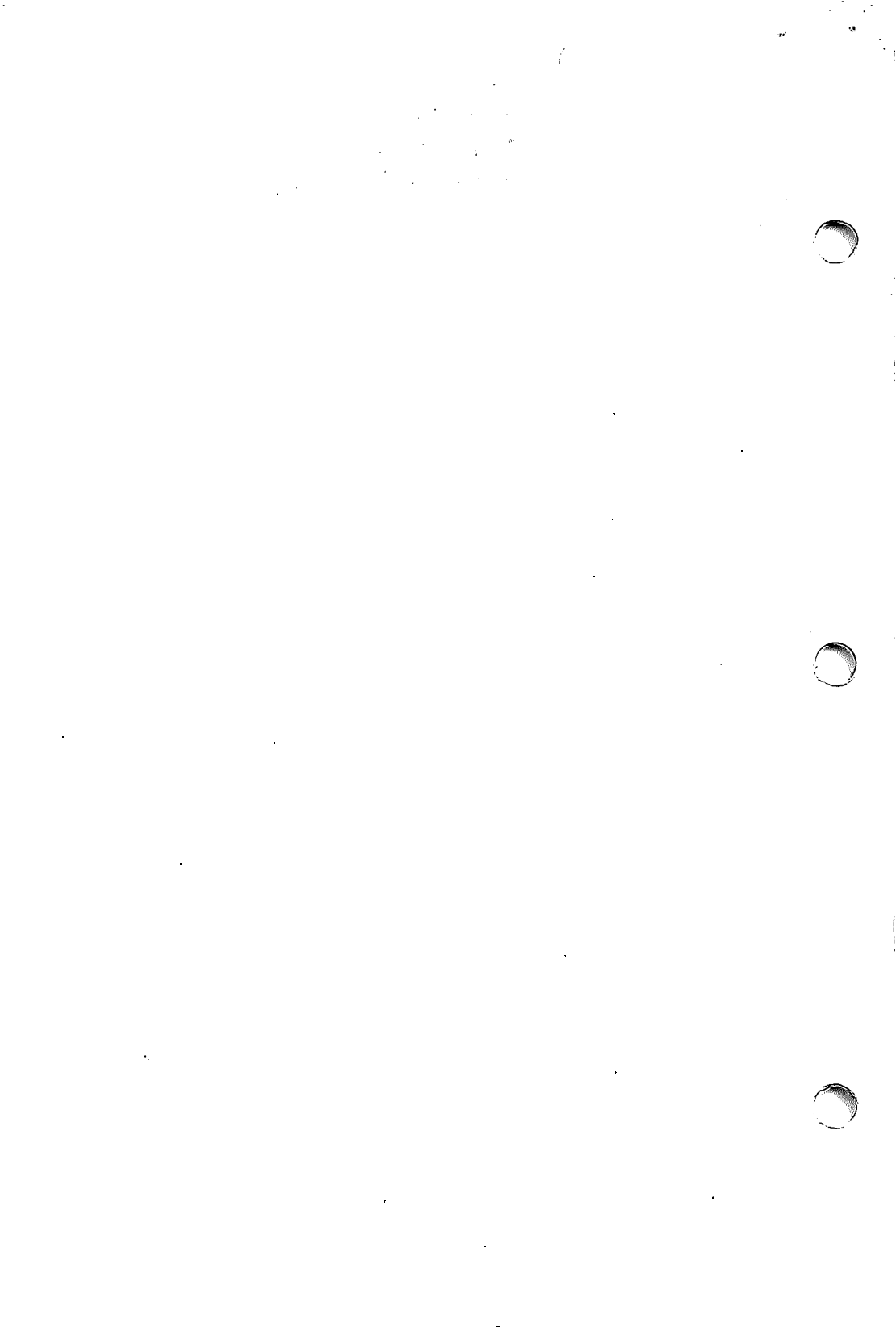
The documentation included in this guide provides information about how to install, use, and maintain the VP/ix Environment and supplements the information found in the *MS-DOS Reference Guide* and the *VP/ix Technical Guide*. For a complete listing of all INTERACTIVE UNIX System related documentation, refer to the "Documentation Roadmap" included in the *INTERACTIVE UNIX Operating System Guide*.



VP/ix Environment
Release 1.2.1
Release Notes

CONTENTS

1. INTRODUCTION	1
2. ENHANCEMENTS ADDED IN RELEASE 1.2	1
2.1 LIM 4.0 EMS	1
2.2 The AUTOFLUSH Mechanism	2
2.3 Personal autoexec.bat Files	2
3. PROBLEM RESOLUTIONS	3
3.1 Keyboard Issues	3
3.2 Serial Terminal Issues	3
3.3 VP/ix Image Issues	3
3.4 Execution with Microsoft Windows 3.0	4
3.5 Shared Libraries	4
4. INSTALLATION	5
4.1 Before You Install the VP/ix Environment	5
4.2 New Distribution of Subsets	5
4.3 Updating the VP/ix Environment	5



VP/ix* Environment

Release 1.2.1

Release Notes

June 1991

1. INTRODUCTION

This document describes the changes made in Release 1.2.1 of the VP/ix Environment and outlines important installation instructions for updating your system.

- ▣ Read this entire document before attempting to update your VP/ix Environment software.

2. ENHANCEMENTS ADDED IN RELEASE 1.2

A number of enhancements were added in the previous release.

2.1 LIM 4.0 EMS

VP/ix now supports the Lotus*-Intel*-Microsoft* (LIM) Enhanced Memory Specification 4.0.

The new LIM 4.0 driver is backward compatible with the old LIM 3.2 driver. The MS-DOS* (DOS) driver is named PEMM.SYS. It uses the same command line syntax as the previous 3.2 driver (EMS.SYS). The driver must be placed in the CONFIG.SYS file on the C: drive if it is to be used. You must have the INTERACTIVE UNIX* Operating System Version 2.2 or later installed on your system in order to use the PEMM.SYS driver. If you are using a release earlier than Version 2.2, you can use only EMS.SYS. You may use either EMS.SYS or PEMM.SYS if you are using INTERACTIVE UNIX Operating System Version 2.2 or later. When using any LIM driver on INTERACTIVE UNIX Operating System Version 2.2 or later, the system performance will be significantly degraded. It is recommended that you use only 32 pages of LIM when using

EMS.SYS and 48 pages of LIM when using PEMM.SYS. The LIM performance issue will be addressed in a future release of VP/ix. See section 5.2 of the “VP/ix Environment Maintenance Procedures” for more information.

2.2 The AUTOFLUSH Mechanism

VP/ix now has automatic printer buffer flushing.

This is an option you can choose to configure into your VP/ix session. It will automatically flush the data from the print buffer whenever a printer does not receive anything to print for a specified period of time. Your VP/ix configuration file contains the AUTOFLUSH entry. To enable it, remove the semicolon. This will cause the printer to AUTOFLUSH every 20 seconds. The number after the AUTOFLUSH entry in the `vpix.cnf` file determines the number of seconds of idle time before a printer is flushed. If you set this value to 0, the autoflush mechanism will be disabled and your printer will not be automatically flushed. Otherwise, the minimum accepted value is 5 seconds. Values set to less than this will default to 5. There are no warning messages printed on startup if the number can not be read correctly; the autoflush mechanism is just disabled. If the AUTOFLUSH entry is commented out, the autoflush mechanism will be disabled.

When using `PRT-SCR` with AUTOFLUSH enabled, the printer buffer will be flushed automatically after the `PRT-SCR` is done.

The VP/ix User Interface Menu still contains the (P)rinter flush entry. Using this option with AUTOFLUSH enabled has no adverse effects on printing.

2.3 Personal `autoexec.bat` Files

A feature was added that allows individual users to have their own `autoexec.bat` files. Users can create a DOS text file containing DOS batch file commands. The file must reside in the user's home directory and must have the UNIX System file name `autoexec.bat`. When DOS boots, it will execute the default `autoexec.bat` file located on the C: drive, followed by the user's `autoexec.bat` file if it exists. This change allows a user to create a customized copy of the `autoexec.bat` file without the need to maintain a personal copy of the C: drive as well.

3. PROBLEM RESOLUTIONS

A number of problems have been resolved for this release. The most important of these are discussed in the following subsections.

3.1 Keyboard Issues

When the VP/ix session on a serial terminal was rebooted using the **CTRL ALT DEL** key sequence (a “warm” reboot), the **CTRL** and **ALT** keys were still active at the prompt after the warm reboot. This problem caused the system to not take any input at the prompt since most keys were interpreted to be control sequences. The states of these keys are now cleared.

The state of the **NUM-LOCK** key is saved before VP/ix is entered and restored upon exit from VP/ix. Previously, the **NUM-LOCK** status was not restored upon exit, causing problems for some users.

3.2 Serial Terminal Issues

On serial terminals, VP/ix used to send the `clear` escape sequence after the `fvpix` escape sequence. On some terminals, the `clear` escape sequence is different between PC-compatible mode and other modes, and the screen was not cleared correctly. To eliminate the problem, VP/ix now sends the `clear` escape sequence before the `fvpix` escape sequence.

3.3 VP/ix Image Issues

If a user invoked the DOS `format` command when executing from a VP/ix image, the `format` command failed. VP/ix has been fixed to `format` correctly while executing from a stored image.

Important Note: Earlier versions of VP/ix did not set the `date` correctly to the current date when executing from an image. This release contains a new version of the program `rundos.com` that sets the `date` correctly on start up when executing from an image file. All users who use private C: drive pseudo-volumes should copy the file `/usr/vpix/dosbin/rundos.com` to their C: drive root directories. The command to do this from within VP/ix is:

```
copy z:\usr\vpix\dosbin\rundos.com c:\
```

3.4 Execution with Microsoft Windows 3.0

VP/ix has been enhanced to support execution of Microsoft Windows* 3.0 in real mode. The previously released fix disk that addressed this issue is no longer necessary with this release. You can run Windows 3.0 only in real mode, which is invoked using:

```
win /r
```

Using `win` without `/r` will cause you to run in protected mode.

If you install Windows 3.0 using the setup program supplied with the product, it may add the line `device=C:\himem.sys` to your `C:\config.sys` file. This will cause the following message to appear:

```
Installed A20 handler number 1
Error: Can't enable A20!
      XMS Driver not installed
```

This message can be safely ignored; however, to prevent it from occurring, remove the line `device=C:\himem.sys` from your `C:\config.sys` file.

3.5 Shared Libraries

When creating a `vpix` binary, if the user used VP/ix Installable Emulation Module (IEM) modules that require shared libraries, such as the shared TLI library, the linking process produced errors. The new link procedure solves this problem and allows users to utilize shared libraries from VP/ix IEM modules when creating a `vpix` binary.

4. INSTALLATION

4.1 Before You Install the VP/ix Environment

The installation instructions in these release notes apply only to systems on which a previous release of the VP/ix Environment is currently installed. If you do not currently have the VP/ix Environment installed on your system, you can follow the procedures found in the “VP/ix Environment Installation Instructions.”

If you *do* have the VP/ix Environment currently installed on your system and you have changed the `/usr/vpix/etc/vpixdevs` or the `/usr/vpix/defaults/vpix.cnf` file from its original configuration, save the file or files temporarily under another name before you follow these instructions. If you do not do so, your customized file(s) will be overwritten. Note that the `vpixdevs` file *has* been changed if you added any VP/ix IEMs to your original VP/ix installation.

4.2 New Distribution of Subsets

This release of VP/ix is distributed as three separate subsets, *Core*, *Configuration*, and *DOS*. The Core subset contains the VP/ix binaries. The Configuration subset allows you to reconfigure the VP/ix Environment. The Core and Configuration subsets are distributed on one diskette, labeled *Core*, and the DOS subset is distributed on the second diskette, labeled *MS-DOS*.

If you received the VP/ix Release 1.2.1 update package, you were given only the *Core* diskette since the DOS subset did not change.

4.3 Updating the VP/ix Environment

To update your VP/ix Environment software, follow this procedure:

1. If you customized `/usr/vpix/etc/vpixdevs` and/or `/usr/vpix/defaults/vpix.cnf`, save them temporarily under other names.
2. Use the `sysadm installpkg` option to install the new release of the VP/ix Environment. Install both subsets on the *Core* diskette before you install the *MS-DOS* diskette. (See the “VP/ix Environment Installation Instructions” if you need information on installing the VP/ix Environment.)

3. If you saved `/usr/vpix/etc/vpixdevs` and `/usr/vpix/defaults/vpix.cnf`, you can edit the ones distributed with this release to include your customized changes, or you may simply overwrite the release versions with your customized versions.
4. If you added any new IEMs to your old VP/ix installation, you must use the `sysadm vconfig` option to reconfigure the VP/ix Environment so that the IEMs will be accessible to your updated system.

VP/ix* Environment

Release 1.2

Release Notes

March 1990

1. INTRODUCTION

This document describes the enhancements made to release 1.2 of the VP/ix Environment and outlines important installation instructions for updating your system.

- ☛ Read this entire document before attempting to update your VP/ix Environment software.

2. ENHANCEMENTS

A number of enhancements have been included in this release.

2.1 LIM 4.0 EMS

VP/ix now supports the Lotus*-Intel*-Microsoft* (LIM) Enhanced Memory Specification 4.0.

The new LIM 4.0 driver is backward compatible with the old LIM 3.2. The DOS driver is named PEMM.SYS. It uses the same command line syntax as the previous 3.2 driver (EMS.SYS). The driver must be placed in the CONFIG.SYS file on the C: drive if it is to be used. If you are using a release earlier than INTERACTIVE UNIX Operating System Version 2.2, you can only use EMS.SYS. You may use either EMS.SYS or PEMM.SYS if you are using INTERACTIVE UNIX Operating System Version 2.2. When using any LIM driver on INTERACTIVE UNIX Operating System Version 2.2, the system performance will be significantly degraded. It is recommended to use only 32 pages of LIM when using EMS.SYS and 48 pages of LIM when using PEMM.SYS. The LIM performance issue will be addressed in a future release of VP/ix. See section 5.2 of the "VP/ix Environment Maintenance Procedures" for more information.

2.2 The AUTOFLUSH Mechanism

VP/ix now has automatic printer buffer flushing.

This is an option you can choose to configure into your VP/ix session. It will automatically flush the data from the print buffer whenever a printer does not receive anything to print for a specified period of time. Your VP/ix configuration file contains the AUTOFLUSH entry. To enable it, remove the semi-colon. This will cause the printer to AUTOFLUSH every 20 seconds. The number after the AUTOFLUSH entry in the `vpix.cnf` file determines the number of seconds of idle time before a printer is flushed. If you set this value to 0, the autoflush mechanism will be disabled and your printer will not be automatically flushed. Otherwise, the minimum accepted value is 5 seconds. Values set less than this will default to 5. There are no warning messages printed on startup if the number can not be read correctly; the autoflush mechanism is just disabled. If the AUTOFLUSH entry is commented out, the autoflush mechanism will be disabled.

When using `PRT-SCR` with AUTOFLUSH enabled, the printer buffer will be flushed automatically after the `PRT-SCR` is done.

The VP/ix User Interface Menu still contains the (P)rinter flush entry. Using this option with AUTOFLUSH enabled has no adverse effects on printing.

2.3 Personal `autoexec.bat` Files

A new feature has been added that allows individual users to have their own `autoexec.bat` files. Users can create a DOS text file containing DOS batch file commands. The file must reside in the user's home directory and must have the UNIX System file name `autoexec.bat`. When DOS boots, it will execute the default `autoexec.bat` file located on the C: drive, followed by the user's `autoexec.bat` file if it exists. This eliminates the need to give VP/ix users their own copy of the C: drive if only the `autoexec.bat` file needs to be customized.

3. PROBLEM RESOLUTIONS

A number of problems have been resolved for this release.

3.1 DOS Applications Under VP/ix

Some DOS applications would not work when executing a DOS *lseek* system call or checking for disk space on the z: drive. Specifically, this caused problems when using the DOS application WORD* 5.0. These bugs have been fixed.

The shift state of the left **SHIFT** and/or **NUM-LOCK** keys would change intermittently. This caused problems when using Super Calc5* and other new DOS applications. This has now been fixed.

Applications that use the DOS system call to print files will now work faster.

VP/ix has been made less prone to failure when using the serial mouse. This will alleviate problems seen when using the Microsoft serial mouse with Microsoft Windows*.

3.2 European Keyboard Emulation

The European keyboard emulation would not type the U.S. characters when using the **ALT** key on the right side of the keyboard. This has been fixed.

3.3 DDA Devices on IRQ Lines

Installing DDA devices on IRQ lines greater than 7 would not work. This has been fixed and DDA devices installed on IRQ lines 8 through 15 will now properly send hardware interrupts into DOS.

3.4 Error Message for 1 to 2 User VP/ix Version

A new error message has been introduced for the 1 to 2 user VP/ix version. The error message "only 2 unique users allowed" will be displayed, and VP/ix will exit if more than two different users try to run VP/ix at the same time. Previously, VP/ix would exit without an error message.

4. INSTALLATION

4.1 Before You Install the VP/ix Environment

The installation instructions in these release notes apply only to systems on which a previous release of the VP/ix Environment is currently installed. If you do not currently have the VP/ix Environment installed on your system, you can follow the procedures found in the “VP/ix Environment Installation Instructions.”

If you *do* have the VP/ix Environment currently installed on your system and you have changed the `/usr/vpix/etc/vpixdevs` or the `/usr/vpix/defaults/vpix.cnf` file from its original configuration, save the file or files temporarily under another name before you follow these instructions. If you do not do so, your customized file(s) will be overwritten. Note that the `vpixdevs` file *has* been changed if you added any VP/ix IEMs to your original VP/ix installation.

4.2 New Distribution of Subsets

This release of VP/ix is now distributed as three separate subsets. The disks are labeled *Core*, *DOS*, and *Configuration*. The *Core* diskette contains the VP/ix binaries. The *DOS* diskette is new for this release and contains MS-DOS* which was contained in the *Core* subset in previous releases. The *Configuration* diskette allows one to reconfigure the VP/ix Environment.

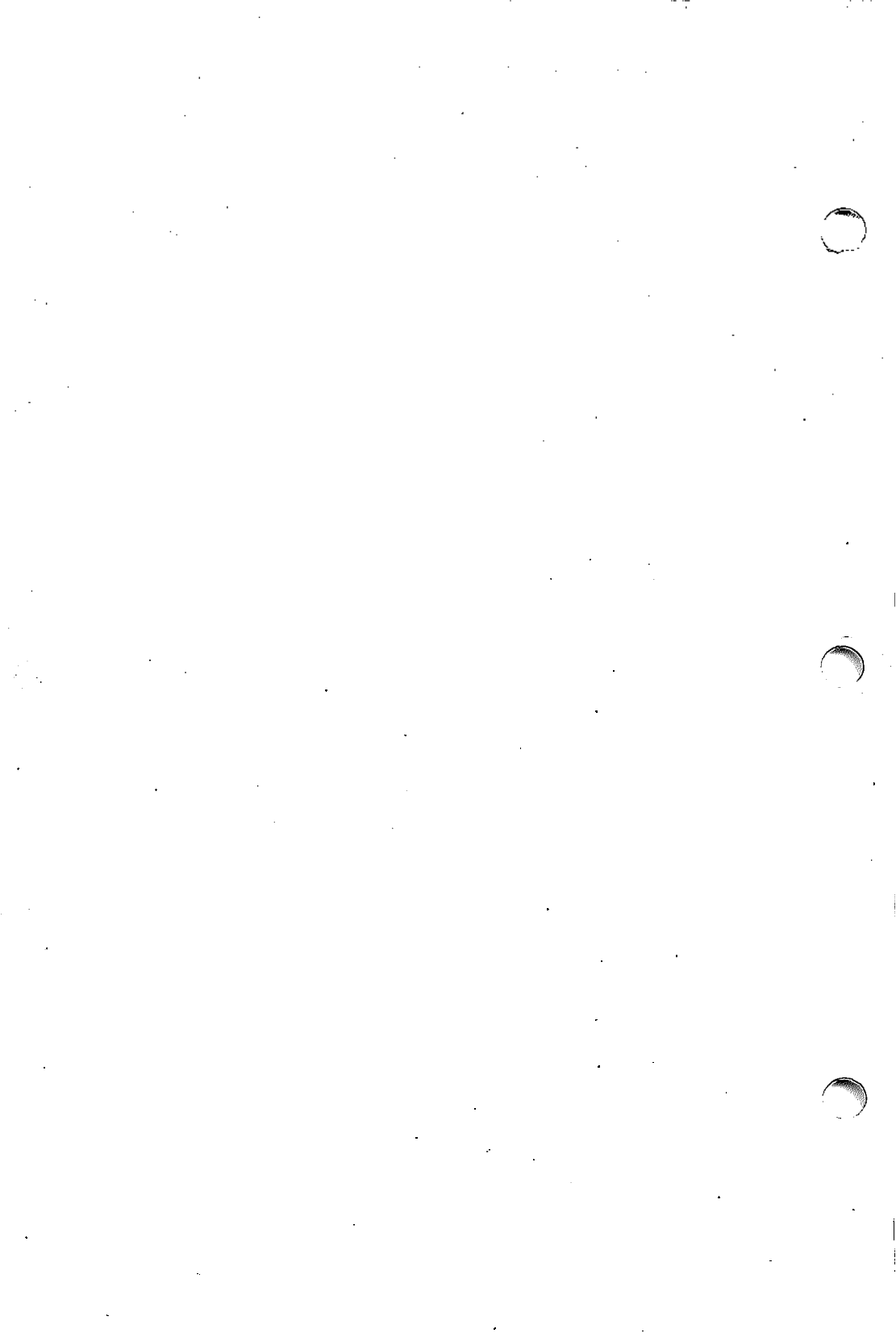
4.3 Updating the VP/ix Environment

To update your VP/ix Environment software, follow this procedure:

1. If you customized `/usr/vpix/etc/vpixdevs` and/or `/usr/vpix/defaults/vpix.cnf`, save them temporarily under other names.
2. Use the `sysadm installpkg` option to install the new release of the VP/ix Environment. You must install the VP/ix Environment *Core* and *MS-DOS* diskettes before installing the VP/ix Environment *Configuration* diskette. (See the “VP/ix Environment Installation Instructions” if you need information on installing the VP/ix Environment.)
3. If you saved `/usr/vpix/etc/vpixdevs` and `/usr/vpix/defaults/vpix.cnf`, you can edit the

ones distributed with this release to include your customized changes, or you may simply overwrite the release versions with your customized versions.

4. If you added any new IEMs to your old VP/ix installation, you must use the `sysadm vconfig` option to reconfigure the VP/ix Environment so that the IEMs will be accessible to your updated system.



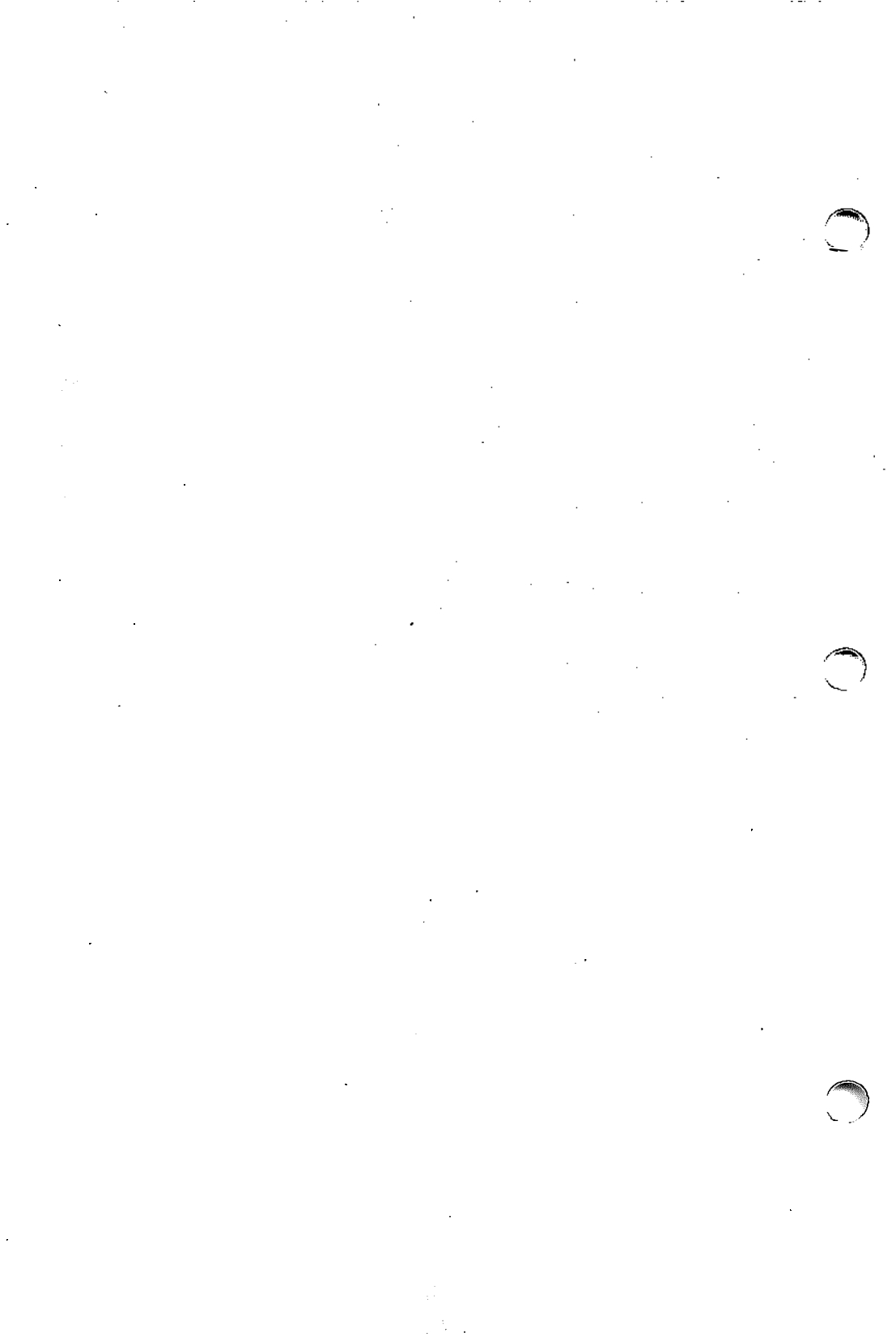
VP/ix and MS-DOS Primer

CONTENTS

INTRODUCTION	1
What Will I Learn From This Primer?	1
GETTING STARTED WITH VP/ix	2
Before You Begin	2
Logging In to the UNIX System	2
Accessing the VP/ix Environment	4
The VP/ix Interface Menu	4
Running DOS Applications	8
Printing From the VP/ix Environment	8
Exiting the VP/ix Environment	10
Logging Out of the UNIX System	10
UNDERSTANDING DISKS AND DRIVES	11
What Is a Disk?	11
What Is a Disk Drive?	11
Accessing the Diskette Drive	12
Using the Z Drive	12
USING DOS COMMANDS	14
What Is a Command?	14
Command Syntax	15
Command Names	16
Using a Simple Command	16
Command Arguments	18
Command Options	19
Stopping Commands	20
USING AND NAMING FILES	21
What Is a File?	21
Naming DOS and UNIX System Files	21
Invalid File Names	22
Wildcards	23
FILE SYSTEMS AND DIRECTORIES	26
Directories	26
File Systems	26
File Names and Path Names	28
Path Name Shortcuts: Dot and Dot Dot	30

Creating Directories (<code>mkdir</code>)	31
Changing Directories (<code>cd</code>)	32
Deleting Directories (<code>rmdir</code>)	33
Renaming a Directory	34
BASIC DOS COMMANDS	35
Copying DOS Files (<code>copy</code>)	35
Deleting Files (<code>del</code> and <code>erase</code>)	37
Renaming Files (<code>rename</code>)	38
Displaying Files (<code>type</code>)	39
Displaying Files (<code>more</code>)	40
Searching for Text (<code>find</code>)	41
Sorting Files (<code>sort</code>)	42
Formatting Diskettes (<code>format</code>)	42
Copying Diskettes (<code>diskcopy</code>)	44
Checking Diskettes (<code>chkdsk</code>)	45
DIFFERENCES BETWEEN DOS AND UNIX SYSTEM	
FILE SYSTEMS	47
The Z Drive	48
File Access Permissions	49
Determining UNIX System File Information	
(<code>xdir</code>)	50
Converting Text File Formats	51
Transferring Files Between the UNIX System and	
DOS	53
Changing UNIX System and DOS Conventions	53
ADVANCED DOS OPERATIONS	55
Loading DOS Applications	55
Internal and External Commands	55
Setting a Search Path (<code>path</code>)	56
Redirecting Input and Output	58
Using Filters	59
Using Pipes	60
USING DOS AND UNIX SYSTEM COMMANDS	
TOGETHER	61
Running DOS Commands From the UNIX System	
Prompt	61
Running DOS Applications From the UNIX System	
Prompt	62
Invoking the UNIX System Shell From the VP/ix Interface	
Menu	62

Running UNIX System Commands from the DOS Prompt	63
Using Pipes With Both UNIX System and DOS Commands	63
CONFIGURING THE VP/ix ENVIRONMENT	65
The <code>autoexec.bat</code> File	65
The VP/ix Configuration File	66
Appendix A: DOS AND VP/ix COMMAND SUMMARY	69
Appendix B: DOS UNDER THE VP/ix ENVIRONMENT	71
Understanding Disks and Drives	71
Using DOS Commands	72
Using and Naming Files	72
File Systems and Directories	73
Basic DOS Commands	73
Using DOS and UNIX System Commands Together	74
Configuring the VP/ix Environment	74
Appendix C: USING ASCII TERMINALS	77
GLOSSARY	81



VP/ix* and MS-DOS* Primer

INTRODUCTION

What Will I Learn From This Primer?

This primer is an introduction to the basic capabilities and commands available with VP/ix and MS-DOS (DOS). In this primer you will learn:

- How to enter and exit the VP/ix Environment.
- How and when to use the VP/ix Interface Menu.
- How to print files and run DOS *applications* in the VP/ix Environment.
- What disks and drives are and how they are used with DOS.
- How to use the VP/ix Z drive.
- The components that make up DOS commands and how to execute them.
- How to give files names that will be acceptable to both DOS and the UNIX* *operating system*.
- What directories are and how to create, delete, and rename them.
- What a file system is and the differences between the DOS and the UNIX System file systems.
- How to redirect input and output.
- How to use pipes and filters.
- How to use DOS and UNIX System commands together.
- How the VP/ix Environment is configured.

If you are a first time DOS user, read this entire primer.

If you are an experienced DOS user, read the sections entitled "GETTING STARTED WITH VP/ix," "DIFFERENCES BETWEEN DOS AND UNIX SYSTEM FILE SYSTEMS," and "USING DOS AND UNIX SYSTEM COMMANDS TOGETHER," then refer to Appendix B, which explains the important differences between running DOS under VP/ix and native DOS.

GETTING STARTED WITH VP/ix

In this section, you will learn:

- How to begin and end a VP/ix session.
- How to perform basic tasks, such as printing a file or running a DOS application.

Before You Begin

This document assumes that your system administrator has already installed your system, installed the VP/ix Environment, and set up your *login account*. Your login account is the information the computer uses to identify you and allow you access to the system. To log in, you must know your user name and your password (if one has been issued to you). If you do not know your user name and password, see your system administrator.

The VP/ix Environment can be invoked from either a system *console* or an ASCII terminal attached to a serial communications port on your computer. If you are operating VP/ix from an ASCII terminal, the VP/ix Environment may behave somewhat differently than described in this primer. ASCII terminal users should consult Appendix C for more information.

Logging In to the UNIX System

To use the UNIX System, you must turn on your computer or terminal and log in. If you are logging in to the console, your screen will look similar to this:

```
Console Login:
```

If you are using a terminal other than the console, your screen will look similar to this:

```
login:
```

Type the user identification name (ID) assigned to you, then press **ENTER**. Your user ID can contain a maximum of eight characters and can consist of uppercase letters, lowercase letters, and numerals. By convention, most user IDs on UNIX systems consist of lowercase letters.

After a user ID is entered, the system may request a password:

```
login: tony
Password:
```

If you have been assigned a password, type your password when the **Password:** prompt appears and press **ENTER**.

☛ The system does not display the password you type on the screen.

If you have not been assigned a password and the **Password:** prompt is displayed, simply press **ENTER**. Your screen will look similar to this:

```
login: tony
Password:
```

```
*** Welcome to the UNIX Operating System. ***
```

```
$
```

When you have completed the login procedure, the system displays a *prompt* on the screen. The prompt is a symbol, usually a dollar sign (\$) or a percent sign (%), but because this character can be changed by your system administrator, your prompt may be different. The prompt indicates that the system is ready to receive information. When the prompt is displayed on your screen, you can enter a command or run an application program.

Each time you log in, the system places you in your *home directory*. Your home directory serves as your personal work area.

A *directory* (such as your home directory) is a special type of file that can contain both files and other directories. It is similar in function to a file cabinet: a file cabinet has several drawers (directories), each of which can contain many folders (subdirectories), each holding one or more documents (files). You will learn more about directories in the section entitled “FILE SYSTEMS AND DIRECTORIES” in this primer.

To determine the name of your home directory after logging in to the UNIX Operating System, type:

```
$ pwd
```

Accessing the VP/ix Environment

You will run the VP/ix Environment from the UNIX System prompt. To access the VP/ix Environment, type the `vpix` command, followed by **ENTER**:

```
$ vpix
```

VP/ix displays several messages and then the default VP/ix prompt. Your screen will look similar to this:

```
Welcome to MS-DOS Running in the VP/ix Environment
VP/ix Z:\usr\tony>
```

The default VP/ix prompt is:

```
VP/ix Z:\directoryname>
```

The word `VP/ix` indicates that you are running in the VP/ix Environment. `Z:` indicates that the UNIX System *file system* on the fixed disk (the `Z:` drive) has been established as your default disk drive. The *directoryname* indicates that you are located in the directory from which you ran the `vpix` command; in the example above, `\usr\tony` is the name of the user's home directory. You will learn more about disk drives and directories in subsequent sections. `>` is the prompt symbol that functions similarly to the UNIX System `$` prompt. When the VP/ix prompt is displayed on the screen, you can run DOS applications.

If you are in a hurry, you can use the `vpix` command with its option `-r` to “quickboot” the VP/ix Environment. This means that DOS will start up immediately without displaying all the usual messages on your screen. To start VP/ix quickly, type:

```
$ vpix -r
```

The VP/ix Interface Menu

The *VP/ix Interface Menu* is used to control the VP/ix Environment. It facilitates resource sharing among multiple VP/ix users and manages each user's individual VP/ix session. The VP/ix Interface Menu can be accessed at any time during a VP/ix session. All DOS processes are temporarily suspended when the VP/ix menu is active.

☛ To access the menu on a system console with an 84-key keyboard, use **SYS-REQ**.

- ☛ To access the menu on a system console with a 101-key keyboard (the function keys are across the top rather than along the left side), use **[ALT]** and **[SYS-REQ]** simultaneously, followed by **[m]**.
- ☛ To access the menu on an ASCII terminal, consult your system administrator for the proper key sequence.

Throughout this document **[SYS-REQ]** is used to invoke the VP/ix Interface Menu. Use the key sequence that is appropriate for your system.

Use **[SYS-REQ]** to access the VP/ix menu. You will see the menu and a display similar to the ones shown below; the VP/ix Interface Menu is on the left-hand side:

```

VP/ix Version 1.2
(ESC)ape menu
(F)loppy release
(P)rinter flush
(Q)uit VP/ix
(R)eset VP/ix
(S)ound OFF
(C)lose serial
(E)nter shell
  
```

Device	Assignment
A:	/dev/rdisk/f0q15dt
C:	/usr/vpix/defaults/C:
Terminal	Assignment
AT386	/dev/console

There may be more or fewer menu options, depending on your hardware.

The secondary display shown on the right-hand side of your screen appears every time you invoke the VP/ix Interface Menu. It displays the user's current default VP/ix Environment configuration and is for informational purposes only. No action is possible. Note that the values in the display will change if the user's defaults change or if the user is configured to use specific DOS devices. For more information about this display, refer to the "The VP/ix Configuration File" section of this document and section 2 of the "VP/ix Environment Maintenance Procedures" in this guide.

To select a menu option, position the cursor using the arrow keys (up-arrow **[↑]** or down-arrow **[↓]**) and press **[ENTER]**, or use the key (lowercase or uppercase) indicated in parentheses that corresponds to the option. The menu disappears from the screen after an option is selected. (Note: the VP/ix menu only responds to keys that are in the same key location as those labeled on a U.S. English Keyboard.

Since some keys on European Keyboards are labeled differently, the key indicated in parentheses on the VP/ix menu, may not be the correct key to select on that particular keyboard. To avoid confusion when using European Keyboards, users may select options from the VP/ix menu by positioning the cursor using the arrow keys and pressing **[ENTER]**.)

You will use the VP/ix menu primarily to print text or data from a DOS application and to reset the VP/ix Environment. All of the options are described briefly below. Some are described in greater detail in later sections of this primer.

■ If you are running a *graphics application* and invoke the VP/ix Interface Menu, it will *not* appear on your screen. However, selected options will take effect exactly as if you could see the menu. For example, you will still be able to reset VP/ix by using **[SYS-REQ]** **[F]**.

(Esc)ape menu

Removes the menu without invoking an option and returns you to your VP/ix session.

(F)loppy release

Releases all *diskette* (floppy disk) drives. Diskette drives in the VP/ix Environment must be exclusively owned by one VP/ix user at a time. When your DOS application attempts to access a diskette drive, VP/ix automatically reserves that drive for your exclusive use, if it is available. However, the VP/ix system cannot automatically release the drive once it is acquired, since it has no way of knowing when you are finished with the drive. Therefore, you must explicitly release the drive with this option. This option is automatically invoked when you exit the VP/ix Environment.

(P)rinter flush

Sends text that is waiting to be printed to the UNIX System printer *spooler*. In the VP/ix Environment, when you use a DOS command or a DOS application command to print data on a printer, the print data is stored rather than sent directly to the printer. On a *multi-user* system such as the INTERACTIVE UNIX System, someone else may be using the printer at any given time. Since there is no explicit “done with printer” operation in DOS, you must tell the VP/ix system to actually send the stored data to the printer via the UNIX System printer spooler. The data will then be

queued for printing and actual printing will begin as soon as the printer is available. (The printer is also automatically flushed when you exit the VP/ix Environment.) Note that if you are printing a large file, you must wait a few minutes after using the DOS command or the DOS application command before you flush the printer, or your print job may be split into parts.

(Q)uit VP/ix

Exits the VP/ix Environment. This is the same as typing `quit` at the VP/ix prompt, except that the screen will be restored to the way it looked when you entered VP/ix. The system automatically releases all system resources (diskette drives, serial ports) when this option is used. Simply use **SYS-REQ** **q** to exit the VP/ix Environment.

(R)eset VP/ix

Reboots the VP/ix Environment. This option is equivalent to using the standard **CTRL** **ALT** **DEL** keyboard sequence to reboot (which may be used in the VP/ix Environment as an alternative to this menu option); the VP/ix Environment is restarted just as if you had exited and reentered it. This option is used if your DOS application has inadvertently locked your terminal.

(S)ound OFF

(S)ound ON

This option turns off and on the “beeps” that the system produces. If the computer beeps frequently while you are using an application or a game, you may want to turn off the sound by selecting this option. When the **(S)ound OFF** option has been selected, the menu displays this option as **(S)ound ON**. If you turn on the sound again by selecting this option, the menu displays the option as **(S)ound OFF**.

(C)lose serial

As with diskette drives, *serial communications ports* must be owned exclusively by one VP/ix user at a time. The VP/ix Environment has no way of knowing when you are finished with a port unless you indicate to the system that you are finished. If you have been using a serial communications port (e.g., with an application such as CrossTalk* or Pro-Comm), use this option to release it as soon as you have completed your session, to make the port available to other

users. This option will appear on your menu if your system configuration supports it. This option is automatically invoked when you exit the VP/ix Environment. (Note that the VP/ix Environment does not currently support the serial communications devices named COM1: and COM2: on MULTIBUS* systems.)

(E)nter shell

Invokes a new UNIX System shell without terminating the VP/ix session (see the section entitled “Invoking the UNIX System Shell From the VP/ix Interface Menu”). When you exit the new UNIX System shell, the current VP/ix screen (without the menu) is redisplayed and the VP/ix session is resumed.

Running DOS Applications

In the VP/ix Environment, DOS applications are installed and run using the same procedures you normally use in a native DOS environment. Ask your system administrator or refer to the documentation supplied with your DOS application to learn how to run your application. For example, if you have a word processing application that you normally run by using the `word` command, type `word` at the VP/ix prompt:

```
VP/ix Z:\usr\tony> word
```

Printing From the VP/ix Environment

The VP/ix Environment has an option to enable automatic printer buffer flushing (AUTOFLUSH). If you choose to configure this option into your VP/ix session, it will automatically cause the data to be flushed from the printer buffer whenever a printer does not receive anything to print for a specified period of time.

Your VP/ix configuration file contains an entry for AUTOFLUSH. To enable it, remove the semi-colon. This will cause the printer to AUTOFLUSH every 20 seconds. The number after the AUTOFLUSH keyword in the `vpix.cnf` file determines the number of seconds of idle time before a printer is flushed. If you set this value to 0, the autoflush mechanism is disabled. The minimum accepted value following the AUTOFLUSH keyword (other than 0) is 5. If the value specified is less than this, it will default to 5 seconds. Note, if the value specified cannot be read

correctly at startup, the autoflush mechanism is disabled. No warning messages are printed.

If the autoflush mechanism is disabled, then in most cases you must start two distinct processes to print from the VP/ix Environment. You must:

- Initiate the print request, either through a DOS command or from within a DOS application.
- Cause the print request to be passed to the spooler program so that it can be queued for printing.

If you use **PRT-SCR** or initiate a print request from a DOS application, following the instructions provided with that application, or use a DOS print command at the VP/ix prompt, the text or data that would normally begin printing at this point is instead stored by the VP/ix Environment.

You must then instruct the system to send the print request to the printer. Access the VP/ix menu, using **SYS-REQ**:

VP/ix Version 1.2

```
(ESC)ape menu
(F)loppy release
(P)rinter flush
(Q)uit VP/ix
(R)eset VP/ix
(S)ound OFF
(E)nter shell
```

Type p to select the (P)rinter flush option. It is not necessary to press **ENTER**. The menu disappears from the screen, and your data begins to print as soon as the printer is available.

If you are printing a long file, you must wait a few minutes after using the DOS or DOS application print command before you flush the printer. If you do not wait for the DOS command to complete before using the Printer Flush option, only part of your print job may appear. The rest will appear the next time you use the Printer Flush option. A rule of thumb is to wait approximately 1 minute for every two pages of print job before flushing the printer.

The printer is also automatically flushed when you exit the VP/ix Environment.

Note that the use of the (P)rinter flush option when AUTOFLUSH is enabled has no adverse effects upon printing. The use of **PRT-SCR** with AUTOFLUSH enabled causes the printer buffer to be flushed automatically after the **PRT-SCR** is pressed.

Exiting the VP/ix Environment

You may exit the VP/ix Environment at any time. If you are using an application that is stored on a diskette, always exit the VP/ix Environment when you have finished so that the diskette drive will become available to other users. If your DOS application requires that you save or exit your file, use the appropriate commands before you exit the VP/ix Environment.

To exit the VP/ix Environment, type `quit` at the VP/ix prompt:

```
VP/ix Z:\> quit
```

```
$
```

You will be returned to the UNIX System prompt.

Logging Out of the UNIX System

When the UNIX System prompt displays, you can exit (log out of) the UNIX System. To exit the UNIX System, hold down the **CTRL** key and simultaneously type `d`, or type `exit` at the prompt:

```
$ exit
```

The computer displays the `login:` prompt, indicating that it is ready to accept a login name for the next session.

UNDERSTANDING DISKS AND DRIVES

This section provides an overview of the fixed *disk drive* and diskette drives, which are the primary media for storing data and programs. In this section you will learn:

- What fixed disks and diskettes are.
- What a disk drive is.
- How to access a disk drive.
- What the Z drive is.

What Is a Disk?

All commands and data are stored on a medium called a disk. There are two kinds of disks: *fixed disks* and *diskettes*. A fixed disk is a collection of rotating, magnetized disks sealed into a unit that resides permanently in your computer. A diskette is a removable medium composed of a magnetized surface, on which information can be stored.

What Is a Disk Drive?

Fixed disks and diskettes are accessed using a device called a disk drive. There are two kinds of disk drives on DOS systems: fixed disk drives and diskette drives. The fixed disk drive is usually the *default* drive on a computer, that is, it is automatically made the current drive when you invoke the VP/ix Environment.

On DOS, each available drive is assigned a single letter as its name. The default drive name assignments in the VP/ix Environment are shown below:

<i>Drive Name</i>	<i>Device</i>
A	the first diskette drive
B	the second diskette drive
C	the bootable DOS image on the fixed disk drive
Z	the UNIX file system on the fixed disk drive

Drive assignments and the default drive assignment can be altered. Consult your system administrator for more information.

Only one drive is active at any given time. To use a different drive, you must activate it with a command. Note that before you activate a diskette drive, you must insert a diskette into that drive.

To activate a new drive, type the name of the drive and a colon (:), followed by **ENTER**. For example, if your current drive is Z, the UNIX System file system on the fixed disk, you may activate the first diskette drive by inserting a diskette into that drive and typing:

```
VP/ix Z:\usr\tony> a:
```

The first diskette drive, A, will become the active drive. Your prompt will now resemble this:

```
VP/ix A:\sales>
```

Accessing the Diskette Drive

On a VP/ix system the diskette drive is referred to as drive A.

- ☛ Your environment must be configured correctly to use drive A; not all system users have access to it. You can tell if you have access to drive A: by using the pop-up VP/ix Interface Menu. The secondary display that appears on the right will have an entry for A: in it if you are allowed access to the diskette drive. If it does not and you require this access, consult your system administrator.

Only one user can use the diskette drive at a time. If another person is using the diskette drive, you will not be able to access it. If you are using the diskette drive, when you exit the VP/ix Environment, it is automatically released. If you are finished with the diskette drive, but want to remain in the VP/ix Environment, you should release the diskette drive by invoking the VP/ix Interface Menu (using **SYS-REQ**) and selecting the (F)loppy release option.

Using the z Drive

Unlike *single-user* computers running DOS, the VP/ix Environment supports a multi-user file system, located on the Z drive. The Z drive contains the UNIX System file system on the fixed disk. This is the same file system you access when you log in to the

INTERACTIVE UNIX Operating System. (For more information on UNIX file systems, refer to the “INTERACTIVE UNIX Operating System Primer” in the *INTERACTIVE UNIX System Guide for New Users.*)

- ☛ Only files accessed through the UNIX System file system on the Z drive can be shared by multiple VP/ix users.

All VP/ix users on your computer can simultaneously access the Z drive; therefore, files and directories that others need to access frequently should be created on the Z drive. Although the files on the Z drive are UNIX System files, all the DOS commands described in this primer can be used with them.

USING DOS COMMANDS

Once you have entered the VP/ix Environment, you can run both DOS and UNIX System commands and programs. This section discusses how to use DOS commands. In this section you will learn:

- What a DOS command is.
- DOS command syntax.
- What options and arguments are.
- How to issue a DOS command.
- How to stop a DOS command.

What Is a Command?

The words *command* and *program* are nearly synonymous. In simple terms, the user types a command, followed by **ENTER**, and the operating system executes the program that performs the user's "command."

Commands are typed at the system prompt. When a command is entered, it is called a *command line*. It can have three parts: the command itself, its *options* (also referred to as *switches* and *flags*), and its *arguments*. The command name is the name of the program that performs the desired action. An option is a special kind of argument that is specific to a particular command. It changes the behavior of the command in some way. In DOS, an argument usually consists of a file, directory, user name, or drive name. (The UNIX Operating System does not use drive names.) An argument gives the system information that is required to process a specific command or to change the standard behavior of a command.

Each command line is terminated by using **ENTER**. The **ENTER** key is labelled **RETURN** on some systems but is frequently referred to as **ENTER** in DOS software documentation. When the **ENTER** is received, the command is sent to the operating system for execution.

Command Syntax

The UNIX Operating System is *case sensitive*, which means that it distinguishes between upper- and lowercase letters. DOS is *case insensitive*, that is, it does *not* distinguish between upper- and lowercase letters. Throughout this document, commands are shown in lowercase letters; however, you can use any combination of uppercase and lowercase letters when typing commands to DOS. It converts everything you type to uppercase letters.

■ When using a UNIX System command, you may *not* use either upper- or lowercase characters; you must type the command in the correct case.

DOS options typically begin with a forward slash (/). Each command, option, or argument consists of one word, which is interpreted as a group, or *string*, of characters surrounded by spaces.

If you make an error when typing a command, use **BACKSPACE** to correct the error. You may use the *left* and *right* cursor positioning keys (such as **←**), but you may *not* use the up and down cursor positioning keys (such as **↑**).

Always type the command name first, followed by a space. Next type the desired option or options, each followed by a space, then any arguments, separated by spaces. (You can also use the semicolon (;), the equal sign (=), or **TAB** between DOS commands and their options. In this manual, commands and options are shown separated by a space.)

This primer presents commands in the following format:

COMMAND NAME	command name
FORMAT	command [option(s)] argument(s)
DESCRIPTION	A brief description of what the command does.
OPTIONS	A list of the most useful options and a brief description of each.
ARGUMENTS	Mandatory or optional arguments.

If an argument is not required, it is shown in square brackets []. Options are always “optional,” so they are always shown in square brackets. Only the most common options and arguments are discussed in this primer. Some commands may also include step-by-step instructions. For a complete list of the available options and arguments, refer to the *MS-DOS Reference Guide*.

Command Names

Command names are short or abbreviated words that describe the programs they invoke. They are deliberately kept short to save time and reduce typing errors. On DOS, you can abbreviate some command names to two or three letters. For example, you can either type `rename` or `ren` when using the `rename` command. (Note that this is *not* true on UNIX systems.) In the discussion of each command, there may be information on acceptable abbreviations (or substitutions) that can be used when typing the command. For complete information on command abbreviations, refer to the *MS-DOS Reference Guide*.

Using a Simple Command

The `dir` command is an example of a command that can be executed using only the command name. It is used to list the names of the files and directories in the current directory. Files and directories are discussed in more detail in the sections entitled “USING AND NAMING FILES” and “FILE SYSTEMS AND DIRECTORIES”.

COMMAND NAME	<code>dir</code>
FORMAT	<code>dir [/w] [drive:][pathname]</code>
DESCRIPTION	Displays information about the files and directories in the specified drive and directory. If no drive or path is specified, list all files and directories in the current directory on the default drive.
OPTIONS	<p><code>/w</code> Display the names of all files and directories in the current directory, listed across the screen.</p> <p><code>/p</code> Display the names of all files and directories in the current directory one page at a time.</p>
ARGUMENTS	A drive and a path name to a file or a directory can be specified. If no drive is specified, use the default drive. If no path name is specified, list all directory entries in the current directory.

If you type `dir` at the prompt the system displays information about the file and directory names on the default drive in the *current directory*. For example, if the current drive is C and you type:

```
VP/ix C:\> dir
```

the screen will look similar to this:

```
Volume in drive C is DOS
Directory of C:\

COMMAND  COM   23612   4-21-87  12:00p
BACKUP   EXE   22906   4-21-87  12:00p
CHKDSK   EXE   9680    4-21-87  12:00p
MODE     EXE   13652   4-21-87  12:00p
      4 File(s) 232608 bytes free
```

The `dir` command lists the full names of all files and directories in the current directory, and also the size in *bytes* (a unit for measuring disk space), the time of last modification for each file and directory, and the number of files and the amount of free space available (in bytes). Note that the `dir` command does not display the period that separates a file name from its extension.

After issuing a command, you may be instructed to provide more information. Usually the command provides a description of the appropriate responses.

When the instruction **Press any key** appears, you may press any letter of the alphabet (a–z), number (0–9), or the **SPACEBAR** key.

Command Arguments

An argument gives the system information that is required to process a specific command or to change the default, or standard, behavior of a command. DOS command arguments usually include the following components:

drive: Disk drive name. A disk drive is either a fixed disk drive or a diskette drive. Fixed disk and diskette drives are sometimes referred to as *source drives* and *target* or *destination drives*. A source drive is the drive from which you will be transferring information. A destination drive is the drive to which you will be transferring information.

file name The name of a document or collection of information stored on the computer, including the *file name extension*, if one exists. The file name extension is a suffix of one to three characters, separated from the file name by a period (.). The file name option never refers to the name of a device or a drive. You must include the file name extension when referring to a file that has a file name extension. File names are discussed in more detail in the section entitled “USING AND NAMING FILES.”

path name The sequence of directory and file names that describes the location of a file or directory on the system. The path name of a file follows the general format:

`\directory\directory\filename`

Path names are discussed in greater detail in the section entitled “FILE SYSTEMS AND DIRECTORIES.”

strings Strings of text that are specific to a command and give the DOS system additional information.

Some commands require one or more arguments; arguments are optional for other commands. The `dir` command, for example, accepts a file or directory name as an optional argument. If you type `dir` followed by a file name, the system displays the file name and information about the file if the file exists in the current directory:

```
VP/ix Z:\usr\tony> dir budget

Volume in drive Z is UNIX
Directory of Z:\usr\tony

BUDGET TXT    23612    4-21-87  12:00p
          1 File(s) 2232008 bytes free
```

If a directory name is used as the argument, the system displays the names of all the files and directories in that directory:

```
VP/ix Z:\usr\tony> dir letters

Volume in drive Z is UNIX
Directory of Z:\usr\tony\letters

ACKNOWL      TXT    23612    4-22-87  10:54a
MARKET       TXT    22906    4-21-87  11:02a
MARKET2      TXT    23332    4-21-87  11:45a
HARRIS       TXT    34118    4-19-87  9:02a
          4 File(s) 2232608 bytes free
```

You can also display the files on a specific drive by specifying the drive name as an argument. A drive name tells DOS to look on the diskette in a specific drive to find the named file. For example, to display the files and directories in the default directory on drive B, you would make sure a diskette is in the B drive and type:

```
VP/ix Z:\usr\tony> dir b:
```

Command Options

An option is a special kind of argument that is specific to a particular command. As its name implies, an option is not required, but it provides additional versatility when used with a command. Most DOS options begin with a forward slash (/). Note that this is different from the UNIX System. UNIX System options begin with a dash (-). Some DOS commands accept both arguments and options.

For example, the `dir` command can be used with the `/w` or “wide” option, which displays the names of the files and directories in columns across the display. If you type:

```
VP/ix C:\> dir /w
```

then the screen will look similar to this:

```
Volume in drive C is DOS  
Directory of C:\
```

```
COMMAND COM  BACKUP EXE  CHKDSK EXE  MODE EXE  
4 File(s) 232608 bytes free
```

Stopping Commands

You can permanently stop commands while they are running by using **CTRL** **c**.

When commands produce a large amount of output on the screen, the display automatically scrolls to the next screen. Use **CTRL** **s** to suspend the scrolling. Use **CTRL** **q** to resume scrolling.

USING AND NAMING FILES

This section reviews how DOS files are used and named. In this section you will learn:

- What a file is.
- How to name a DOS file.
- How to name a file so that it is acceptable to both the UNIX System and DOS.
- What DOS wildcards are and how to use them in DOS commands.

What Is a File?

A file is a named collection of related information stored on a computer, a letter or a report, for example. You create a file each time you enter and save data or text at your terminal. Files are also created when you write, name, and save programs.

Naming DOS and UNIX System Files

All programs, text, and data on disks or diskettes reside in files. Each file has a unique name. In DOS, a complete file name consists of a file name and its extension. File names can be one to eight characters and can contain letters, numbers, and symbols. The optional file name extension can be one to three characters and can contain letters, numbers, and symbols. It must be separated from the file name by a period.

A typical DOS file name looks like this:

FORECAST.EXE

FORECAST is the file name; **EXE** is the file name extension.

Examples of other acceptable file names are:

```
accounts.feb
budget.84
smithco.ltr
chapter1.nvl
CHAP.6A
schedule
Workup.jun
```

If you use both upper- and lowercase letters in file names, be aware that DOS converts all lowercase letters into uppercase. Therefore, the file names `letter`, `Letter`, and `LETTER` are all viewed by DOS as the same file. It is a good idea to use only lowercase letters in your file names.

DOS and the UNIX Operating System have different file naming conventions. In contrast to DOS file names, UNIX System file names can be one to fourteen characters long and can include both uppercase and lowercase letters. In the UNIX System, lowercase letters are *not* converted to uppercase letters. Therefore, the names `letter`, `Letter`, and `LETTER` are all viewed by the UNIX System as different files. UNIX System file names do not have “extensions,” so they can contain more than one period and have more than three characters after a period.

When accessing the UNIX System file system through the DOS Z drive, UNIX System file names that do not follow the DOS naming conventions will be changed into names that *do* follow the DOS conventions. These names will contain one or more tildes (~), so it will be easy to see which UNIX System file names do not follow the DOS conventions.

☛ To make a UNIX System file name identical when viewed through both the UNIX System and DOS, give it a name that follows all the limits imposed on DOS file names and contains *only* lowercase letters. It is probably best to use only letters and numbers in your file names and extensions, although some other characters are allowed.

Invalid File Names

DOS treats some device names in special ways. Certain three-letter names are reserved for the names of these devices. These three-letter names cannot be used as file names, but can be used as extensions. You must *not* use any of the following as file names:

AUX	Refers to input from, or output to, an auxiliary device, such as a printer or a diskette drive.
COM1	Refers to the first asynchronous communications port.
COM2	Refers to the second asynchronous communications port.
CON	Refers to keyboard input, or to output to the terminal console (screen).
LPT1	Refers to the first line or parallel printer.
LPT2	Refers to the second line or parallel printer.
LPT3	Refers to the third line or parallel printer.
NUL	Is used when you do not want to create a file, but the command requires an input or output file name.
PRN	Refers to the printer.

You may not add device designations or file name extensions to these file names. DOS assumes that all files that contain these characters refer to the devices listed above. For example, CON.XXX is interpreted as a reference to the console by DOS. It cannot be used to name a file stored on a fixed disk or diskette.

Wildcards

A *wildcard* is used to match a character or a string of characters in a file or directory name. It is a type of shortcut that can be used in commands that require file names and extensions as arguments. A wildcard is also referred to as a *pattern matching character*.

Wildcards are convenient because they enable you to use a file name or group of file names without having to type the complete name. Wildcards can also be used when you only remember part of a file name.

There are two DOS wildcard characters:

- ? Matches any one character.
- * Matches zero or more characters.

When you create new files, be very careful not to use a wildcard character as part of the file name. It is very difficult to access or remove a file if there is a wildcard character in its name.

WILDCARD	?
FORMAT	<i>string?</i> <i>?string</i> <i>?string?</i> <i>string??</i>
DESCRIPTION	Substituting a question mark in a file or directory name argument matches any one character for each ? supplied.

The question mark means that no character or one character can occupy that position. For example, the DOS command:

```
VP/ix Z:\usr\tony> dir test?run.exe
```

lists all directory entries on the default drive that begin with the string `test`, have one character or no characters in the next position, are followed by the letters `run`, and have a file name extension of `.exe`. Here are some examples of files that might be listed using the above `dir` command:

```
test1run.exe
test2run.exe
testbrun.exe
```

An asterisk (*) means that zero or more characters can occupy that position or any of the remaining positions in the file name or extension. For example, the command:

```
VP/ix Z:\usr\tony> dir test*.exe
```

lists all directory entries on the default drive with file names that begin with the characters `test` and have an extension of `.exe`. Here are some examples of files that would be listed using the above `dir` command:

```
test.exe
test1.exe
test2run.exe
test6run.exe
testall.exe
```

To list the directory entries for all files named `accounts` on drive `A` (regardless of their file name extensions), type:

```
VP/ix Z:\usr\tony> dir a:accounts.*
```

To list the directory entries for all files with file name extensions of `.txt` (regardless of their file names) on the disk in drive `B`, type:

```
VP/ix Z:\usr\tony> dir b:*.txt
```

This command is useful if, for example, you have given all your text files a file name extension of `.txt`. By using the `dir` command with wildcard characters, you can obtain a list of all your text files even if you do not remember all their names.

The wildcard designation `*.*` refers to all files in the current directory. Note that this can be very powerful and destructive when used with DOS commands. The command:

```
VP/ix Z:\usr\tony> del *.*
```

deletes all files in the current directory, regardless of file name or extension.

The DOS wildcards are similar to two of the UNIX System wildcard characters, `?` and `*`. For more information about UNIX System wildcard characters, see the “INTERACTIVE UNIX Operating System Primer” in the *INTERACTIVE UNIX System Guide for New Users*. For more information about DOS wildcard characters, see the *MS-DOS Reference Guide*.

FILE SYSTEMS AND DIRECTORIES

This section discusses directories and file systems and their place in DOS. In this section you will learn:

- What a directory is.
- What a file system is.
- What the `root` directory is.
- What a path name is.
- Path name shortcuts.
- How to create a directory.
- How to change from one directory to another.
- How to delete a directory.
- How to rename a directory.

Directories

Directories are used to organize files into convenient groups. A directory is a special kind of file that stores other files and information about the files and subdirectories stored in it. Directories contain information about the sizes of the files, their locations on the disk or diskette, and the dates that they were created and last updated. DOS directories are named using the same conventions used for DOS file names.

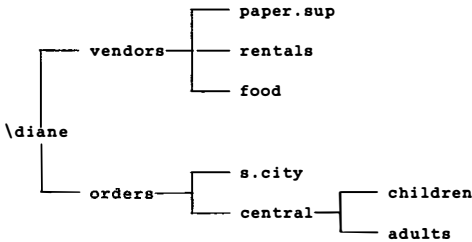
File Systems

Information on the computer is stored in an organized structure, called a *hierarchical file system*. A file system is a collection of individual files that are stored on a disk. It is called hierarchical because of its multi-level structure. A single master directory is at the top level, and additional files and directories are defined at various levels below it.

The master directory for the entire system is called the *root directory*. It is the highest-level directory in the file system structure and is named backslash (`\`) on DOS. (On UNIX systems, the `root` directory is named slash (`/`.) The `root` directory contains several directories and each of these directories can contain other directories. Because a hierarchical file system is sometimes

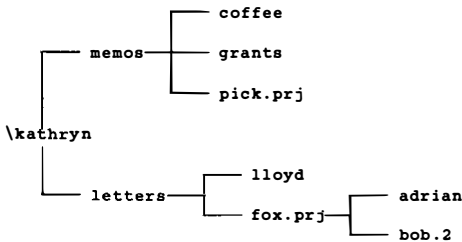
visualized as a “tree,” each new directory created on the system can be viewed as a new branch added to the directory tree.

Here is a diagram of a simple directory structure for a computer user named Diane:

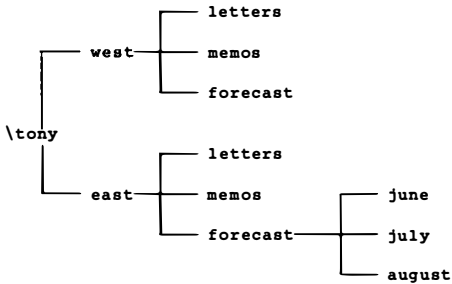


The root directory, (\), contains a directory named `diane`. `\diane` contains two directories: `vendors` and `orders`. These in turn, hold other files and directories.

The organization of your files and directories can be changed at any time to fit your needs. For example, you might have a directory in your home directory called `memos`, which contains all the memos you have written. Or, you could have a `letters` directory, which could contain a number of other directories, one for each of your major projects:



If you deal with different regional offices, you might have a major directory for each region, with each region directory containing parallel directories for letters, memos, and forecasts:



The only limit on the number of files and directories you can have, or the number of directory levels you can have, is the amount of disk or diskette space available to you.

File Names and Path Names

When you want to access a directory or file in the file system, you use a *path name*. A path name is a sequence of directory names and file names, separated by backslashes, that designates the location of a particular file or directory. Since the *full path name* is actually the complete name of the file or directory, it ensures that no two directories (and no two files) on the system have exactly the same full name. To use the filing cabinet analogy again, you can locate a letter addressed to John Smith using its “path name,” by knowing it is in the second cabinet drawer, behind the divider marked “Boni Project,” in a folder marked “letters.”

On DOS, path names and file names may be preceded by a drive name (also known as a *device specifier*). A device specifier is the same as the name of the disk drive you want to access, A: or B:, for example. UNIX System path names and file names do not use drive names; the correct device is automatically accessed with the path name.

There are two kinds of path names, a full path name and a *relative path name*. A full path name on DOS *always* begins with a backslash (\) and consists of the sequence of directories from the *root* directory to the file or directory you wish to access. A relative path name does *not* begin with a backslash. It describes the location of the target file or directory relative to your current location (current directory).

A full path name consists of a backslash followed by one or more directory names separated by backslashes and ending with the name of the target file or directory. It may be preceded by a drive name, which tells the system on which device the desired file or directory is located. It is only necessary to use the drive name if you are *not* referring to the current drive. The backslash at the beginning of the path name tells the system to begin its search for the target from the `root` directory. (If it does not begin with a backslash, DOS begins at the user's current directory.) For example, if the file that contains Tony's eastern region August forecast is on the current drive, its full path name would be:

```
\tony\east\forecast\august
```

If his directory is on the A drive rather than the current drive, its full path name would be:

```
a:\tony\east\forecast\august
```

A relative path name omits all directory names and the drive name from the path name up to and including the current directory. It tells the system to begin the search for the target file or directory from the current directory. You can access a file in your current directory by using its relative path name. In this case, the relative path name consists of only the file name. Or, you can use the relative path name to access a file in a subdirectory of your current directory by specifying the name of the subdirectory followed by a slash and the name of the file. For example, user Tony can list the contents of his `july` directory by using the relative path name with the `dir` command:

```
VP/ix Z:\tony> dir forecast\july
```

It is not necessary to use the full path name to the directory, `\tony\forecast\july`.

When using DOS or VP/ix, you may not specify a path name containing more than 63 characters, including the initial backslash. If you need to specify a path name longer than 63 characters, change to a directory in that path, then specify the relative path name, limiting the path name to no more than 63 characters.

Because a directory is simply a special type of file, you cannot give a subdirectory the same name as another file in a directory; all files within a single directory must have a unique file name.

Path Name Shortcuts: Dot and Dot Dot

DOS provides shorthand notations to indicate the current directory and the directory one level above it. These shortcuts are available to help minimize typing. A single dot (.) indicates the current directory. Two dots (..) indicate the directory directly above the current directory.

For example, you may use the dot with the `copy` command to copy all the files in the current directory to a subdirectory in that directory.

```
VP/ix Z:\usr\tony> copy . tabs
```

The “dot,” when used in place of a path name, indicates that the command should use all the files in the current directory.

To list the files and directories in the directory just above your current directory, type:

```
VP/ix Z:\usr\tony> dir ..
```

To list the files in the directory two levels up from your current directory, type:

```
VP/ix Z:\usr\tony> dir ..\..
```

Creating Directories (`mkdir`)

The `mkdir` command is used to create subdirectories in which you can organize your files.

COMMAND NAME	<code>mkdir</code> or <code>md</code>
FORMAT	<code>mkdir [drive:]filename or pathname</code> <code>md [drive:]filename or pathname</code>
DESCRIPTION	Creates a directory with the name given.
OPTIONS	None.
ARGUMENTS	To create a directory on a drive other than the default drive, you must supply a drive name. To create a directory in your current directory, give a file name. To create a directory somewhere other than your current directory, give the path name.

For example, to create a new directory named `memos` in your current directory, type:

```
VP/ix Z:\usr\tony> mkdir memos
```

After DOS runs this command, the new subdirectory, `memos`, exists in your current directory. You are not automatically moved to the new directory; you remain in your current directory. To verify that the directory was created, use the `dir` command to list the directory contents:

```
VP/ix Z:\usr\tony> dir
```

Your screen should look similar to this:

```
MEMOS   <DIR>      7-19-87   11:22a
      1 File(s)  2443575 bytes free
```

You can create directories anywhere in the directory structure by using `mkdir`, followed by the path name of the directory you want to create.

Changing Directories (cd)

To change to another directory, use the `cd` command.

COMMAND NAME	<code>cd</code> or <code>chdir</code>
FORMAT	<code>cd [drive:] [pathname]</code> <code>chdir [drive:] [pathname]</code>
DESCRIPTION	Changes the current directory to another directory. Without an argument, <code>cd</code> prints the name of the current drive and the path name of the directory in which you are currently located.
OPTIONS	None.
ARGUMENTS	The drive name where the directory is located (if it is not on the current drive) and the path name of the directory to which you want to change.

For example, to change from your current directory to the directory `\tony` on your current drive, type:

```
VP/ix Z:\usr\bob> cd \tony
```

followed by **ENTER**. If the directory `\tony` is not on the current drive, you must use the appropriate drive name as well. For example, `cd a:\tony`.

Use `cd` without an argument to verify that your current directory is now `\tony`. Of course, in the VP/ix Environment, the prompt keeps you informed of the current directory.

To change to the directory directly above your current directory, use the shortcut you learned previously:

```
VP/ix Z:\usr\tony> cd ..
```

Deleting Directories (rmdir)

To delete a directory on a disk or diskette, use the DOS `rmdir` command.

COMMAND NAME	<code>rmdir</code> or <code>rd</code>
FORMAT	<code>rmdir [drive:]pathname</code> <code>rd [drive:]pathname</code>
DESCRIPTION	Removes the specified directory, if empty.
OPTIONS	None.
ARGUMENTS	The path name of the directory to be removed.

The directory must be empty in order to be removed. If it is not empty, DOS returns the message `directory not empty`. This prevents you from accidentally deleting files and directories.

For example, to remove the directory `\tom\letters`, first remove all the files in the directory by typing:

```
VP/ix Z:\tom> del \tom\letters
```

Since you do not give a specific file name to be removed, DOS assumes you want to delete *all* the files in the named directory. DOS asks:

```
Are you sure (Y/N)?
```

Type `y`. Then remove the directory itself by typing:

```
VP/ix Z:\tom> rmdir \tom\letters
```

If you are in the directory `\tom`, you can use the relative path name and simply type `rmdir letters`.

You can delete any directory by specifying its full path name. To remove the `\amy\sales_n` directory on a diskette in drive `A`, make sure that it is empty, then type:

```
VP/ix Z:\tom> rmdir a:\amy\sales_n
```

Renaming a Directory

There is no single command in DOS that allows you to rename a directory. To rename a directory that has no subdirectories, do as follows. If you have a directory named `\memos` that you would like to rename `memos_aug`, create a new directory with the new name using `mkdir`:

```
VP/ix Z:\usr\tony> mkdir \memos_aug
```

Copy all files from the old directory to the new directory using `copy`:

```
VP/ix Z:\usr\tony> copy \memos\*.* \memos_aug
```

Next, delete the contents of the old directory, using `del`:

```
VP/ix Z:\usr\tony> del \memos\*.*
```

The system will issue the following prompt:

```
Are you sure (Y/N)?
```

Type `y`. Delete the old directory using `rmdir`:

```
VP/ix Z:\usr\tony> rmdir \memos
```


BASIC DOS COMMANDS

This section presents some basic DOS commands used to perform necessary tasks on your files and diskettes. In this section you will learn:

- How to copy DOS files.
- How to delete DOS files.
- How to rename DOS files.
- How to display DOS files.
- How to display DOS files one screen at a time.
- How to find a particular string of text in files.
- How to sort DOS files.
- How to format a DOS diskette.
- How to copy a DOS diskette.
- How to check a DOS diskette for errors.

Copying DOS Files (`copy`)

Use the `copy` command to copy one or more files. It will copy a file or files onto the same drive or onto another drive.

COMMAND NAME	<code>copy</code>
FORMAT	<code>copy [drive:]pathname [drive:][pathname]</code>
DESCRIPTION	Copies one or more files to another drive or to the same drive.
OPTIONS	Presented later in this primer.
ARGUMENTS	The source drive (the drive that has the file to be copied on it) and the target drive (the drive onto which you want to copy the files).

To copy a file from a diskette in the current drive (A, for example) to the same diskette in drive A, use the command:

```
VP/ix A:\usr\phil> copy a:john.ltr a:phil.ltr
```

The `copy` command copies the file named `john.ltr` on the diskette in drive `A` to the file `phil.ltr` on the diskette in drive `A`. Note that you do not have to use a drive letter when you designate the current drive. If the current drive is `A`, (the prompt is `VP/ix A:\usr\phil>`), you can make a copy of the file named `john.ltr` and call it `phil.ltr` by typing:

```
VP/ix A:\usr\phil> copy john.ltr phil.ltr
```

To make an exact copy of a file from the diskette in drive `A` to the fixed disk (with `A` as the current drive), use the command:

```
VP/ix A:\usr\phil> copy a:john.ltr z:john.ltr
```

The `copy` command copies the file `john.ltr` on drive `A` to the file `john.ltr` on drive `Z`. Because drive `A` is the current drive, you could also type the following to accomplish the same task:

```
VP/ix A:\usr\phil> copy john.ltr z:john.ltr
```

If you do not specify a file name for the copy, `copy` uses the name of the original file for the new file. For example, each of the following commands copies `john.ltr` to the current directory on the `Z` drive:

```
copy a:john.ltr z:john.ltr
copy john.ltr z:john.ltr
copy john.ltr z:
```

The command:

```
copy john.ltr z:\john.ltr
```

copies `john.ltr` to the root directory of the UNIX System file system, if that directory is writable.

See the section entitled “Converting Text File Formats” for more information about transferring DOS and UNIX System files.

Deleting Files (del and erase)

Use the `del` or `erase` commands to delete a file.

COMMAND NAME	<code>del</code> <code>erase</code>
FORMAT	<code>del [drive:]pathname</code> <code>erase [drive:]pathname</code>
DESCRIPTION	Deletes all files with the designated file specification. Wildcards are accepted; however, only one path name can be deleted at a time.
OPTIONS	None.
ARGUMENTS	A drive and a path name to the file or directory to be deleted.

For example, to delete the file `sales.txt` from the current directory on drive `Z`, use the command:

```
VP/ix Z:\usr\tony> del sales.txt
```

The file is permanently deleted from the fixed disk.

You may also use wildcards when deleting files from a directory.

■ *Be careful when you use the `*.*` abbreviation. Typing the command `del *.*` will cause DOS to delete all files in the current directory.*

If you type:

```
VP/ix Z:\usr\tony> del *.*
```

DOS asks you to confirm that you want to delete the files:

```
Are you sure (Y/N)?
```

Type `Y` (yes) or `N` (no).

Note that the `del` command does not work if you type the word `delete`.

Renaming Files (`rename`)

To change the name of a file, use the `rename` command.

COMMAND NAME	<code>rename</code> or <code>ren</code>
FORMAT	<code>rename [drive:]pathname filename</code> <code>ren [drive:]pathname filename</code>
DESCRIPTION	Renames files.
OPTIONS	None.
ARGUMENTS	If the file resides on a drive other than the default drive, you must supply a drive name. Any drive name to the <i>filename</i> option is invalid and produces an error message. You cannot rename files across drives.

To change the name of a file on the fixed disk called `memo.doc` to `billmemo.doc`, type:

```
VP/ix Z:\usr\tony> rename memo.doc billmemo.doc
```

The file `memo.doc` no longer exists; an identical file called `billmemo.doc` is now on your fixed disk.

☛ You can use `rename` only on files on the *same* disk or diskette. That is, you cannot use drive letters to change `a:memo.doc` to `b:memo.doc`.

To change the name of a file called `story.doc` on drive A to `fable.doc`, first make sure that the diskette containing `story.doc` is in drive A, then type this command:

```
VP/ix Z:\usr\tony> rename a:story.doc fable.doc
```

The file is now named `fable.doc`.

Displaying Files (type)

To display the contents of a file on the screen, use the `type` command.

COMMAND NAME	<code>type</code>
FORMAT	<code>type [drive:][pathname]filename</code>
DESCRIPTION	Displays the contents of a UNIX System or DOS text file on the screen.
OPTIONS	None.
ARGUMENTS	If the file resides on a drive other than the default drive, you must supply a drive name. You must supply a file name.

To view the contents of the file named `boats.cal` in the current directory, type:

```
VP/ix Z:\usr\tony> type boats.cal
```

`boats.cal` is displayed on the screen. If the file is longer than a single screen, it will continue to scroll by quickly until you stop the scrolling. Use **CTRL** **s** to stop the scrolling. To continue displaying the text of the file, use **CTRL** **q**. Depending on the length of the file, you may have to do this several times if you wish to read the entire file.

You can display both DOS and UNIX System files using the `type` command.

To display the contents of a file named `deadline.10` on a diskette in drive A, first make sure that the diskette with the file named `deadline.10` is inserted in drive A. If A is not the default drive, then change to the A drive. Either type:

```
VP/ix Z:\usr\tony> a:
VP/ix A:\> type deadline.10
```

or type:

```
VP/ix Z:\usr\tony> type a:deadline.10
```

followed by **ENTER**. The contents of `deadline.10` is displayed on the screen.

Program files (files with a file name extension of `.com` or `.exe`) yield only uninterpretable symbols when displayed using `type`. Because of the format differences between DOS and UNIX System text files, the `type` command does not work very well on UNIX System files. See the section entitled “DIFFERENCES BETWEEN DOS AND UNIX SYSTEM FILE SYSTEMS” for information on converting files from one format to the other.

DOS files created while running some application programs may require that you run the application program to view the contents of those files. They cannot be viewed using `type`. Refer to the material provided with your application program for more information.

Displaying Files (`more`)

The `more` command is similar to the `type` command, but sends its output to the console one screen at a time.

COMMAND NAME	<code>more</code>
FORMAT	<code>more < [pathname]filename</code>
DESCRIPTION	Sends output to the display one screen at a time. This command takes its input from the keyboard. On displaying one screen of text, the <code>more</code> command pauses and displays the message <code>--More--</code> at the bottom of the screen. To display the next screen, press the ENTER key. This process continues until all the text has been displayed.
OPTIONS	None.
ARGUMENTS	None.

`more` is useful for displaying the contents of files. To use `more` to display a file, you must use the input/output redirection symbol `<`. This is discussed in detail in “Redirecting Input and Output” in the section entitled “ADVANCED DOS OPERATIONS.” To use `more` to display the text of a file named `tom.txt`, type:

```
VP/ix Z:\usr\tom> more <tom.txt
```

Because of the format differences between DOS and UNIX System text files, the `more` command does not work very well on UNIX

System files. See the section entitled “DIFFERENCES BETWEEN DOS AND UNIX SYSTEM FILE SYSTEMS” for information on converting files from one format to the other.

Searching for Text (`find`)

The `find` command searches for a specific string of text in a file or files.

COMMAND NAME	<code>find</code>
FORMAT	<code>find [/n] "string" [drive:][pathname]</code>
DESCRIPTION	Searches for a specified string in the designated file name or names, and displays the line or lines containing the string.
OPTIONS	<code>/n</code> Displays each line found preceded by the relative line number.
ARGUMENTS	The string, surrounded by double quotes. Do not include a double quote as part of the search string. You can optionally specify the drive and path name of the file or files to be searched.

For example, to search for the string `Halloween` in the file `review.10`, type:

```
VP/ix Z:\usr\tony> find "Halloween" review.10
```

This displays all lines from `review.10` that contain the string `Halloween`.

As another example, to search for the string `fool's paradise` in the files `book1.txt` and `book2.txt`, type:

```
VP/ix Z:\usr\tony> find "fool's paradise" book1.txt book2.txt
```

This displays all lines from `book1.txt` and `book2.txt`, in that order, that contain the string `fool's paradise`.

☛ Note that unlike most DOS commands, `find` is case-sensitive with regard to the search string.

Sorting Files (`sort`)

The `sort` command is used to sort data alphabetically, in either ascending or descending order.

COMMAND NAME	<code>sort</code>
FORMAT	<code>sort [/r] [drive:][pathname]</code>
DESCRIPTION	Reads standard input, sorts the data, then writes the sorted data out.
OPTIONS	<code>/r</code> Sort in reverse order, starting with the letter Z.
ARGUMENTS	The drive and path name of the file to be sorted. Otherwise, <code>sort</code> takes input from the keyboard as the data to be sorted.

For example, to alphabetically sort names as you type them in at the terminal, type:

```
VP/ix Z:\usr\tony> sort
```

followed by **ENTER**.

The `sort` command is generally used to sort data already in files. For an example of this, see “Using Filters” in the section entitled “ADVANCED DOS OPERATIONS”.

Formatting Diskettes (`format`)

You must “format” all new diskettes before they can be used by DOS. Formatting checks the diskette for defects and makes the diskette usable by DOS. To format a diskette, use the `format` command. Diskettes need to be formatted only once, the first time they are used. Diskettes formatted by a UNIX System utility do not need reformatting with a DOS utility, and vice versa.

☛ Note, however, that the `format` command destroys any information that is on a diskette.

COMMAND NAME	<code>format</code>
FORMAT	<code>format [drive:] [/4]</code>
DESCRIPTION	Prepares the diskette in the specified drive to accept DOS files.
OPTIONS	/4 Format diskette double-sided, double-density (the default is to format it as a high-density diskette)
ARGUMENTS	The drive in which the diskette to be formatted is located. If no drive is specified, the default drive is used.

To format a blank diskette, first make sure that you are using DOS (that is, you have logged in and at the UNIX System prompt, you have typed `vpix`). To format a high-density diskette, type:

```
VP/ix Z:\usr\tony> format a:
```

Your screen will look similar to this:

```
Insert new diskette for drive A
and strike ENTER when ready
```

Insert a diskette in drive A and press **ENTER**.

When formatting is complete, DOS asks:

```
Format another? (Y/N)
```

Type N (no) or Y (yes) and press **ENTER**.

To format a double-sided, double-density diskette in drive A, type:

```
VP/ix Z:\usr\tony> format a:/4
```

☛ *Use the format command on diskettes only.* Although the `format` command is sometimes used to format fixed disks on standard DOS systems, it should *not* be used for that purpose in the VP/ix Environment.

Copying Diskettes (`diskcopy`)

The `diskcopy` command is used to copy the contents of diskettes. To copy a diskette using `diskcopy`, you must have:

- The diskette you want to copy.
- A diskette to copy it to. (If the diskette is not blank, any existing data will be destroyed; it will be overwritten by the contents of the diskette you are copying.)

COMMAND NAME	<code>diskcopy</code>
FORMAT	<code>diskcopy [source drive:] [target drive:]</code>
DESCRIPTION	Copies the contents of the diskette in the first specified drive (the source drive) to the diskette in the second specified drive (the target drive).
OPTIONS	Not presented in this document.
ARGUMENTS	A source drive, containing the diskette with the files to be copied, and a target drive, containing the diskette on which the files are to be copied. If you give only one drive name as an argument, it is used as both the source and target drive. You are prompted to remove the first diskette and insert the second one.

To copy the contents of a diskette from drive A to drive B, first put the diskette you want to copy into drive A. Put another diskette into drive B. (The diskette does not need to be formatted because the `diskcopy` command will automatically format it during the command process.) Type:

```
VP/ix Z:\usr\tony> diskcopy a: b:
```

and press **ENTER**. Your screen will look similar to this:

```
Insert SOURCE diskette in drive A:
```

```
Insert TARGET diskette in drive B:
```

```
Press any key when ready ...
```

Since you have already placed the diskettes in the proper drives, you can press any key to start the `diskcopy` program. When the disk has been copied, DOS asks:

Copy another? (Y/N)

Type N to exit the `diskcopy` program or Y to copy another diskette.

If you have only one diskette drive, give that drive name only once as an argument:

```
VP/ix Z:\usr\tony> diskcopy a:
```

DOS will prompt you at the appropriate time to remove the diskette you wish to copy and insert the blank, formatted diskette.

☛ *Use the `diskcopy` command on diskettes only. It is not intended for use on fixed disks.*

Checking Diskettes (`chkdsk`)

The DOS `chkdsk` command is used to check diskettes for consistency and errors, much like a secretary proofreading a letter. The `chkdsk` command analyzes the specified diskette and produces a status report of any problems, such as files that have a non-zero size in the directory but really have no data in them. You should run `chkdsk` occasionally on each diskette to make sure the files on it are not corrupted.

☛ You cannot use the `chkdsk` command on the Z drive because the Z drive is not a DOS file system and it is not on a diskette; the Z drive contains the UNIX System file system on the fixed disk.

COMMAND NAME	chkdsk
FORMAT	chkdsk [/f] [drive:] [pathname]
DESCRIPTION	Scans the diskette in the specified drive and checks it for errors.
OPTIONS	/f Correct any errors found in the directory.
ARGUMENTS	The drive that contains the diskette with the files to be copied. A path name; if a path name is specified, DOS displays a status report for the diskette and for the specified file.

To check the diskette in drive A, type:

```
VP/ix Z:\usr\tony> chkdsk a:
```

This displays a status report and any errors that `chkdsk` has found:

```
Volume DOSDISK create July 12, 1987 3:21p
```

```
362496 bytes total disk space
 38912 bytes in 2 hidden files
  1024 bytes in 2 directories
282624 bytes in 42 user files
 39936 bytes available on disk
```

```
524288 bytes total memory
487248 bytes free
```

DIFFERENCES BETWEEN DOS AND UNIX SYSTEM FILE SYSTEMS

This section discusses the differences between the DOS and UNIX System file systems. In this section you will learn:

- How permissions apply to the Z drive.
- How to determine the permissions of files and directories on the Z drive.
- How to determine the real name of UNIX System files with illegal DOS names.
- How to copy text files from DOS format to UNIX System format and from UNIX System format to DOS format.
- How to change path name and option conventions from DOS format to UNIX System format and from UNIX System format to DOS format.

The directory structures of DOS and the UNIX System are similar. Both store data in a collection of files and directories that are arranged in a hierarchical order. However, it is not possible to access files in a UNIX System file system from native DOS or files in a DOS file system with a standard UNIX System. The formats of DOS and UNIX System files are somewhat different. Certain UNIX System programs that are not part of the VP/ix Environment, such as `cpio` and `tar`, use different strategies for storing data on diskettes. Disks created using these programs cannot be used by the DOS system. The path name conventions for the two operating systems are also different.

The VP/ix Environment supports many features that make DOS and UNIX System files and path names compatible. The VP/ix Environment provides:

- Access to the UNIX System file system while in the VP/ix Environment.
- Utilities to convert DOS text files to a UNIX System format.
- Utilities to convert UNIX System text files to a DOS format.
- Utilities to change DOS-style path name and option conventions to UNIX-style conventions.
- Utilities to change UNIX-style path name and option conventions to DOS-style conventions.

The z Drive

The z drive, as implemented by the VP/ix Environment, enables you to access the UNIX System file system while running DOS. The UNIX System file system, when accessed through the z drive, behaves exactly like a native DOS file system. You can use standard DOS commands and editors (such as `mkdir` or `edlin`) to manipulate all of the files in the UNIX System file system. You may also use any UNIX System file with DOS applications and install your DOS applications on the UNIX System file system.

In many 386* configurations there are no true DOS file systems available on the computer. If this is the case on your computer, the files you wish to create and access while in the VP/ix Environment should either be on diskettes (in drive A or B) or on the Z drive. Ask your system administrator if drives other than the Z drive are available on your fixed disk. Regardless of the availability of other drives, files placed on the Z drive are more versatile because they are available to you and to others when you are at the UNIX System prompt running UNIX System commands.

The Z drive is the default drive in the VP/ix Environment. When you type `vpix` at the UNIX System prompt, you are automatically placed in the Z drive. Your prompt will look similar to this:

```
VP/ix Z:\usr\tony>
```

Your current directory is the same UNIX System directory from which you started the VP/ix Environment. While on the Z drive, you can access any file or directory that is stored under the UNIX System file system.

Each time you change from one drive to another in the VP/ix Environment, you are placed in the directory you accessed last on that drive. When you access the Z drive, VP/ix places you in the last directory you accessed on the Z drive under the UNIX System or DOS. For example, if you started the VP/ix Environment from the UNIX System directory `/usr/tony`, the system places you in `z:\usr\tony`:

```
VP/ix Z:\usr\tony>
```

If you then access another drive (the A drive, for example) and return to the Z drive, you will be returned to the directory `z:\usr\tony`.

File Access Permissions

The UNIX System is a multi-user file system. This means that multiple users have access to the files and directories that are stored on the computer. To prevent unauthorized users from accessing data on the system, a set of *file access permissions* (also known as “protection modes”) is assigned to each file and directory on the system. In addition, each file and directory is assigned an owner (user) and a group identification number. File and directories are usually owned by the user who created them and are assigned the same group ID number as the user.

Because the VP/ix Environment runs in a multi-user environment and provides access to the UNIX System file system through the Z drive, it must also support file access permissions. The file access permissions determine who can access a given file on the Z drive. File access permissions are not assigned to files on any other DOS disk drive (A, B, or C) because they are DOS file systems. Each file and directory on the Z drive has two protection modes, read (r) and write (w), associated with it.

Read permission

Allows a user to view the contents of the file or copy it to another area. Read permission by itself does not allow a user to change a file.

Write permission

Allows a user to make changes to a file. Write permission on a directory allows a user to create and delete files in that directory.

The protection modes are assigned to three categories of users: *user*, *group*, and *other*. The “user” of the file owns the file or directory. The “group” often consists of people working on the same project or in the same department who may need to use the same files. Group membership is usually determined by the system administrator. “Other” refers to everyone on the system who is neither the user nor in the user’s group.

When a file or directory is created on the Z drive, it is automatically assigned default permissions. These are as follows:

<i>User Category</i>	<i>Permissions</i>
User	read, write
Group	read
Other	read

To change the permissions of files on the Z drive, the UNIX System command `chmod` must be used. See the “INTERACTIVE UNIX Operating System Primer” in the *INTERACTIVE UNIX System Guide for New Users* for information on how to use `chmod`.

Determining UNIX System File Information (`xdir`)

The VP/ix Environment command `xdir` is used to display file access permissions for files on the Z drive, as well as their user and group ID numbers. It also displays the actual UNIX System names of files. If you have given a file a name that is invalid in the DOS operating system, `xdir` can be used to determine its true name. It is otherwise identical to the DOS command `dir`.

COMMAND NAME	<code>xdir</code>
FORMAT	<code>xdir z:[pathname]</code>
DESCRIPTION	Displays information about the files and directories in the specified drive and directory. If no path name is specified, lists all files and directories in the current directory on the current drive. Information includes file name, type, creation date and time, access permissions, and size.
OPTIONS	None.
ARGUMENTS	The <code>z:</code> drive and a path name to a file or a directory.

If you are not on the Z drive, you can type:

```
VP/ix A:\tony> xdir z:\usr\bob
```

to get a listing of the files in `\usr\bob`.

If you are already on the Z drive and in the directory you want to list, type:


```
VP/ix Z:\usr\bob> xdir
```

Your screen will look similar to this:

```
Volume in drive Z is UNIX
Directory of Z:\usr\bob

.           <DIR>    8/05/87 12:48p           .           [rwx-r-] <200,3>
..          <DIR>    8/05/87 12:46p          ..          [rwx-r-] <200,0>
^PROF^AA   150      8/05/87 12:00a   .profile    [rw----] <200,3>
VPIX       <DIR>    8/05/87 12:46p           vpix        [rwxrwx] <200,3>
REPORT     1107    8/05/87 12:47p           report      [rwxr-]  <200,3>

    5 File(s) 33553920 bytes free
```

Each line contains the standard DOS information, as well as the additional UNIX System information. The file's read and write permissions for the user, group, and others are displayed within square brackets. The UNIX System user and group ID numbers are displayed in the last column in angle brackets (< >). The true UNIX System names of the files are shown. The name of the file ^PROF^AA is really .profile.

The permissions for the file named REPORT are as follows:

File Access Permissions

rw	rw	r-
↑	↑	↑
User	Group	Other

The first two columns specify the read (r) and write (w) permissions for the user. In this case, the user can read and modify the file. The next two columns specify the permissions set for the user's group. The permissions indicate that members of the group can read and write the file. The last two columns specify permissions for others on the system. Others are allowed to read the file but cannot modify it. A dash indicates that a permission is denied.

Converting Text File Formats

Text files are stored differently under the UNIX System and DOS. The UNIX System uses the new-line character to specify the end of a line; DOS uses a carriage-return followed by a new-line to specify the end of a line. Because of this, DOS files viewed through a UNIX System editor display control characters at

the end of each line, and UNIX System text files are improperly displayed by many DOS utilities and applications.

You can use the DOS `copy` command with its special VP/ix options to copy a text file from DOS format to UNIX System format and from UNIX System format to DOS format.

COMMAND NAME	<code>copy</code>
FORMAT	<code>copy [/d /u] [drive:]pathname [drive:][pathname]</code>
DESCRIPTION	Copies one or more files to another disk or to the same disk.
OPTIONS	<p><code>/d</code> Copies a UNIX-format text file into DOS format.</p> <p><code>/u</code> Copies a DOS-format text file into UNIX format.</p>
ARGUMENTS	The source drive (the drive containing the diskette that has the files to be copied on it) and the target drive (the drive that contains the diskette onto which you want to copy the files).

To copy a UNIX-format text file to DOS format, use the `/d` option. For example, if the current drive is `A`, use the following command to copy the file called `unix.doc` to drive `B:`, converting the UNIX System file format to DOS file format:

```
VP/ix A:\> copy unix.doc b: /d
```

To copy a DOS text file to UNIX System format, use the `/u` option. For example, to copy the file called `msdos.doc` to `unix.doc`, converting the DOS file format to UNIX System file format, type:

```
VP/ix A:\> copy msdos.doc unix.doc /u
```

There is also a UNIX System utility called `lef` that can convert DOS files to UNIX System format and UNIX System files to DOS format. It is useful if you wish to use a DOS editor to edit a UNIX System file or vice versa. It is normally run from the UNIX System prompt, but it can be run in the VP/ix Environment as well.

Use the `lef` command to change either UNIX System files to DOS format or DOS files to UNIX System format. `lef` automatically determines the current format of the file and changes it to the opposite format.

COMMAND NAME	<code>lef</code>
FORMAT	<code>lef pathname pathname</code>
DESCRIPTION	Determines the current format of the specified text file (either DOS or UNIX) and converts to the other format.
OPTIONS	None.
ARGUMENTS	<code>lef</code> accepts zero, one, or two arguments. If no argument is specified, <code>lef</code> takes its input from the keyboard and sends its output to the terminal screen. If one argument is specified, it is assumed to be the input file and the output is sent to the terminal screen. If two arguments are specified, the first is the input file and the second is the output file.

For example, to change a file named `report` to the other file format, type:

```
VP/ix Z:\tony> lef report report1
```

Transferring Files Between the UNIX System and DOS

Other utilities are available for transferring files across file systems. Standard UNIX System utilities, such as the `cp` and `mv` commands, can be used to copy DOS files on the `Z` drive just as if they were UNIX System files.

Changing UNIX System and DOS Conventions

The syntax used to separate path names and to indicate options in DOS is different from that used on the UNIX System. When you enter the VP/ix Environment, the DOS path name separation character (`\`) and option character (`/`) are in effect, rather than the equivalent UNIX System characters (`/` and `-`, respectively).

If you are more comfortable using the UNIX System and want to have the VP/ix Environment accept and display only the UNIX

System conventions, use the VP/ix utility `unixpath`. Once you invoke `unixpath`, even the VP/ix prompt immediately changes from `VP/ix C:\>` to `VP/ix C://>`.

COMMAND NAME	<code>unixpath</code>
FORMAT	<code>unixpath</code>
DESCRIPTION	Causes the system to use UNIX-style option characters (-) and path name separators (/).
OPTIONS	None
ARGUMENTS	None

If you want to return to the DOS-style characters after you have invoked `unixpath`, use the utility `dospath`.

COMMAND NAME	<code>dospath</code>
FORMAT	<code>dospath</code>
DESCRIPTION	Causes the system to use DOS-style option characters (/) and path name separators (\).
OPTIONS	None
ARGUMENTS	None

It makes no difference whether you access files in a DOS file system (on the A drive, for example) or the UNIX System file system on the Z drive; the VP/ix Environment lets you use one consistent set of characters, either UNIX-style or DOS-style.

ADVANCED DOS OPERATIONS

This section describes advanced DOS commands. In this section you will learn:

- The difference between internal and external commands.
- How to set a search path.
- How to redirect input and output.
- What a filter is.
- What a pipe is.

Loading DOS Applications

DOS applications are loaded onto the Z drive of your fixed disk under the VP/ix Environment exactly as they are loaded onto your fixed disk under native DOS. The loading procedure depends directly on the application. To find out how to load a particular application onto your fixed disk, refer to the documentation that accompanied the application. If you are asked for a drive name during the procedure, use Z rather than C, because the C drive should only be used for booting the VP/ix Environment.

Internal and External Commands

DOS and UNIX System programs are files that are located on either the fixed disk or a diskette. When you invoke a command (`vpix`, for example), the system must locate the command before it can execute the command. DOS stores its commands in two ways. They are either stored as *internal commands* or as *external commands*.

An internal command is a program that is always resident in memory. The internal commands `del`, `mkdir`, `cd`, `copy`, and `type`, for example, are listed in the file `command.com`. The system reads `command.com` when you initially *boot* DOS, and all of the commands listed there are automatically loaded into memory. Commands loaded into memory execute faster than commands that do not reside in memory.

External commands are all the other commands that are not resident in memory. They can be stored in any directory on a fixed disk or diskette and must have a file name extension of either `com`

(command), `exe` (execute), or `bat` (batch), separated from the file name by a period. `diskcopy`, `format`, and `chkdisk` are examples of external commands. Each has the file name extension `com`. However, when you invoke the command, you do not need to type the file name extension. Because external commands are files, you can create commands and add them to your DOS system; these should be given the file name extension `exe` or `com`.

External commands take longer to execute than internal commands because they must be located by the system via a search path, read, and then executed. By default, the system searches the current directory when a command is issued. If the command is not there, the system displays an error message. Unless the commands are located in your current directory, DOS does not know where to find them. You must use the `path` command to explicitly tell DOS the name(s) of the directory or directories that contain your external commands. The system automatically searches each of the of the directories listed, in order, for the command that is invoked.

Setting a Search Path (`path`)

Use the `path` command to set a search path to locate external commands.

COMMAND NAME	<code>path</code>
FORMAT	<code>path [drive:][pathname]; [drive:][pathname]...</code>
DESCRIPTION	Sets a command search path. Specifying <code>path</code> with no argument causes the system to display the current search path.
OPTIONS	None.
ARGUMENTS	The drive(s) and path name(s) to a directory or directories containing external commands.

For example, if you need an external DOS command such as `format`, located in `\bin`, you must tell DOS to look in the `\bin` path to find it. If you type:

```
path \bin
```

DOS searches in your current directory and the `\bin` directory for all commands invoked thereafter. To specify more than one path, separate them with semicolons (`;`). For example, type:

```
path \bin;\usr\tony\bin
```

This sets the search path to include the directories `\bin` and `\usr\tony\bin`. After a path is set, the system looks for commands in your current directory. If it does not locate the command in your current directory, it searches `\bin`. If it does not locate the command in `\bin`, it looks in `\usr\tony\bin`. If the command is not found in any of the three directories, the system returns the error message `Bad command or file name`.

When you are working with several directories, it is a good idea to put all DOS external commands into one separate directory so that you need only include one path name in your `path` command. This practice speeds up command execution, since only one directory needs to be searched.

All VP/ix users may execute `path` as a command at the DOS prompt. When you type the command `path \bin` at the DOS prompt, your path will be set to `\bin` for the duration of your current login session only. When you log out, this search path is no longer in effect.

You can permanently set a new search path by modifying a special file called `autoexec.bat`. This special DOS file is used to set *variables* and to establish a set of login parameters. It is a *batch file* that is located in the `root` directory of the drive from which DOS is booted (started). A batch file is a DOS file that contains one or more commands that are executed sequentially when the file is invoked. In the case of `autoexec.bat`, it is automatically read and executed when DOS is booted. DOS is normally booted from the C drive. `autoexec.bat` can also be used to install programs that remain in memory. `autoexec.bat` can be modified using an editor (such as `edlin` on DOS or `e` or `ed` on the UNIX System). It is typically located on the C drive. If you cannot edit your copy of `autoexec.bat`, consult your system administrator.

To display your current path, type the `path` command without any arguments at the system prompt. If no search path has been set, this returns the message `No path`.

When a VP/ix account is created, a default path is assigned. When initiating a VP/ix session, the system invokes the `autoexec.bat`

file, which contains the `path` command. If you type `path` immediately after entering `VP/ix`, your screen will look similar to this:

```
PATH=Z:\USR\VPPIX\DOSBIN;Y:\VPPIX\UNIXBIN;Z:\USR\VPPIX\UNIXBIN
```

`Y:` is your home directory on the `Z` drive and `\vpix\unixbin` is a special directory for storing your own collection of UNIX System commands. `Z:` is the `Z` drive, which holds the UNIX System file system, `\usr\vpix\dosbin` is where all DOS commands are located,

and `\usr\vpix\unixbin` stores UNIX System programs that can be executed from the `VP/ix` Environment.

If you plan to use your own DOS or UNIX System commands, it is most convenient to place the UNIX System commands in the directory `\vpix\unixbin` under your home directory, since it is already part of your default search path. However, you can create a new directory for them and add that directory to your search path, as explained previously.

Redirecting Input and Output

Every command to the computer requires input and produces output. DOS commands take their input from the *standard input* and direct it to the *standard output*. The standard input is taken from your terminal keyboard. For example, when you type a command and press **ENTER**, you are sending standard input to the system. The standard output for most DOS commands is to the terminal screen. When you type the `dir` command, for example, the standard input is taken from your terminal keyboard and the standard output is written to the terminal screen.

You can change the flow of command input and output by directing a command to take its input from a source other than its standard input and to send its output to a destination other than its standard output. For example, you can direct a command to take its input from a file already stored on the computer and to direct its output to another file or to a device such as a printer. The ability to take input from one place and direct it to another at the time you start a process is called *redirection* of input/output (I/O). Input can come from a file rather than a keyboard, and output can go to a file or to a line printer instead of to the screen.

Two symbols are used to indicate I/O redirection. The “greater than” (>) symbol redirects output, and the “less than” symbol (<) redirects input.

Most commands produce output that is sent to your terminal screen. You can send this information to a file instead by following your command with a greater than symbol (>) and a file name. For example, the `dir` command displays the directory listing of the default drive on the screen. You can use the same command and redirect its output to a file named `newfile` with the command:

```
VP/ix Z:\usr\tony> dir >newfile
```

You will not see any output on your screen. It is placed in the file `newfile` instead.

When redirecting output, you must be careful not to use an existing file name as your output destination. You can *overwrite* the contents of the existing file if the file's permissions allow it, thereby losing all the text or data in it.

If the file `newfile` does not exist, DOS creates it and stores your directory listing in it. If `newfile` already exists, DOS replaces whatever is in the file with the directory listing. If the output file name is a device, such as `LPT1:`, the output will be redirected to that device.

To redirect the output of a command and store the results at the end of a file that already exists *without* overwriting the contents of the file, use two greater than (>>) symbols. This *appends* the output of the command to the end of the file you specify. For example, to add the current directory listing to an existing file called `tracking`, type:

```
VP/ix Z:\usr\tony> dir >>tracking
```

It is often useful to have input for a command come from a file rather than from a terminal. Use the less than symbol (<) to redirect the input from a file. For example, to display the contents of a file named `guests` one screen at a time, type:

```
VP/ix Z:\usr\tony> more <guests
```

Using Filters

A *filter* is a program that accepts input from one source, performs the appointed task on the data, and then writes the results to

another place without changing the input file in any way. The data is “filtered” by the program. Since filters can be put together in many different ways, a few filters can take the place of many specific commands. Examples of DOS filters include the commands `find`, `more`, and `sort`.

For example, to sort a file called `east.emp` and redirect the output to a new file called `eastemp.srt`, type:

```
VP/ix Z:\> sort <east.emp > eastemp.srt
```

You have directed `sort` to take the file `east.emp` as its input by using the `<` symbol and to place the sorted output in a file called `eastemp.srt` by using the `>` symbol.

Using Pipes

A *pipe* is a connection between two or more commands. To create a *pipeline*, you execute a command and pass its output (“pipe” its output) so that it becomes the input for the next command. A pipe is indicated by typing the vertical bar symbol (`|`). While using DOS, you must have write permission in your current directory in order to pipe commands.

Two filters are often used to create a pipeline. The output of a pipeline can be displayed on the screen, or it may be redirected to a file. For example, sometimes you may want the output of one program to become the input for another program. You can use `sort` and `more` together in a pipe. For example, type:

```
VP/ix C:\> sort <mars.txt | more
```

This causes the system to sort the file `mars.txt` and to display it on your terminal one screen at a time.

Another useful pipe involves programs that produce output in columns, such as `dir`. If you want to have these columns sorted alphabetically, type:

```
dir | sort
```

USING DOS AND UNIX SYSTEM COMMANDS TOGETHER

The VP/ix Environment eliminates the boundaries between DOS and the UNIX Operating System. In this section, you will learn:

- How to run DOS commands from the UNIX System prompt.
- How to run DOS applications from the UNIX System prompt.
- How to invoke the UNIX System shell using the VP/ix Interface Menu.
- How to run UNIX System commands from the DOS prompt.
- How to use DOS and UNIX System commands together on the same line by using pipes and filters.

Running DOS Commands From the UNIX System Prompt

It is not necessary to run the `vpix` command each time you want to run a DOS command or application. If you prefer to work in the UNIX System environment, you may run DOS commands or applications that are stored on your fixed disk by simply typing the standard DOS command sequence. For example, at the UNIX System prompt, you may type:

```
$ dir
```

The system displays a directory listing of your current directory and returns you to the UNIX System prompt. All DOS commands, even when run from a UNIX System prompt, follow standard DOS conventions. That is, you must use the slash character (/) to indicate options and the backslash (\) as the separator in path names.

Users running DOS commands directly from the UNIX System shell should note that certain special characters (such as the DOS wildcard characters * and ?, the DOS file separator \, and the DOS switch character /) are interpreted by the UNIX System shell. In these cases the system may behave in an unexpected manner.


To avoid this problem, enclose arguments to DOS commands in single quotes (') when running them from the UNIX System shell. For example, to run the `dir` command with the `/w` option on the directory `sys` on the first diskette, type:

```
$ dir '/w A:\sys'
```

Running DOS Applications From the UNIX System Prompt

Before you run a DOS application, read the user documentation shipped with the software and be prepared to follow its instructions. For example, 1-2-3* requires that your current directory contain all 1-2-3 system files. This is required no matter whether you run 1-2-3 under native DOS, in the VP/ix Environment, or at the UNIX System shell prompt. Thus, if you want to run 1-2-3 at the UNIX System shell prompt, go to the directory where 1-2-3 is installed.

To run a DOS application at the UNIX System prompt, type the name of the DOS application.

 Note that the application *must* be installed on the fixed disk; you cannot run DOS applications that are stored on diskettes from the UNIX System prompt.

The VP/ix Environment is automatically invoked when you access a DOS application. The VP/ix Interface Menu is also available. To access the VP/ix Interface Menu while running a DOS application, use **SYS-REQ**.

If you are unable to run DOS applications directly from the UNIX System prompt, consult your system administrator or “VP/ix Environment Maintenance Procedures” in this guide. Your VP/ix system may be incorrectly configured.

When you are finished using a DOS application, follow the instructions provided with the program to exit the application. The system will return you to the UNIX System prompt.

Invoking the UNIX System Shell From the VP/ix Interface Menu

You can run UNIX System commands from the VP/ix prompt by accessing the VP/ix menu and selecting the option (E)nter shell. The system refreshes (clears) the screen and displays the UNIX System prompt. Your VP/ix session is temporarily suspended. When you are finished using UNIX System commands, exit the shell by using **CTRL d**. The screen refreshes and your VP/ix session is resumed.

Running UNIX System Commands from the DOS Prompt

All of the standard UNIX System commands, namely those that reside in the directories `/bin` and `/usr/bin` on the UNIX System file system, are automatically available in the VP/ix Environment. Simply type the UNIX System command. For example, type:

```
VP/ix Z:\> who
```

The UNIX System `who` command produces a list of the users currently logged into your system, and the DOS prompt is redisplayed.

All UNIX System commands run from DOS must be issued in the UNIX System format. That is, options on the command line must be preceded by a dash (`-`) rather than the DOS convention slash (`/`) and path names must use the slash (`/`) as a separator rather than the DOS separator character, backslash (`\`). For example, to obtain a long UNIX-style listing of the directory `/usr/lib` in the UNIX System file system, type:

```
VP/ix Z:\> ls -l /usr/lib
```

Note that a UNIX System command run in the VP/ix Environment is *not* able to access files stored on DOS drives A, B, and C. Only files located on the Z drive, i.e., in the standard UNIX System file system, can be used as arguments.

Using Pipes With Both UNIX System and DOS Commands

When you run a DOS command at the UNIX System shell prompt, you can pipe its output to a UNIX System command; you can also pipe the output of a UNIX System command into a DOS command. For example, if you type the following from the UNIX System shell prompt:

```
$ dir | wc
```

the DOS `dir` command runs and its output is piped to the UNIX System `wc` command. (The `wc` command counts the number of lines, words, and characters output by `dir` and displays the result.) As another example, type:

```
$ cat file1 file2 file3 | more
```

`file1`, `file2`, and `file3` are displayed sequentially by the UNIX System command `cat` and the resulting output is displayed on the terminal one screen at a time by the DOS command `more`. This ability facilitates the integration of the DOS and UNIX System environments. In most cases, you do not have to think about whether you are running a DOS command or a UNIX System command.

CONFIGURING THE VP/ix ENVIRONMENT

The VP/ix Environment is installed with a default configuration. It can be tailored to support a different configuration on a system-wide or individual basis. You may need to tailor your system if you plan to use the diskette drive, a mouse, or if you want to run new programs. The VP/ix Environment is customized by modifying:

- The DOS file named `autoexec.bat` on the C drive.
- The VP/ix configuration file (`vpix.cnf`).

Depending on your requirements, you may modify one or both of these files.

This section contains a brief overview of each file and explains why you would want to tailor your environment. Since VP/ix is a multi-user system, you should consult with your system administrator if you find that you need access to resources other than those assigned to you. Your system administrator can help you tailor your environment so that there are no resource conflicts with other users.

The `autoexec.bat` File

This special DOS file will execute commands and programs that tailor a user's and program's environment. It is a batch file that is located in the `root` directory of the drive from which DOS is booted (started). A batch file is a DOS file that contains one or more commands that are automatically executed sequentially when the file is invoked. `autoexec.bat` is a special batch file because it is invoked immediately when the DOS system is booted. It is used for setting the `path` variable discussed in the section entitled "ADVANCED DOS OPERATIONS." The `autoexec.bat` file allows you to execute programs automatically when you start a VP/ix session. For example, if you want to run WordPerfect* every time you boot the DOS system, your `autoexec.bat` file would look similar to this:

```
cd wp
wp
```

You are placed in the WordPerfect program as soon as DOS has booted, rather than getting the DOS prompt.

You may have a personal `autoexec.bat` file only if you also have a personal copy of the bootable DOS image on your C drive. This consists of all the files and programs necessary to boot DOS, including `autoexec.bat`. The files and programs needed to boot DOS require a considerable amount of disk space. Your system administrator can provide you with your own copy of the bootable DOS image if your system space permits.

The VP/ix Configuration File

The VP/ix configuration file enables you to access hardware devices such as diskette drives, the fixed disk, communication ports, printers, and a mouse pointing device. If you attempt to use a device and the system responds with an error message, your system is probably not configured to allow you access to it.

Each time you access the VP/ix Environment, the VP/ix configuration file is read. If your system administrator has properly installed you as a VP/ix user, you will have your own version of the VP/ix configuration file named `$HOME/vpix/vpix.cnf`. `$HOME` is your login directory. Alternatively, if you are very familiar with the UNIX System shell, you may designate a different path name for the `vpix.cnf` file by setting a shell variable called `VPIXCNF` in your `.profile` to be the name of your VP/ix configuration file.

A typical configuration file looks similar to this:

```

; VP/ix Configuration File
; Lines beginning with ";" are comments

; DOS device or          UNIX device path name or
; VP/ix auxiliary file   VP/ix auxiliary file name

A                /dev/rdisk/f0d9dt
C                $HOME/vpix/C:
COM1             /dev/tty00
LPT1             /bin/lp
MOUSE            /dev/mouse
;LMOUSE          /dev/logi
ROM              /usr/vpix/defaults/rom
EGAROM           /usr/vpix/defaults/romega
CMOS             /usr/vpix/defaults/cmos
BOOTIMAGE        $HOME/vpix/vpix.img
;AUTOFLUSH       20

```

There are two types of entries in this file. A comment line begins with a semicolon (;). When the system reads the configuration file, all lines that begin with a semicolon are ignored. They are placed in the file to assist users by providing information and example

entries. In most cases, all you will need to do to change `vpix.cnf` is to remove the appropriate semicolon.

Entries that do not begin with a semicolon are device assignments. A keyword or letter such as `MOUSE` or `A` is followed by the UNIX System device name. A device name is the location of the software that is required to run the specific device, such as the diskette drive or the mouse.

For example, to make a 360 KB diskette drive available, remove the semicolon on this line with a UNIX System editor at the UNIX System prompt:

```
;A    /dev/rdisk/f0d9dt
```

The next time you enter the VP/ix Environment, your new configuration file will be in effect. For more information about configuring your VP/ix Environment, refer to “VP/ix Environment Maintenance Procedures” in this guide. To understand the naming conventions used to access each device (such as `/dev/rdisk/f0d9dt`), consult the “INTERACTIVE UNIX Operating System Maintenance Procedures” in the *INTERACTIVE UNIX Operating System Guide*.

The secondary display that appears on the right when you use the VP/ix Interface Menu tells you which devices you have available, and which UNIX System files they are mapped to.

```
VP/ix Version 1.2
(ESC)ape menu
(F)loppy release
(P)rinter flush
(Q)uit VP/ix
(R)eset VP/ix
(S)ound OFF
(C)lose serial
(E)nter shell
```

Device	Assignment
A:	/dev/rdisk/f0q15dt
C:	/usr/vpix/defaults/C:
Terminal	Assignment
AT386	/dev/console

If the display says that the device is `released`, then you will have access to it only if it is not being currently accessed by another user. If no entry for a device appears, this indicates that there is no valid, uncommented entry for the device in your VP/ix Configuration File.



Appendix A: DOS AND VP/ix COMMAND SUMMARY

DOS and VP/ix Command Summary			
<i>Command</i>	<i>Action</i>	<i>Example</i>	<i>Equivalent UNIX Command</i>
cd	Change from one directory to another	cd \tony\memos	cd /tony/memos
chkdsk	Scan a diskette for errors	chkdsk a:	
copy	Copy one or more files	copy a:elm z:oak	cp elm oak
del	Delete one or more files	del a:tree	rm -r tree
dir	Display the contents of the indicated directory	dir \joe	ls -l /joe
diskcopy	Copy one disk to another	diskcopy a: b:	
erase	Delete one or more files	erase a:tree	rm -r tree
find	Search for a specified string of text in a file or files	find "Li" file.1	grep "Li" file.1
format	Prepare a diskette for use by DOS	format a:	format /dev/rdisk/f0d9dt
mkdir	Create a directory	mkdir letters	mkdir letters
more	Send output to the display one screen full at a time	more <file.1	pg file.1
path	Set the DOS search path	path \deb\games	PATH=/deb/games
rename	Give a file a new name	rename oak elm	mv oak elm

DOS and VP/ix Command Summary (continued)			
<i>Command</i>	<i>Action</i>	<i>Example</i>	<i>Equivalent UNIX Command</i>
<code>rmdir</code>	Delete a directory	<code>rmdir files.dir</code>	<code>rmdir files.dir</code>
<code>sort</code>	Sort and then write the sorted data	<code>sort file.1</code>	<code>sort file.1 -o file.1</code>
<code>type</code>	Display a file on the screen	<code>type elm</code>	<code>cat elm</code>
<code>copy</code> <code>/d</code> <code>/u</code>	Used with options to convert DOS-style file formats to UNIX-style file formats and vice versa	<code>copy /d u.txt d.txt</code> <code>copy /u d.txt u.txt</code>	<code>lef u.txt d.txt</code> <code>lef d.txt u.txt</code>
<code>dospath</code>	Convert to DOS-style file separators and switch characters	<code>dospath</code>	
<code>lef</code>	Convert DOS files to UNIX files or UNIX files to DOS files	<code>lef elm elm.unx</code>	<code>lef elm elm.unx</code>
<code>unixpath</code>	Convert to UNIX-style file separators and switch characters	<code>unixpath</code>	
<code>xdir</code>	Display contents of the indicated directory, including file access UNIX permissions and real file names	<code>xdir z:\usr\amy</code>	<code>ls -l /usr/amy</code>

Appendix B: DOS UNDER THE VP/ix ENVIRONMENT

This appendix is intended for experienced DOS users. It summarizes the major differences between DOS under VP/ix and native DOS. The major headings in this appendix generally parallel the sections of the primer; if you need more information, refer to the appropriate section.

Understanding Disks and Drives

VP/ix Drive Assignments.

The default drive name assignments in the VP/ix Environment are shown below:

<i>Drive Name</i>	<i>Device</i>
A	the first diskette drive
B	the second diskette drive
C	the bootable DOS image on the fixed disk drive
Z	the UNIX file system on the fixed disk drive

The default drive is Z. The C drive should only be used to alter `autoexec.bat` or to add DOS drivers. It is only writable by you if your system administrator gave you a private copy of the C drive when you were installed as a VP/ix user. Drive assignments and the default drive assignment can be altered. Consult your system administrator for more information.

On a VP/ix system the diskette drive is referred to as drive A. Not all VP/ix users have access to drive A. To gain access, you must revise the VP/ix configuration file. To learn how to do this, consult your system administrator or read the section entitled “CONFIGURING THE VP/ix ENVIRONMENT.”

Only one user can use the diskette drive at a time. When you exit the VP/ix Environment, the diskette drive is automatically released. If you are finished with the diskette drive, but want to remain in the VP/ix Environment, release the diskette drive by invoking the VP/ix Interface Menu (using **SYS-REQ**) and selecting the release diskette drive option (F).

Using the z Drive.

In the VP/ix Environment, the multi-user file system is located on the z drive. This is the UNIX file system and it is located on the fixed disk. This is the same file system you access when you log in to the INTERACTIVE UNIX Operating System.

Using DOS Commands

DOS and UNIX System Command Differences.

The UNIX Operating System is case sensitive, which means that it distinguishes between uppercase and lowercase letters; DOS is not. When using a UNIX System command, you may *not* use either uppercase or lowercase characters; you must type the command in the correct case.

On DOS, you can sometimes abbreviate command names to two or three letters; this is *not* true of UNIX System commands.

DOS command options begin with a slash (/); UNIX System command options begin with a dash (-).

UNIX System Commands.

Certain UNIX System programs that are not part of the VP/ix Environment, such as `cpio` and `tar`, use different strategies for storing data on diskettes. Disks created using these programs cannot be used by DOS.

Using and Naming Files

Acceptable File Names for DOS and the UNIX System.

When accessing the UNIX System file system through the DOS Z drive, UNIX System file names that do not follow DOS naming conventions will be changed to unique DOS names, usually containing one or more tildes (~). The `xdir` command can be used to determine the real UNIX System name of the file.

- ☛ To make a UNIX System file name identical in both the UNIX System and DOS, give it a name that is acceptable to DOS and contains *only* lowercase letters.

File Systems and Directories

Path Names.

DOS path names are separated by backslashes (\); UNIX System path names are separated by slashes (/).

DOS path names frequently include a drive name (A: or B:, for example); UNIX System path names do not use drive names.

The file naming conventions above are also applicable to directory names. If VP/ix is started from a directory with a name that does not map exactly to a DOS directory name, the user will be logged into the default drive's root directory.

Basic DOS Commands

Some common DOS commands must be used differently in the VP/ix Environment. These differences are listed below.

`format`

Although the `format` command is sometimes used to format fixed disks in standard DOS systems, it should *not* be used for that purpose in the VP/ix Environment.

`diskcopy`

The `diskcopy` command can be used on diskettes only. It is *not* intended for use on fixed disks.

`chkdsk`

You cannot use the `chkdsk` command on the Z drive since Z is not a DOS file system. The Z drive contains the UNIX System file system.

`copy`

You can convert text between DOS and UNIX System formats using the `copy` command with the `/d` or `/u` options. For example:

```
copy unix.doc dos.doc /d
```

is used to copy the `unix.doc` file to `dos.doc`, converting the UNIX System file format to DOS file format.

To copy a DOS file to UNIX System format, use the `/u` option of the `copy` command. For example:

```
copy dos.doc unix.doc /u
```

is used to copy the `dos.doc` file to `unix.doc`, converting the DOS file format to UNIX System file format.

Using DOS and UNIX System Commands Together

Running DOS Commands From the UNIX System Prompt.

All DOS commands, even when run from a UNIX System prompt, follow standard DOS conventions. That is, you must use the slash character (/) to indicate options and the backslash (\) as the separator in path names.

Users running DOS commands directly from the UNIX System shell should note that certain special characters (such as the DOS wildcard characters * and ?, the DOS file separator \, and the DOS switch character /) are interpreted by the UNIX System shell. In these cases the system may behave in an unexpected manner. To avoid this problem, enclose arguments to DOS commands in single quotes (') when running them from the UNIX System shell.

Running DOS Applications From the UNIX System Prompt.

To run a DOS application at the UNIX System prompt, the application *must* be installed on the fixed disk; you cannot run DOS applications that are stored on diskettes from the UNIX System prompt.

Running UNIX System Commands From the DOS Prompt.

A UNIX System command run in the VP/ix Environment is *not* able to access files stored on DOS drives A, B, and C. Only files located on the Z drive, i.e., in the standard UNIX System file system, can be used as arguments. Also, the UNIX System option and path separator characters (- and /, respectively) must be used.

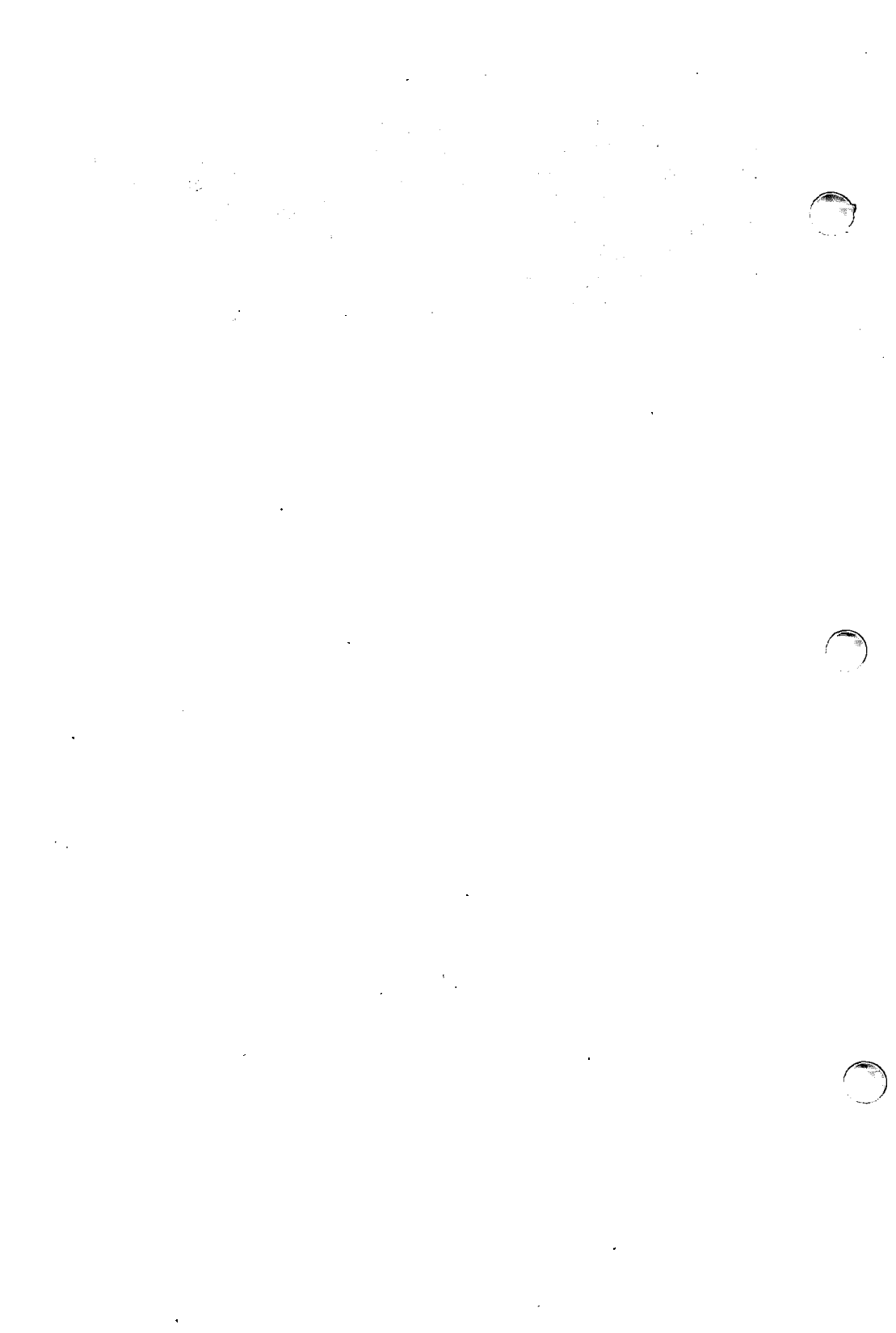
Configuring the VP/ix Environment

The DOS File `autoexec.bat`.

When you enter the VP/ix Environment, DOS searches the disk for the `autoexec.bat` file. It must be located in the `root` directory of the fixed disk or in the bootable image of the DOS file used to boot the VP/ix Environment. The `autoexec.bat` file is *not* optional under the VP/ix Environment. If you have your own copy of the bootable DOS image, it must contain an `autoexec.bat` file with certain required entries. For more information, see the “VP/ix Environment Maintenance Procedures” in this guide or consult your system administrator.

The VP/ix Configuration File (vpix.cnf).

To use a diskette drive while in the VP/ix Environment, your system administrator must have given you access to the diskette drive. If you cannot access the diskette drive, check the VP/ix configuration file found in your home directory (`$HOME/vpix/vpix.cnf`) on the Z drive for an A drive entry. If there is an entry for A, be sure there is not a semicolon (;) in front of it. If there is no entry for A, it must be added; consult your system administrator.



Appendix C: USING ASCII TERMINALS

VP/ix users who are using an ASCII terminal should be aware of the following:

- There are three basic differences between an ASCII terminal and a system console:
 1. The keyboards.
 2. The characters that the terminal is capable of displaying on the screen.
 3. The need for special key sequences for refreshing a garbled screen (which cannot happen on a console) and for “rocking” between the 1st through the 24th and 2nd through 25th lines of the screen (since the console has 25 lines and most ASCII terminals have only 24).
- Only a limited set of the hundreds of different models of ASCII terminals is explicitly supported by VP/ix. These include the Wyse* 60, the AT&T 605, the DEC* VT52*, and the DEC VT100*. Support for these terminals is custom-tailored, which means that all of the keys on these terminals’ keyboards are used in simulating the console keyboard, and all of the output characters that it is possible to display on each terminal are used in simulating console output.
- For all other terminal models, one generic set of keyboard mappings is used. This is shown in the next table. Special sequences for rocking and refreshing the screen are also included. Furthermore, all non-ASCII screen output characters are represented by a dot (.), regardless of whether the terminal has the capacity to display them. In addition, screen attributes such as blinking or reverse video may not be properly displayed. Note that it is possible to tailor support for your terminal model even if it is not explicitly supported by VP/ix, as described previously. Consult your system administrator, or read Appendix A of “VP/ix Environment Maintenance Procedures” in this guide for details.
- Not all DOS applications can be run on an ASCII terminal. Applications that can run on a monochrome display will operate on an ASCII terminal; those that require EGA or CGA support will not.

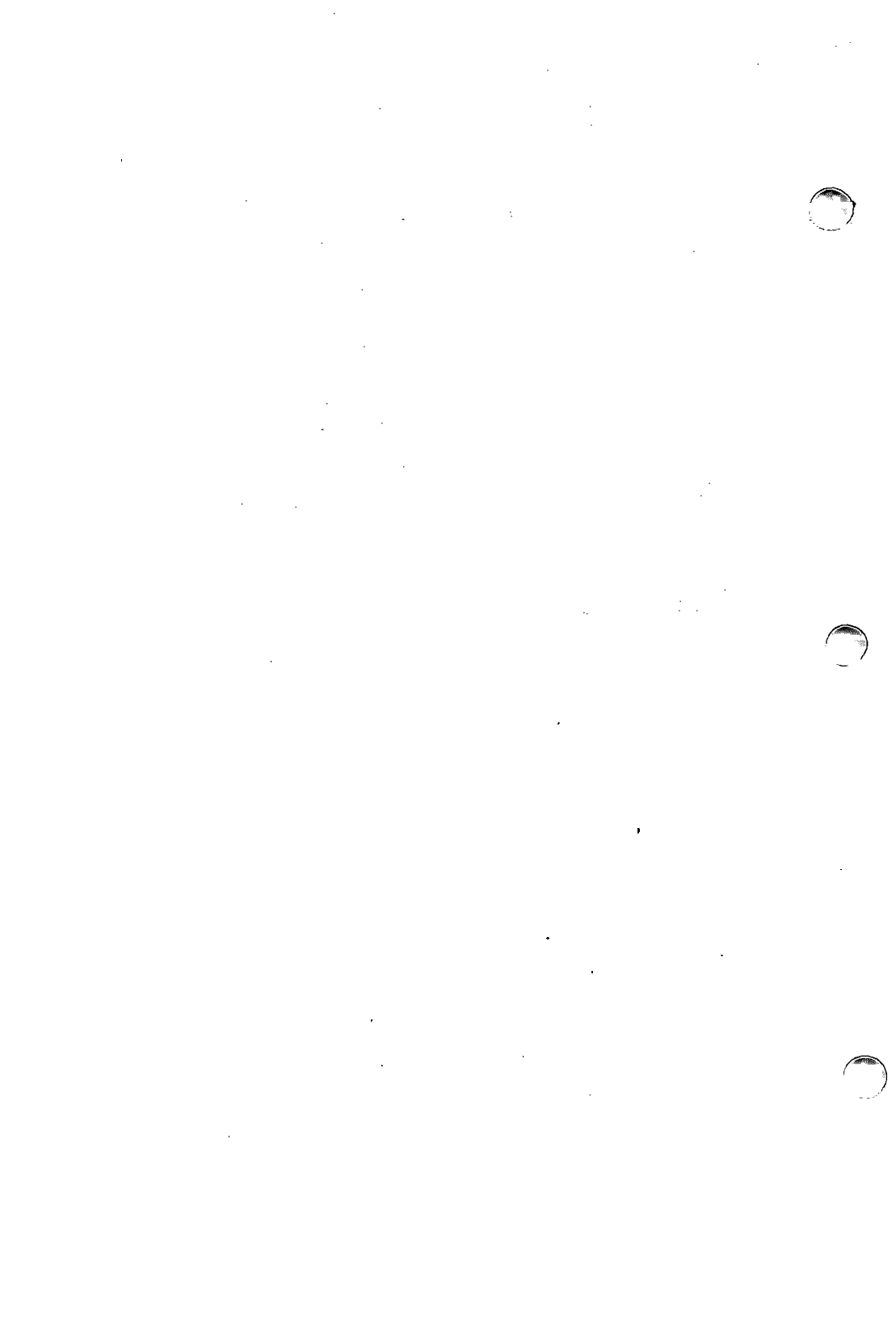
- A mouse controller cannot be used from an ASCII terminal. Applications should be configured accordingly.
- On an ASCII terminal, when running DOS commands at the UNIX System prompt or UNIX System commands at the DOS prompt, the system will display the message:

[Press Enter to Continue]

after processing the command. Press **RETURN** to return to the UNIX System or DOS prompt.

Generic ASCII Terminal Key Mapping	
<i>PC Key</i>	<i>Equivalent ASCII Terminal Key Sequence</i>
F 1	ESC 1
F 2	ESC 2
F 3	ESC 3
F 4	ESC 4
F 5	ESC 5
F 6	ESC 6
F 7	ESC 7
F 8	ESC 8
F 9	ESC 9
F 10	ESC 0
SYS-REQ	ESC s
↑	ESC ^
↓	ESC v
→	ESC >
←	ESC <
depress ALT	ESC a
release ALT	ESC A
depress CTRL	ESC c
release CTRL	ESC C
depress left SHIFT	ESC l
release left SHIFT	ESC L
depress right SHIFT	ESC r
release right SHIFT	ESC R
CAPS LOCK	ESC K
NUM LOCK	ESC N
SCROLL LOCK	ESC S
HOME	ESC h

Generic ASCII Terminal Key Mapping	
<i>PC Key</i>	<i>Equivalent ASCII Terminal Key Sequence</i>
PG UP	ESC u
PG DN	ESC n
END	ESC e
DEL	ESC d
INS	ESC i
PRTSC	ESC p
BREAK	ESC b
< - TAB	ESC TAB
- on numeric keypad	ESC -
+ on numeric keypad	ESC +
blank (under 5) on numeric keypad	ESC SPACEBAR
* under PRTSC key	ESC *
rocker key	ESC /
refresh screen	ESC ,



GLOSSARY

access permissions

See *file access permissions*.

append

To add text to the end of an existing file.

application

A program that is written to perform a specific task, such as word processing, database manipulation, or accounting.

argument

A string of text that accompanies a command and gives the computer additional information to modify the result of the command.

batch file

A file containing DOS commands, named with the .BAT extension, used to run the commands in the order listed.

boot

To load an operating system or a standalone program into memory and execute it.

byte

A unit for storing data in a computer.

case insensitive

Does not distinguish between uppercase letters and lowercase letters.

case sensitive

Distinguishes between uppercase letters and lowercase letters.

command

An instruction the user gives to the computer. The command is interpreted by the computer, which then runs the program that performs the task requested by the command. See also *argument* and *option*.

command line

The complete instruction the user gives to the computer, including the command name, options, arguments, and pipes.

console

The main keyboard and monitor on a computer.

current directory

The directory in which you are located. Its path name is displayed on the screen when you use the `cd` command.

default

The alternative chosen by the system when no choice is specified by the user.

destination drive

See *target drive*.

device specifier

The name of a disk drive.

directory

A special type of file that contains other files and/or directories. A typical directory contains related documents, such as memoranda or monthly sales reports.

disk drive

See *drive*.

diskette

A device used for storing data (also known as a “floppy disk”). Data is stored or read on a diskette by inserting the diskette into a disk drive, then issuing the appropriate command.

drive

A device on which a disk can be placed to be used for storing data. Also called a disk drive.

extension

See *file name extension*.

external command

Any file with a file name extension of `com`, `exe`, or `bat`.

file

A collection of information stored on the computer. For example, a file could be a list of phone numbers, a memorandum, or a report.

file access permissions

In the VP/ix Environment, file access permissions determine who can read or write a file or directory stored on the Z drive. They are indicated by the letters *r* (read) and *w* (write). The permissions can be viewed by using the `xdir` command to list the contents of a directory. Permissions can be changed using the UNIX System `chmod` command.

file name extension

That portion of a full DOS file name that follows the period. It can be one to three characters in length.

file system

A collection of individual files and directories that are stored on a portion of a disk.

filter

A program that accepts its input from one source, such as the standard input, performs the appointed task on the data, and then writes the results to the standard output without changing the input file in any way.

fixed disk

A medium for storing data. The fixed disk is permanently attached to your computer system. Also called a hard disk.

flag

An option to a command. See *option*.

full path name

The path name of a file that completely describes the location of the file in the system. It is the sequence of directories from the `root` directory to the file or directory you wish to reference. In DOS, it consists of a backslash (`\`), followed by one or more directory names, and a file name, each separated by a backslash. UNIX System path names begin with a slash (`/`), and the components are separated by slashes.

graphics application

An application that puts the console into “graphics mode,” where dots (pixels) on the screen can be turned on or off individually. This is in contrast to “text mode,” where only entire characters on the screen can

be manipulated. Graphics applications can only run on consoles that are equipped with special monitors, for example, EGAs or CGAs.

group

A group is a collection of users. Permission for a group refers to the second of three permissions for a file or directory. These apply to members of the owner's group only.

hard disk

See *fixed disk*.

hardware

The physical components of a computer. Examples include the keyboard, the screen (sometimes called the display or the monitor), and the printer.

hierarchical file system

A directory structure that is arranged in a ranked series, with a single master directory at the top level and additional levels of directories or files defined beneath it.

home directory

The directory in which you are placed when you initially log into a computer system. It contains the "master list" or index of the information you are personally storing on the computer. Most multi-user systems automatically place you in your home directory when you log in.

install

To make software available on your computer. For example, an operating system must be installed on your computer before you can use it.

internal command

The simplest, most commonly used commands on a DOS system. These commands are part of a large file named `command.com`.

log in

To type your login name and password onto your computer or terminal, indicating that you are ready to gain access to your information.

login account

The information stored in a computer that provides

authorization for a person to use that computer's resources. The computer's system administrator usually sets up login accounts.

login directory

See *home directory*.

log out

To terminate your access to the system. On UNIX Systems, this is accomplished by holding down the **CTRL** key while simultaneously typing d.

multi-user

Capable of supporting more than one user at a time. On a multi-user computer system, each user has his or her own terminal plugged into the computer and all can share its data and resources at the same time.

operating system

The operating system is part of the system software. It is a program that tells the computer how to manage its operations, process and execute the user's requests, and run application software.

option

An optional argument that is available to modify the results of a specific command. Options are preceded by a slash (/) on DOS. On UNIX Systems, options are preceded by a dash (-).

other

The third of three permissions for a file or directory, which applies to everyone on the system who is neither the owner nor in the owner's group.

overwrite

To perform an operation on a file that deletes its contents and replaces it with the output of the operation. For example, if you `sort` a file and use the name of an already existing file as its destination, the contents of the original file is replaced with the contents of the new file.

owner

The user who creates a file and has control over the file's permissions.

path

The list of directories through which the DOS or UNIX System command processors look to find programs to be executed. This list is made available to the appropriate command processor through a specific environment variable. Also used as a shortened term for *path name*.

path name

The sequence of directory names and (optionally) a file name, separated by backslashes (\) on DOS or by slashes (/) on UNIX Systems, that describes the location of a file or directory.

pattern matching character

See *wildcard*.

permissions

See *file access permissions*.

pipeline

To take the output of one program and direct it to be the input to another program.

pipes

Connections between two or more commands, indicated by a vertical bar symbol (|).

program

A set of instructions for a specific set of tasks, written in a machine or programming language, that tells the computer how to perform the tasks.

prompt

A symbol that displays on the screen to indicate that the system is ready to receive your commands. You may execute a command or run an application when the prompt is displayed.

read permission

Allows a user to view the contents of the file or copy it to another area. Read permission by itself does not allow a user to change a file.

reboot

To reload an operating system or a standalone program into memory and execute it.

redirection

To change the destination of the output or the input of a command.

relative path name

The path name of a file or directory, omitting the `root` directory and directory names up to the current directory.

root directory

The top-level directory, designated by a backslash (`\`) on DOS and by a slash (`/`) on UNIX Systems. All other directories in the system are under the `root` directory.

serial communications port

Part of a computer that receives data from or transmits data to an external remote device. On DOS the serial ports are usually given the names `COM1:` and `COM2:`.

single-user

Capable of supporting only one user at a time. On a single-user computer operating system, no one else can simultaneously share the computer's data or resources. Most small personal computers are single-user systems.

software

The instructions that make a computer perform its functions. Software is divided into two main groups: applications and the operating system. Application software performs specific tasks, such as word processing, spreadsheets, educational programs, and games. The operating system software tells the computer how to run application software.

source drive

The drive from which you are taking information. For example, when copying a diskette, the diskette drive that contains the diskette to be copied is the source drive. See also *target drive*.

spooler

A program that gathers up data and enqueues it for processing by another program or a piece of hardware (such as a printer).

standard input

Information coming from the keyboard, unless otherwise specified by the user. The user can specify that information come from a file, a device, or a pipe.

standard output

The destination of a program's data, considered to be the terminal screen, unless otherwise specified by the user. The user can specify that information be written to a file, a device, or a pipe.

string

A series of characters surrounded by spaces.

switch

An option provided to a command. See *option*.

system administrator

The person responsible for maintaining a computer system.

target drive

The drive to which you will transfer information. For example, when copying a diskette, the diskette drive that contains the blank diskette is the target drive.

variable

A string that has special meaning to the operating system. It is usually assigned a value.

VP/ix Interface Menu

The VP/ix resource menu invoked by using **SYS-REQ**. This menu is used to control system resources available to each VP/ix user.

wildcard

A wildcard is a character that is used to match any character or string of characters in a file or directory name. It is a type of shortcut.

write permission

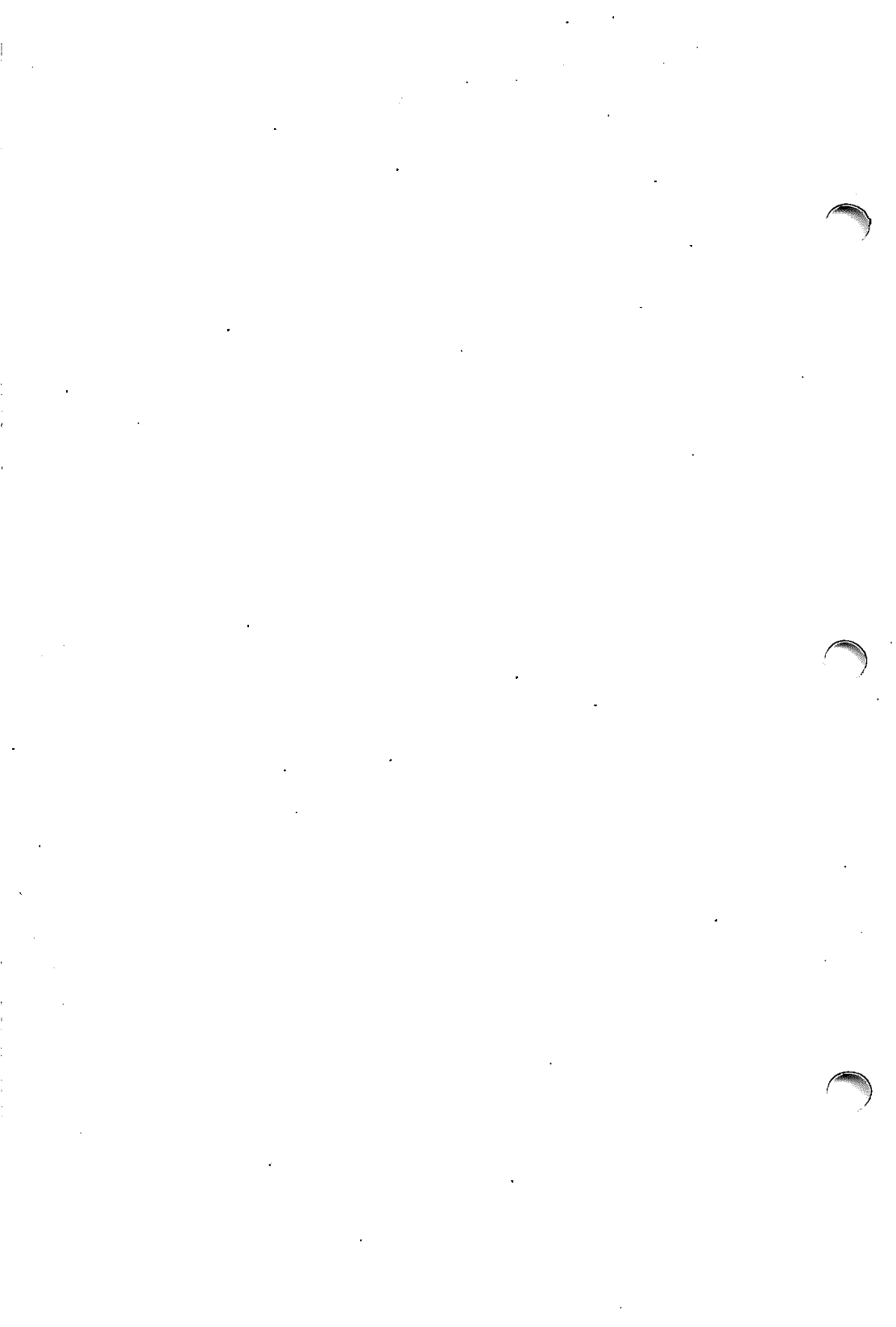
Allows a user to make changes to a file. Write permission on a directory allows a user to create and delete files in that directory.

INDEX

- / directory 26
- / symbol 15, 19
- < symbol 40
- > symbol 59
- * wildcard 23, 24
- A drive 11
- accessing a diskette drive 12
- accessing VP/ix 4
- ALT SYS-REQ m 5
- appending one file to another 59
- application, running DOS from UNIX 62
- argument 14, 16, 18
- ASCII terminal 2, 5, 77
- ASCII terminal, using with VP/ix 77
- autoexec.bat 65
- AUTOFLUSH 8
- AUX 22
- available devices 67
- B drive 11
- BACKSPACE 15
- bat 55
- batch file 57, 65
- C drive 11
- case insensitive 15
- case sensitive 15
- cd 32
- changing conventions 53
- changing directories 32
- chdir 32
- checking a diskette for errors 45
- chkdsk 45
- Close serial 7
- com 55
- COM1 22
- COM2 22
- command 16, 35
- command abbreviations 16
- command, DOS 35
- command line 14
- command, summary of DOS 69
- command, summary of VP/ix 69
- command syntax 15
- commands, DOS 14
- commands, external 55
- commands, internal 55
- commands, running DOS from UNIX 61
- commands, running UNIX from DOS 63
- commands, stopping 20
- commands, using DOS with UNIX 61
- commands, using pipes with UNIX and DOS 63
- CON 22
- configuring the VP/ix Environment 65
- conventions, changing 53
- conventions, UNIX System and DOS 53
- copy 35, 52
- copying a diskette 44
- copying a file 35
- copying files from one format to the other 52
- creating a directory 31
- CTRL c 20
- CTRL q 20, 39
- CTRL s 20, 39
- CTRL-ALT-DEL 7
- del 37
- deleting a file 37
- determining UNIX System file information 50
- device-specifier 28
- dir 16, 20
- directory 3, 26
- directory, changing 32
- directory, creating 31
- directory, deleting 33
- directory, home 3
- directory, renaming 34
- directory, root 26
- disk 11
- disk drive 11, 18
- disk, fixed 11
- diskcopy 44
- diskette 11
- diskette, checking for errors 45
- diskette, copying 44
- diskette drive, accessing 12
- diskette drive, releasing 6, 12
- diskette, formatting 42
- displaying a file 39
- displaying a file one screen at a time 40
- DOS and UNIX System conventions 53
- DOS applications, running 8
- DOS commands 14
- dospath 53
- dot (.) 30
- dot dot (..) 30
- drive, A 11
- drive, activating 12
- drive, B 11
- drive, C 11
- drive, changing 12
- drive, destination 18
- drive, disk 11, 18
- drive, diskette 11
- drive, fixed disk 11
- drive, source 18
- drive, target 18
- drive, Z 11, 12, 22, 45, 48
- ENTER 14
- Enter shell 8
- exe 55
- exiting, VP/ix Environment 10
- file 21
- file access permissions 49
- file, appending 59
- file, batch 57, 65
- file, copying 35
- file, copying from one format to the other 52
- file, deleting 37
- file, displaying 39
- file, displaying one screen at a time 40
- file, DOS 21
- file, incompatible formats 47
- file name 18
- file name extension 18, 21, 55

- file name, invalid 22
- file, naming 21
- file, overwriting 59
- file, printing 8
- file, renaming 38
- file, sorting 42
- file system 26
- file system, differences between DOS and UNIX System 47
- file system, hierarchical 26
- file, VP/ix configuration 66
- filter, using 59
- find 41
- fixed disk 11
- flag 14
- Floppy release 6
- format 42
- formatting a diskette 42
- full path name 28
- generic ASCII terminal key mapping 78
- group 49
- input, redirecting 58
- input, standard 58
- invalid file name 22
- lef 51
- loading DOS applications 55
- logging in 2
- logging out 10
- login account 2
- LPT1 22
- LPT2 22
- LPT3 22
- menu, VP/ix interface 4
- menu with graphics application 6
- mkdir 31
- more 40
- MULTIBUS 8
- naming a DOS file 21
- naming a UNIX System file 22
- NUL 22
- option 14, 16, 19
- options, conventions used in DOS and UNIX 53
- other 49
- output, redirecting 58
- output, standard 58
- password 3
- path 56
- path name 18, 28
- path name, full 28
- path name, relative 28, 29
- path name, restrictions 29
- path name shortcut 30
- path names, conventions used in DOS and UNIX 53
- path, VP/ix 57
- pattern matching character 23
- permission 49
- permission, read 49
- permission, write 49
- pipe symbol 60
- pipe, using 60
- pipe, using with UNIX and DOS commands 63
- pipeline 60
- Printer flush 6
- printing a file 6, 8
- PRN 22
- program 14
- prompt 3
- protection modes 49
- question mark wildcard 23
- Quit VP/ix 7
- rd 33
- read permission 49
- redirecting input and output 58
- redirection of input 40
- relative path name 28, 29
- removing a directory 33
- rename 38
- renaming a directory 34
- renaming a file 38
- Reset VP/ix 7
- RETURN 14
- rmdir 33
- root directory 26
- running DOS applications 8
- VP/ix, accessing 4
- VP/ix, exiting 7, 10
- VP/ix Interface Menu 4
- VP/ix Interface Menu, invoking 4
- VP/ix, invoking a UNIX shell from 8
- VP/ix, printing from 6, 8
- VP/ix, resetting 7
- scrolling, resuming 20, 39
- scrolling, stopping 20, 39
- search path, setting 56
- searching for text 41
- secondary display 67
- serial port, releasing 7
- setting a search path 56
- shell, invoking from VP/ix 62
- shortcut, path name 30
- sort 42
- sorting a file 42
- Sound OFF 7
- Sound ON 7
- sound, turning off 7
- sound, turning on 7
- standard input 58
- standard output 58
- string 15, 18
- string, searching for 41
- switch 14
- SYS-REQ 4, 5, 12
- system console 77
- terminal, ASCII 2, 5, 77
- terminal key mapping, ASCII 78
- text file, converting formats 51
- text file, incompatible formats 51
- text, searching for 41
- type 39
- UNIX shell 8
- UNIX shell, invoking from VP/ix 8
- UNIX System and DOS conventions 53
- UNIX System file information, determining 50
- unixpath 53
- user 49

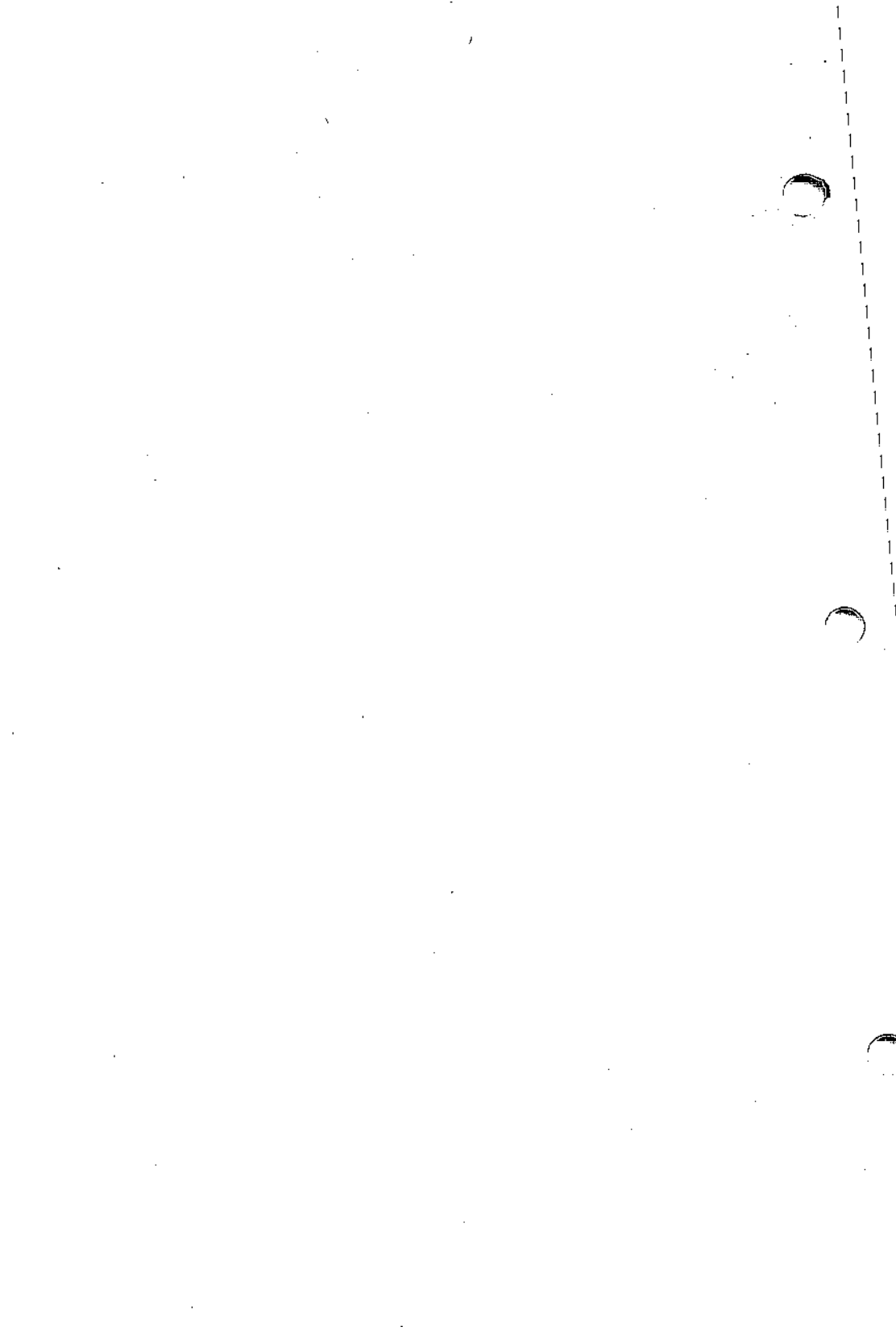
user categories 49
user ID 2
user identification 2
vpix 4
VP/ix configuration file 66
VP/ix Environment configuration 65
VP/ix Interface Menu 62
VP/ix Interface Menu, escaping 6
wildcard 23, 61
write permission 49
xdir 50
Z drive 4, 11, 12, 22, 45, 48



VP/ix Environment Installation Instructions

CONTENTS

1. OVERVIEW	1
2. INSTALLING THE VP/ix ENVIRONMENT	1
3. CONFIGURING LOGIN ACCOUNTS FOR THE VP/ix ENVIRONMENT	4
4. LIST OF DISTRIBUTION FILES	7



VP/ix* Environment Installation Instructions

1. OVERVIEW

The VP/ix Environment is installed on your fixed disk using the `sysadm` utility. It is comprised of three subsets:

- The *VP/ix Core subset* contains the base VP/ix Environment. It requires about 1 MB of disk space and requires that the INTERACTIVE UNIX* System Core subset already be installed. It is distributed on one diskette.
- The *MS-DOS* subset* contains the MS-DOS operating system and associated files. It is distributed on one diskette and requires about 0.7 MB of disk space. If you are installing an upgraded version of VP/ix onto your existing VP/ix Environment, you do not need to install this subset.
- The *VP/ix Configuration subset* is optional and may be installed by those users who plan to customize the VP/ix Environment (i.e., to support third-party hardware devices). It takes about 0.5 MB of disk space and requires that the INTERACTIVE UNIX System Kernel Configuration subset already be installed. It is distributed on one diskette.

This document assumes that you have read and understood the following INTERACTIVE UNIX System documents:

- “INTERACTIVE UNIX Operating System Installation Instructions”
- “INTERACTIVE UNIX Operating System Maintenance Procedures”
- “INTERACTIVE UNIX Operating System Primer”

These documents were delivered with your INTERACTIVE UNIX Operating System.

2. INSTALLING THE VP/ix ENVIRONMENT

1. To begin the installation, use the System Administration command, `sysadm`, or log in as `sysadm` to access the Main menu. Your screen will look similar to this:

SYSTEM ADMINISTRATION

1	diskmgmt	disk management menu
2	filemgmt	file management menu
3	machinmgmt	machine management menu
4	packagemgmt	package management menu
5	softwaremgmt	software management menu
6	syssetup	system setup menu
7	ttymgmt	tty management menu
8	usermgmt	user management menu

Enter a number, a name, the initial part of a name, or
? or <number>? for HELP, q to QUIT:

2. Type 5 to access the Software Management menu. Your screen will then look similar to this:

SOFTWARE MANAGEMENT

1	installpkg	install new software package onto built-in disk
2	listpkg	list packages already installed
3	removepkg	remove previously installed package from built-in disk
4	runpkg	run software package without installing it

Enter a number, a name, the initial part of a name, or
? or <number>? for HELP, ^ to GO BACK, q to QUIT:

3. Select option 1, `installpkg`. The system prompts you to insert the VP/ix diskette into the diskette drive. The screen will look similar to this:

Insert the removable medium for the package you want to
install into the diskette drive.
Press <RETURN> when ready. Type q to quit.

4. Insert the *Core* diskette for the VP/ix Environment into the diskette drive. The system asks you to confirm that this is the package you want to install. Press **ENTER** to start the installation process. (The file names listed below may appear in a different order during your installation.)

```

Install the VP/ix Environment Core- Version 1.2.0 Package? (y):
Installing the VP/ix Environment Core- Version 1.2.0
Copyright (c) 1988 Interactive Systems Corp. and Phoenix
Technologies Ltd.
All Rights Reserved
The following files are being installed:
/usr/options/vp.name
/usr/vpix/dosbin/append.exe
/usr/vpix/dosbin/assign.com
/usr/vpix/dosbin/attrib.exe
.
.
Floppy diskette number 1 is complete
Installation of the VP/ix Environment Core- Version 1.2.0
is complete
You may now remove the medium from the diskette drive.

```

5. Note: In general, the *MS-DOS subset* does not change between releases of VP/ix. If you are upgrading from an earlier version of VP/ix, you may skip this step. Otherwise, insert the diskette labeled *DOS* and press **ENTER**.

```

Install the VP/ix Environment Multi-User DOS-Version 1.2.0? (y)
Copyright (c) 1988 Interactive Systems Corp. and Phoenix
Technologies Ltd.
Copyright 1981-1988 Microsoft Corporation.
.
.
Floppy diskette Number 1 is complete.
Installation of VP/ix Environment Multi-User DOS-Version 1.2.0
is complete.

```

6. Insert the *Configuration* diskette and press **ENTER** to continue the installation.

3. CONFIGURING LOGIN ACCOUNTS FOR THE VP/ix ENVIRONMENT

Before a user can access the VP/ix facilities, VP/ix privileges must be added to each individual login account. This section explains how to configure a login account for use with the VP/ix Environment; it assumes that you have already established user login accounts as described in section 3.8 of the “INTERACTIVE UNIX Operating System Installation Instructions.” Refer to “INTERACTIVE UNIX Operating System Maintenance Procedures” if you need to learn how to add a new user login account.

1. To add VP/ix privileges to user accounts, use the System Administration command, `sysadm`, or log in as `sysadm` to access the Main menu. Your screen will look similar to this:

```

                                SYSTEM ADMINISTRATION

1 diskmgmt      disk management menu
2 filemgmt     file management menu
3 machinmgmt   machine management menu
4 packagemgmt  package management menu
5 softwaregmt  software management menu
6 syssetup     system setup menu
7 ttygmt      tty management menu
8 usermgmt     user management menu

Enter a number, a name, the initial part of a name, or
? or <number>? for HELP, q to QUIT:

```

2. Select option 4 to access the Package Management menu. Your screen will then look similar to this:

```

                                PACKAGE MANAGEMENT

1 uucpmgmt     basic networking utilities menu
2 vpximgmt    install users for the VP/ix Environment

Enter a number, a name, the initial part of a name, or
? or <number>? for HELP, ^ to GO BACK, q to QUIT:

```

There may be additional options on your screen, depending on the number of subsets purchased.

3. Select the option named `vpixmgmt` to access the VP/ix Management menu. Your screen will look similar to this:

```

                                VP/ix MANAGEMENT

1 addvpixuser  enable users to use the VP/ix Environment
2 delvpixuser  disable users from using the VP/ix Environment
3 ddainstall   install a DDA device into the VP/ix Environment
4 ddaremove    remove a DDA device from the VP/ix Environment
5 vconfig      change device parameters and reconfigure VP/ix

Enter a number, a name, the initial part of a name, or
? or <number>? for HELP, ^ to GO BACK, q to QUIT:

```

4. Select option 1. The system responds:

```
Do you wish to set up a new VP/ix user? [y, n, ?, q]
```

5. Type `y`. The system then asks for the user's login ID:

```
Enter the User's login ID :
```

6. Enter the user's login ID, `jane`, for example. Next, the system asks whether the user should be given a local copy of the C drive. The C drive is usually the fixed disk drive. By default, VP/ix is booted from the C drive.

```
Should user jane have a private copy
of the C: drive? (Default is "n") [y, n, ?]
```

7. Press **ENTER** to accept the default (`n`). In some cases you may wish to give experienced users their own copies of the C drive. Refer to "VP/ix Environment Maintenance Procedures" for a detailed discussion of this issue. The system displays a message similar to this:

```
User jane can now use the VP/ix Environment
```

8. It then offers you the opportunity to install other VP/ix users:

```
Do you wish to set up a new VP/ix user? [y, n, ?, q]
```

Type `y` to repeat the procedure for another user, or type `n` to quit.

- Remember, you must use this procedure each time you add a new user to your system if the new user is to have access to the VP/ix Environment.

You can reverse the actions described here by choosing option 2, `delvpixuser`, from the VP/ix Management menu in step 3. Note that this procedure will remove all files and directories under

the user's `$HOME/vpix` directory. You should warn the user to save any files he or she may have edited or added under this directory *before* you delete that person as a VP/ix user.

4. LIST OF DISTRIBUTION FILES

The VP/ix Environment Core, MS-DOS, and the VP/ix Configuration subsets are distributed on three diskettes. The contents of the diskettes are copied to the `/usr/vpix` directory on your fixed disk. This section briefly describes the files distributed with the VP/ix Environment.

- `/usr/admin/menu/packagegmt/vpixgmt`
The directory used by the `sysadm` utility to locate new VP/ix Environment management shell scripts.
 - `/usr/admin/menu/packagegmt/vpixgmt/DESC`
A description of the scripts in this directory.
 - `/usr/admin/menu/packagegmt/vpixgmt/addvpixuser`
The script used to install new VP/ix Environment users onto the system.
 - `/usr/admin/menu/packagegmt/vpixgmt/delvpixuser`
The script used to remove access of currently installed VP/ix users to the VP/ix Environment.
- `/usr/vpix`
The directory where all VP/ix files are stored.
 - `/usr/vpix/README`
The list of software changes and known problems associated with this VP/ix release.
 - `/usr/vpix/vpix`
The actual `vpix` command that is linked to `/usr/bin` during installation.
- `/usr/vpix/defaults`
The directory where default user configuration files are stored. These files can be copied to each user's private `vpix` directory for customization.
 - `/usr/vpix/defaults/C:`
The default boot image file from which DOS boots. Contains DOS boot programs such as `io.sys` and `msdos.sys`.

/usr/vpix/defaults/cmos

The default CMOS image file used by the VP/ix Environment to retain CMOS information normally stored in battery backup CMOS RAM.

/usr/vpix/defaults/rom

Retains for the VP/ix Environment the default BIOS normally stored in the computer's ROM.

/usr/vpix/defaults/romega

Retains for the VP/ix Environment the default EGA BIOS normally stored in the Enhanced Graphics Adapter's ROM.

/usr/vpix/defaults/romvga

Retains for the VP/ix Environment the default VGA BIOS normally stored in the Video Graphics Adapter's ROM.

/usr/vpix/defaults/vpix.cnf

The default user configuration file that serves as a template for each user.

• /usr/vpix/etc

The directory containing miscellaneous files and directories for customized VP/ix environments.

• /usr/vpix/etc/vpixdevs

Contains descriptions of nonstandard VP/ix devices, such as those supplied by third-party vendors.

• /usr/vpix/term

A directory containing VP/ix terminal specification files for serial terminal support.

/usr/vpix/term/AT386

The terminal specification file for an AT* console being used as a serial terminal, e.g., during a remote login session.

/usr/vpix/term/605

The terminal specification file for an AT&T 605 PC-compatible terminal.

/usr/vpix/term/generic

The terminal specification file used if none matching the user's terminal is found in this directory.

`/usr/vpix/term/ibm3151`

The terminal specification file for an IBM* 3151 PC-compatible terminal.

`/usr/vpix/term/kt70`

The terminal specification file for a KT-70/PC* terminal from Kimtron Corporation.

`/usr/vpix/term/sys.default`

The VP/ix terminal-independent specification file, used for all terminals.

`/usr/vpix/term/vt100`

The terminal specification file for a DEC* VT100* terminal.

`/usr/vpix/term/vt52`

The terminal specification file for a DEC VT52* terminal.

`/usr/vpix/term/wyse60`

A link to `/usr/vpix/term/wyse`.

`/usr/vpix/term/wyse`

The terminal specification file for a Wyse-60* terminal running in PC mode.

- `/usr/vpix/unixbin`

The directory containing files that allow cross-execution of UNIX System programs from DOS. Note that when VP/ix is installed, several new files are created in this directory (see section 3.2 of “VP/ix Environment Maintenance Procedures” for details).

`/usr/vpix/unixbin/rununix.exe`

A DOS program that allows execution of UNIX System programs from within the VP/ix Environment.

- `/usr/vpix/dosbin`

A directory containing standard MS-DOS 3.3 distribution programs and some VP/ix-specific DOS executable programs.

`/usr/vpix/dosbin/ansi.sys`

The DOS device driver for ANSI terminal emulation.

`/usr/vpix/dosbin/append.exe`

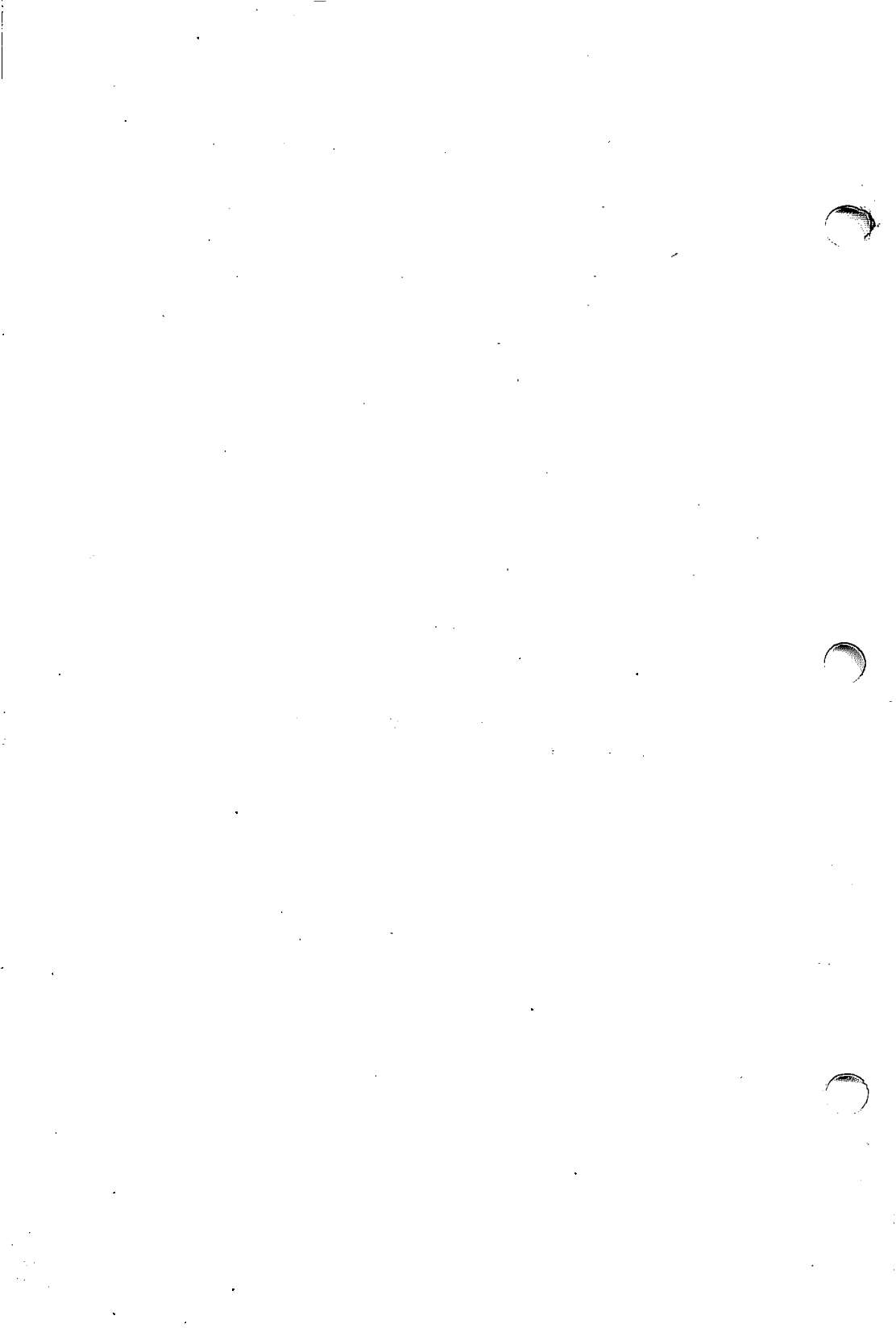
Locates files outside of the current directory with extensions other than `.COM`, `.EXE`, and `.BAT`.

- `/usr/vpix/dosbin/assign.com`
Assigns a drive letter to a different drive.
- `/usr/vpix/dosbin/attrib.exe`
Sets or displays attributes of a file.
- `/usr/vpix/dosbin/backup.com`
Backs up files from a fixed disk to another device.
- `/usr/vpix/dosbin/chkdsk.com`
Scans the directory of the default or designated drive and checks for consistency.
- `/usr/vpix/dosbin/command.com`
The DOS Command Processor that is the equivalent of the UNIX System shell.
- `/usr/vpix/dosbin/comp.com`
Compares the contents of two files.
- `/usr/vpix/dosbin/country.sys`
Default country information file.
- `/usr/vpix/dosbin/debug.com`
The DOS debugger.
- `/usr/vpix/dosbin/dir.bat`
A file that enables the execution of the DOS `dir` command from the UNIX System environment.
- `/usr/vpix/dosbin/diskcomp.com`
Compares the contents of two diskettes.
- `/usr/vpix/dosbin/diskcopy.com`
Copies the contents of one diskette to another.
- `/usr/vpix/dosbin/display.sys`
Allows code page switching for foreign font displays.
- `/usr/vpix/dosbin/dosmount.exe`
Allows DOS drive letters to be assigned to subdirectories in the UNIX System file system.
- `/usr/vpix/dosbin/dospath.exe`
Sets the DOS switch character to “/” and the directory separator character to “\”.
- `/usr/vpix/dosbin/driver.sys`
Standard DOS loadable device driver.

- `/usr/vpix/dosbin/edlin.com`
The standard DOS line editor.
- `/usr/vpix/dosbin/ega.cpi`
Code page information table for EGA displays.
- `/usr/vpix/dosbin/ems.sys`
A DOS-loadable device driver that provides support for the LIM 3.2 standard of extended memory.
- `/usr/vpix/dosbin/exe2bin.exe`
Converts executable (.exe) files to binary format.
- `/usr/vpix/dosbin/fastopen.exe`
DOS command to store in memory the location of directories and recently opened files.
- `/usr/vpix/dosbin/fdisk.com`
Partitions a fixed disk (usable only on drive C in the VP/ix Environment).
- `/usr/vpix/dosbin/find.exe`
Searches a file for a constant string of text.
- `/usr/vpix/dosbin/format.com`
Formats a fixed disk or diskette to receive DOS files.
- `/usr/vpix/dosbin/graftabl.com`
Loads a table of graphics characters.
- `/usr/vpix/dosbin/graphics.com`
Prepares DOS for printing graphics.
- `/usr/vpix/dosbin/gwbasic.exe`
BASIC programming language interpreter.
- `/usr/vpix/dosbin/join.exe`
Joins a disk drive to a path name.
- `/usr/vpix/dosbin/keyb.com`
Loads foreign keyboard programs.
- `/usr/vpix/dosbin/keyboard.sys`
Contains tables that direct the `keyb.com` command to convert scancodes to ASCII characters.
- `/usr/vpix/dosbin/label.com`
Allows labeling of fixed disks or diskettes.

- `/usr/vpix/dosbin/lcd.cpi`
Code page information table for IBM PC Convertible LCD displays.
- `/usr/vpix/dosbin/link.exe`
The standard DOS linker program.
- `/usr/vpix/dosbin/mode.com`
Modifies parameters for the screen, printer port, and communications port.
- `/usr/vpix/dosbin/more.com`
Displays output one screen at a time.
- `/usr/vpix/dosbin/nlsfunc.exe`
Provides support for DOS extended country information.
- `/usr/vpix/dosbin/pemm.sys`
A DOS-loadable device driver that provides support for the LIM 4.0 standard of extended memory.
- `/usr/vpix/dosbin/print.com`
Prints a file on the line printer.
- `/usr/vpix/dosbin/quit.com`
Exits the VP/ix Environment.
- `/usr/vpix/dosbin/ramdrive.sys`
The standard DOS RAM disk driver. In the VP/ix Environment, this works only in lower or expanded memory.
- `/usr/vpix/dosbin/recover.com`
Recovers a bad disk or file.
- `/usr/vpix/dosbin/redir.exe`
The program that provides access to the UNIX file system from within the VP/ix Environment.
- `/usr/vpix/dosbin/replace.exe`
Replaces previous versions of files.
- `/usr/vpix/dosbin/restore.com`
Restores files backed up using `backup.com`.
- `/usr/vpix/dosbin/rundos.com`
A DOS program that allows execution of DOS programs directly from the UNIX System environment.

- `/usr/vpix/dosbin/select.com`
Allows loading of language-specific keyboards and font tables.
- `/usr/vpix/dosbin/share.exe`
Installs DOS file sharing and file locking.
- `/usr/vpix/dosbin/sort.exe`
Sorts data forward and backward.
- `/usr/vpix/dosbin/subst.exe`
Substitutes a string for a path name.
- `/usr/vpix/dosbin/sys.com`
Transfers DOS system files from one drive to another.
- `/usr/vpix/dosbin/tree.com`
Displays directory and file names.
- `/usr/vpix/dosbin/unixpath.exe`
Sets the DOS switch character to “-” and the directory separator character to “/”.
- `/usr/vpix/dosbin/xcopy.exe`
Copies files and subdirectories.
- `/usr/vpix/dosbin/xdir.exe`
Displays complete user, group, and protection information for files and directories.



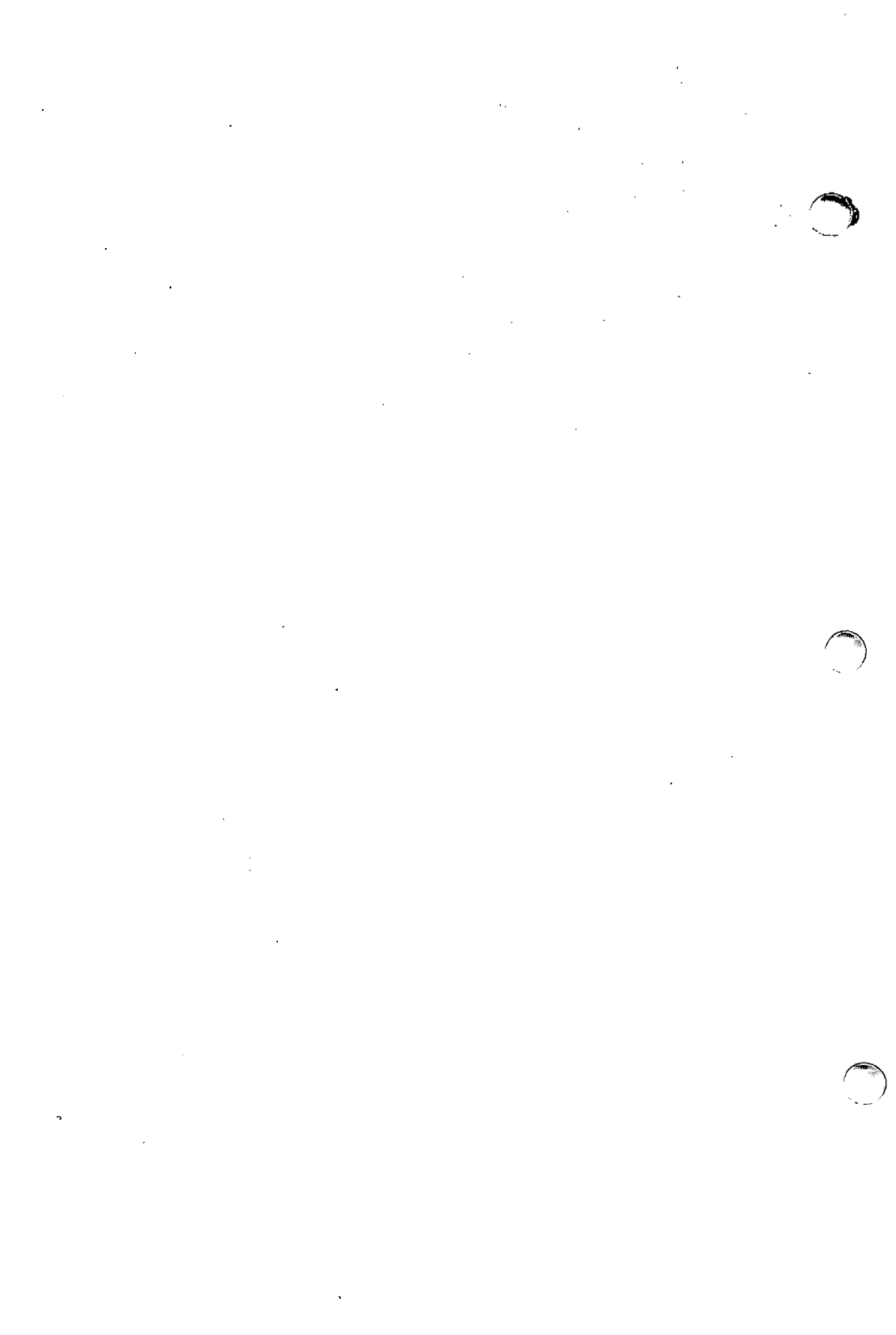
VP/ix Environment Maintenance Procedures

CONTENTS

1.	INTRODUCTION	1
1.1	Before You Begin	1
1.2	Overview of This Document	1
2.	CONFIGURING THE USER ENVIRONMENT	3
2.1	Why Configure the VP/ix Environment?	3
2.2	The VP/ix Configuration File	4
2.3	Summary of Recommendations for Resource Allocation	14
2.4	The VP/ix Resource Allocation Display	14
3.	CROSS-EXECUTION ENVIRONMENT	16
3.1	The Importance of PATHS	16
3.2	Running UNIX System Programs From DOS (rununix.exe)	17
3.2.1	How rununix.exe Works	17
3.3	Running DOS Programs From the UNIX System	19
3.4	Setting Up Login Environments	19
3.4.1	Setting Up a VP/ix Login Shell	19
3.4.2	Setting Up a DOS Application as the Login Environment	19
4.	TROUBLESHOOTING	21
4.1	Special Characters and VP/ix	21
4.2	Problems With Upper- and Lowercase Letters	22
4.3	Accessing a Program From the Wrong Directory	22
4.4	Execution Errors While Running a DOS Application	22
4.5	Attempting to List the C Drive	22
4.6	Problems With Mouse Response	23
4.7	Problems Accessing the Diskette Drive	23
4.8	Problems With the VP/ix Interface Menu	24
4.9	Problems With Printer Output	24
5.	DOS AND DOS FILES IN THE VP/ix ENVIRONMENT	26

5.1	DOS in the VP/ix Environment	26
5.2	DOS Files in the VP/ix Environment	26
5.2.1	The Redirector Program	26
5.2.2	Running DOS Commands From the UNIX System	26
5.2.3	Running UNIX System Commands From DOS	27
5.2.4	Expanded Memory Standard Driver	27
5.2.5	Facilitating Redirector Access	28
6.	ACCESSING NONSTANDARD DEVICES IN THE VP/ix ENVIRONMENT	29
6.1	IEM and DDA Devices	29
6.2	The <code>vpixdevs</code> File	30
6.3	Configuring Nonstandard Devices Into the VP/ix Environment	31
6.3.1	Configuring IEM Devices Into the VP/ix Environment	31
6.3.2	Configuring DDA Devices Into the VP/ix Environment	32
6.4	Making Nonstandard Devices Accessible to VP/ix Users	33
6.5	Modifying the Configuration Parameters for Nonstandard Devices	34
6.6	Removing Nonstandard Devices From the VP/ix Environment	34
	Appendix A: USING VP/ix ON SERIAL TERMINALS	35
1.	INTRODUCTION	35
1.1	Terminal Requirements	35
1.2	Before You Begin	36
2.	WHERE TERMINAL DESCRIPTIONS ARE LOCATED	37
3.	HOW TO DESCRIBE TERMINALS	39
3.1	Terminal Capabilities	39
3.1.1	Screen Attribute Definitions	41
3.1.2	The <code>svpix</code> and <code>fvpix</code> Sequences	41
3.1.3	Special Key Definitions	42
3.1.4	<code>pcompat</code>	43
3.1.5	<code>c_xon</code> and <code>c_xoff</code>	43
3.2	ASCII Input to Scan Code Mappings	44

3.3 Output Character Mappings	46
Appendix B: VP/ix ERROR MESSAGES	49
1. INTRODUCTION	49
2. CONFIGURATION ERRORS	50
3. FAULTY PROGRAM ERRORS	60
4. VP/ix INTERNAL ERRORS	64
Appendix C: INCOMPATIBILITIES BETWEEN DOS AND VP/ix	67



VP/ix* Environment Maintenance Procedures

1. INTRODUCTION

This document provides you with an overview of the basic procedures required to keep the VP/ix Environment running smoothly. It is intended for system administrators who are familiar with UNIX* System maintenance and with the DOS environment.

1.1 Before You Begin

Before you attempt any of the procedures outlined in this document, you must read and understand the following documents:

- “VP/ix and MS-DOS Primer”
- “VP/ix Environment Installation Instructions”
- “System Administration for New Users of the INTERACTIVE UNIX Operating System” in the *INTERACTIVE UNIX System Guide for New Users*
- “INTERACTIVE UNIX Operating System Maintenance Procedures” in the *INTERACTIVE UNIX Operating System Guide*

You must also be familiar with a text editor if your intended system configuration requires you to tailor some of the standard configuration files. Refer to the “Documentation Roadmap” to determine where to obtain additional information about text editors available for your system or additional information about the UNIX System. You may also need to refer to any additional documentation supplied by your vendor to completely install and tailor your system.

1.2 Overview of This Document

This document is divided into nine major sections, including three appendices:

1. INTRODUCTION

This section provides a general overview of this document.

2. CONFIGURING THE USER ENVIRONMENT

This section describes how to set up the VP/ix

Environment for a new VP/ix user and how to tailor the environment to each user's needs. It also explains how to assign UNIX System devices to the *virtual* DOS devices that VP/ix provides.

3. CROSS-EXECUTION ENVIRONMENT

This section explains how to run DOS programs from the UNIX System environment and UNIX System programs from the DOS environment. It also explains how to make the differences between the two kinds of applications transparent to users.

4. TROUBLESHOOTING

This section describes some of the problems that may be encountered when using the VP/ix Environment and discusses what to do about them.

5. DOS AND DOS FILES IN THE VP/ix ENVIRONMENT

This section explains how the version of DOS distributed with the VP/ix Environment differs from the standard MS-DOS* version 3.3.

6. ACCESSING NONSTANDARD DEVICES IN THE VP/ix ENVIRONMENT

This section describes the facilities supplied with the VP/ix Environment that enable you to add support for nonstandard PC hardware devices.

7. Appendix A: USING VP/ix ON SERIAL TERMINALS

This section is intended for experienced UNIX system administrators who want to use a terminal not already supported by their VP/ix vendor. It explains how to configure VP/ix so that DOS users can run applications on a wide variety of ordinary ASCII terminals.

8. Appendix B: VP/ix ERROR MESSAGES

This section lists the error messages that VP/ix can generate and provides the possible causes of and resolutions for each error.

9. Appendix C: INCOMPATIBILITIES BETWEEN DOS AND VP/ix

This section describes some of the differences between the VP/ix Environment and MS-DOS running as the sole operating system on a PC.

2. CONFIGURING THE USER ENVIRONMENT

The VP/ix system administrator is responsible for tailoring the VP/ix Environment. This section explains how to configure the VP/ix Environment for an individual user. In this section you will learn:

- Why the user environment must be configured.
- How to tailor the VP/ix configuration file.
- How to allocate resources to an individual user.

2.1 Why Configure the VP/ix Environment?

The VP/ix Environment is designed to allow multiple users to run multiple, simultaneous DOS sessions. System resources, such as a mouse device, a diskette drive, or a file must be shared by all users who require access to that resource. It is important, therefore, to ensure that no two users are competing for the same resource. For example, on a system with a single serial port and a modem, one user might be running a DOS communications package such as Ptel*, which uses the serial port and modem. If another user logs into the system and also attempts to run Ptel, there will be a problem. A resource such as a serial port cannot be shared simultaneously by two users, so the second user must be denied access to the resource.

A multi-user operating system, such as the UNIX System, provides basic services that regulate the way in which available system resources are to be shared. In contrast, DOS is a single-user system. DOS expects only one user on the system at any given time and, therefore, makes no provision to limit access to resources. Since the DOS system has no mechanism for protecting users from competing for a single resource, the VP/ix Environment provides this service.

The VP/ix Environment facilitates resource sharing through the use of a “lock” mechanism set up in the user configuration file. The user who first enters the VP/ix Environment is given exclusive access to all resources enabled in that user’s configuration file. A user who enters the VP/ix Environment after these resources are locked is denied access to those resources, even though that user’s configuration file may support it. The user who is initially allocated the diskette drive may make the drive available to other users

through the VP/ix Interface Menu. When a user exits the VP/ix Environment, all resources allocated to that user are released.

Because of the resource sharing that is required under VP/ix, a system administrator should configure each user's VP/ix Environment to automatically allocate only those resources that are required. For example, if a user typically logs in at the system console, which supports a mouse device, that user's environment should be configured to have access to the diskette drive and the mouse device. A user who typically accesses the VP/ix system from a serial terminal should not be configured to have access to the diskette drive or the mouse device.

2.2 The VP/ix Configuration File

The VP/ix configuration file is called `vpix.cnf`. It is automatically created for each user when the user's login account is configured for VP/ix (refer to section 3 of the "VP/ix Environment Installation Instructions"). This default file is copied from the directory `/usr/vpix/defaults` into the user's `$HOME/vpix` directory. Therefore, the default configuration file for each user is the same. To make a global change to the VP/ix configuration file, modify `/usr/vpix/defaults/vpix.cnf`. This will ensure that all VP/ix accounts that are subsequently established will receive the new default configuration. You may also tailor each user's individual `vpix.cnf` file to optimize the use of system resources and minimize resource conflicts.

The purpose of the `vpix.cnf` file is to associate the standard DOS device names (for example, A, C, and COM1) with the standard UNIX System device names. Under the VP/ix Environment, the UNIX System is responsible for executing requests to access devices. Therefore, the VP/ix Environment must translate DOS device names to UNIX System device names. For example, the DOS device name for the first diskette drive is A. The UNIX System device name for the first diskette drive is `/dev/rdisk/f0d9dt` (assuming the drive is a double-density, double-sided diskette drive). Refer to section 9 of the "System Administration for New Users of the INTERACTIVE UNIX Operating System" for more information about UNIX System device names.

If the VP/ix Environment is not configured correctly, users will not be able to access devices using DOS naming conventions. For example, a user might attempt to access drive A with this command:

```
VP/ix C:\> dir a:
```

If the user's configuration file does not contain an entry for drive A, the system responds with this message:

```
General failure error reading drive A
Abort, retry, or ignore?
```

You must correct the configuration file to correct this error.

If you modify the VP/ix configuration file from the UNIX System shell prompt, the changes will be reflected the next time you enter the VP/ix Environment. If you make changes to the configuration file from within the VP/ix Environment, you must exit and reenter VP/ix in order to have the changes take effect.

The VP/ix configuration file contains a series of entries. Each entry is either a comment line, a line that associates a DOS device name or auxiliary file with a UNIX System device name or file name, or a line that defines a nonstandard device name. Lines that begin with a semicolon (;) are comment lines. They are not read by the system when VP/ix is started. A comment line provides either descriptive information or an example of a DOS/UNIX device name entry. Remove the semicolon and any extra blank spaces at the beginning of a comment line to enable that entry.

Here is a sample of the default configuration file:

```

;
; VP/ix Configuration File
;
;
; DOS DEVICE          UNIX DEVICE PATH NAME
;
C                      /usr/vpixon/defaults/C:
ROM                    /usr/vpixon/defaults/rom
EGAROM                 /usr/vpixon/defaults/romega
VGAROM                 /usr/vpixon/defaults/romvga
CMOS                   /usr/vpixon/defaults/cmos
BOOTIMAGE              $HOME/vpixon/vpixon.img

; The following entries are provide as examples
; for customization of the VP/ix Environment

; To access the floppy as drive A: uncomment one of the following.
;
; For 360K:
;A                      /dev/rdisk/f0d9dt
; For 1.2 Meg
;A                      /dev/rdisk/f0q15dt
; For high-density 3.5":
;A                      /dev/rdisk/f0q18dt

; To add an additional Pseudo-Floppy (accessible as Drive B:)
;
;B                      $HOME/vpixon/B:

; To add an additional floppy which is attached to the second
; floppy drive (accessible as B:) on your system:
;
; For 360K:
;B                      /dev/rdisk/f1d9dt
; For 1.2 Meg
;B                      /dev/rdisk/f1q15dt
; For high-density 3.5":
;B                      /dev/rdisk/flq18dt

; To access an actual DOS partition on the fixed disk:
;D                      /dev/dsk/0p0

; To access a serial communications port:
;COM1                   /dev/tty00

; To access a serial communications port
; for use with a serial mouse:
;COM1MOUSE              /dev/tty00

; If you have a Microsoft Inport Bus Mouse:
;MOUSE                  /dev/mouse

; If you have a LOGITECH 3-Button Bus Mouse:
;LMOUSE                 /dev/logi

; If you have a line printer and you are using the standard UNIX
; printer spooler:

; LPT1                  /usr/bin/lp

; If you want Automatic Printer Buffer Flushing.
;AUTOFLUSH              20

```

DOS device names in the first column must not be followed by colons. For example, LPT1 is valid, while LPT1: is not. DOS device names in the first column must be flush with the left margin. For example, using an editor's destructive backspace to delete the semicolon (;) preceding LPT1 is valid, while using an editor to space over the semicolon is not.

The keyword for each DOS device or auxiliary file, the associated UNIX System device or file name, and the purpose of each device or file are outlined in the following table.

<i>DOS Device</i>	<i>Meaning Under DOS</i>	<i>Sample UNIX Equivalent</i>
C	The DOS partition on the first fixed disk of a standard DOS system.	<code>/usr/vpix/defaults/C:</code>
ROM	A file image of the standard MS-DOS BIOS, which is stored in the computer's ROM.	<code>/usr/vpix/defaults/rom</code>
EGAROM	For EGA (Enhanced Graphics) adapters. Specifies the EGA BIOS to be loaded when VP/ix is run.	<code>/usr/vpix/defaults/romega</code>
VGAROM	For VGA (Video Graphics) adapters. Specifies the VGA BIOS to be loaded when VP/ix is run.	<code>/usr/vpix/defaults/romvga</code>
CMOS	A file image of the CMOS information normally stored in CMOS RAM.	<code>/usr/vpix/defaults/cmos</code>
BOOTIMAGE	A fast-loading image of the user's VP/ix Environment.	<code>\$HOME/vpix/vpix.img</code>
A B	The diskette drives in a DOS system.	<code>/dev/rdisk/f0d9dt</code>
D	The DOS partition on the second fixed disk of a standard DOS system.	<code>/dev/dsk/0p0</code>
COM1 COM2	DOS serial ports.	<code>/dev/tty00</code>
COM1MOUSE COM2MOUSE	DOS serial ports, when used with a serial mouse.	<code>/dev/tty00</code>
MOUSE LMOUSE	Specifies the mouse device to be used.	<code>/dev/mouse</code> <code>/dev/logi</code>
LPT1 LPT2 LPT3	DOS printer devices.	<code>/usr/bin/lp</code>
AUTOFLUSH	Allows VP/ix to flush the printer buffer after a printer has been idle for a specified time.	

The next section describes each line as it appears in the default configuration file, explains why you might want to modify it, and, where appropriate, provides an example of a modified entry.

C /usr/vpix/defaults/C:

This entry establishes an *image* of the DOS system that is used by each VP/ix user. In a native DOS environment, the files and programs that are used to boot the system and initiate the environment are normally stored on the C drive (the fixed disk partition). Under the VP/ix Environment, an image of the C drive is stored in /usr/vpix/defaults/C:. The image contains the same information as is contained in the DOS partition of the C drive on a native DOS system. The /usr/vpix/defaults/C: file can only be modified by the root user. Refer to section 4 of the “System Administration for New Users of the INTERACTIVE UNIX Operating System” for more information about the root user.

This entry can be modified if a user wishes to customize the DOS environment by making changes to the autoexec.bat or config.sys files, for example. These DOS files are used to set system parameters and are located in the image of the C drive, /usr/vpix/defaults/C:. In order to modify these files (or any other files located on the C drive) without affecting the environment of others, the user must have a private copy of the C drive.

The system administrator can specify whether a user should have a private copy of the C drive when the sysadm program is run during VP/ix user installation. The decision to provide a user with a private copy of the C drive should be based on familiarity with the user’s requirements and the storage limitations of your system (the image of the C drive is a large UNIX System file). If a user is provided with a copy of the C drive, it is placed in the user’s vpix directory and made writable by the user. The sysadm program automatically edits the user’s vpix.cnf file to change the line for the C drive to:

```
C           $HOME/vpix/C:
```

For information about the disadvantages of installing

DOS software on the C drive, see the section entitled “Loading DOS Applications” in the “VP/ix and MS-DOS Primer.”

ROM /usr/vpix/defaults/rom

This entry defines a file image of the standard MS-DOS BIOS (Basic Input-Output System), which is normally stored in the computer’s ROM (read-only memory).

This entry should never be changed by the system administrator or by a user.

EGAROM /usr/vpix/defaults/romega

This entry specifies a file that contains instructions regarding high-resolution adapters, which is to be used by the system when the user has an EGA (Enhanced Graphics) adapter and display.

This entry should never be changed by the system administrator or by a user.

VGAROM /usr/vpix/defaults/romvga

This entry specifies a file that contains instructions regarding high-resolution adapters, which is to be used by the system when the user has a VGA (Video Graphics) adapter and display.

This entry should never be changed by the system administrator or by a user.

CMOS /usr/vpix/defaults/cmos

This entry defines a file image of the CMOS information (definitions of the hardware on the system) normally stored in the battery backup CMOS RAM (random access memory).

This entry should never be changed by the system administrator or by a user.

BOOTIMAGE \$HOME/vpix/vpix.img

This entry specifies the name of the VP/ix quickboot image file, which contains a fast-loading image of a user’s VP/ix Environment. This quickboot image is used to enter the VP/ix Environment (via the `vpix -r` command) without going through the DOS bootup sequence or to run a DOS application (such as

Lotus* 1-2-3* or dBASE*) directly from the UNIX System shell prompt.

This entry will never need to be changed.

A, B /dev/rdisk/f0d9dt

These entries map the standard DOS devices, A and B, to their respective counterparts in the UNIX System environment. For example, drive A, which is normally the first diskette drive in DOS, may be mapped to /dev/rdisk/f0d9dt, which is the UNIX System device name for a double-density double-sided diskette drive. This allows a user in the VP/ix Environment to request a directory listing of this drive, using the DOS device name A:. If there is no entry for drive A in the user's configuration file, a request for a directory listing of drive A will result in an error message similar to this:

```
General Failure error reading drive A
Abort, Retry, Ignore?
```

The VP/ix Environment will automatically determine the density of both the physical drive and the media inserted into it regardless of which UNIX System device name (for double- or high-density) is used as the diskette device. For a more detailed discussion of UNIX System devices and UNIX System device path names, see section 9 of the "System Administration for New Users of the INTERACTIVE UNIX Operating System."

Remove the semicolon from the beginning of the line containing the appropriate entry (depending on the type of drive you have and which drive you wish to assign to the user) to give the user access to that diskette drive. Note that if the user wants to boot DOS from the A drive instead of the C drive, he or she must:

- have an A entry in his or her configuration file,
- have a valid DOS boot diskette in the diskette drive to which A is mapped in that entry, and
- invoke `vpix` with the `-b` option, as described in the `vpix(1)` manual entry.

D /dev/dsk/0p0

If you have an actual DOS partition on your first fixed disk, you can access it under VP/ix in one of two ways. The preferred method is to mount it as part of the UNIX System file system using the DOS-FSS facility described in Appendix A of the “INTERACTIVE UNIX Operating System Primer,” and access it through the Z drive as you would access any other UNIX System file. Alternatively, you may map the D drive (normally the DOS partition on the second fixed disk in native DOS) to the UNIX System device name for the first fixed disk, /dev/dsk/0p0. Note that this will only work if the user has UNIX System read and/or write permissions on /dev/dsk/0p0. As this is not normally desirable, this method of DOS partition access is not preferred.

COM1, COM2 /dev/tty00, /dev/tty01

These entries map DOS serial port names to the names of corresponding UNIX System serial devices. Typically, COM1 is mapped to /dev/tty00 or /dev/tty01 and COM2 is mapped to /dev/tty01.

Remove the semicolon from the beginning of the line containing the COM1 entry if a user wishes to use an application that requires serial communication. To allow a user access to the second serial communications port (if your computer has one) as well as the first, add the following entry to the configuration file:

```
COM2          /dev/tty01
```

COM2 must not be specified unless COM1 is also specified. If you want to only allow access to the second port, add the following entry:

```
COM1          /dev/tty01
```

Note that COM1 and COM2 are not supported on MULTIBUS* systems and their entries should remain commented out on such systems.

COM1MOUSE, COM2MOUSE /dev/tty00, /dev/tty01

These entries should be used instead of COM1 and COM2 whenever the serial port is being used for a

serial mouse. This will ensure proper mouse behavior when switching among virtual terminals, as described below in the MOUSE entry. All the comments pertaining to COM1 and COM2 above apply here as well.

MOUSE, LMOUSE /dev/mouse, /dev/logi

This entry identifies the mouse pointing device. This entry is specifically for users who have a Microsoft* Inport-2 Bus or a LOGITECH* 3-Button Bus mouse pointing device. It does not apply to serial pointing devices that use COM1 or COM2 and plug into your computer's serial ports. (If you have such a mouse and wish to give a VP/ix user access to it, you must uncomment the appropriate COM1 or COM2 entry, depending on which serial port the mouse is plugged into.) The mouse device is not currently supported on MULTIBUS systems.

Remove the semicolon from the line containing the MOUSE entry if the user has a Microsoft Inport-2 Bus mouse pointing device. Remove the semicolon from the line containing the LMOUSE entry if the user has a LOGITECH 3-Button Bus mouse pointing device. Otherwise, these entries should not be modified.

Note that the mouse is automatically released and reacquired whenever you leave or enter the virtual terminal running your VP/ix session. Thus, multiple VP/ix sessions may be run on different virtual terminals, and the mouse will automatically "follow along" when you switch among them.

LPT1, LPT2, LPT3 /usr/bin/lp

These entries map the standard DOS printer devices to the name of the UNIX System printer spooler program. If different printers are available on your system, you may need to provide options to the spooler. For example, this entry maps LPT2 to a command that spools jobs for your laser printer:

```
LPT2          /usr/bin/lp -dlsr
```

LPT2 must not be specified unless LPT1 is also specified, and LPT3 must not be specified unless both LPT1 and LPT2 are specified.

Note that it may be necessary for the user to use the `Flush Printer` option on the VP/ix Interface Menu to actually send text to the printer. Refer to the “VP/ix and MS-DOS Primer” for more information about the printer spooler and the `Flush Printer` option on the VP/ix Interface Menu.

AUTOFLUSH

This entry will enable and disable the automatic printer buffer flushing mechanism. The number following this entry represents the time, in seconds, a printer is idle before it will be flushed. If the number is 0, or the `AUTOFLUSH` entry is commented out, the autoflush mechanism is disabled.

Any line that begins with a keyword not listed here is assumed to be an Installable Emulation Module or a Direct Device Attachment. These facilities are described in detail in section 6 of this document. An entry describing these devices can also be found in the file `/usr/vpix/etc/vpixdevs`.

2.3 Summary of Recommendations for Resource Allocation

- Make global configuration changes to the default `vpix.cnf` file in the `/usr/vpix/defaults` directory to ensure that all new VP/ix user accounts reflect the changes.
- Determine which users need to customize their DOS environments and provide those users with private copies of the C drive.
- Allocate resources according to individual user's needs as well as the physical proximity of users to resources. Edit each user's `vpix.cnf` file accordingly.
- Determine whether users will be allowed to modify their configuration information or whether the system administrator will retain control of all resource allocation. Assign permissions to the users' `vpix.cnf` files accordingly.

2.4 The VP/ix Resource Allocation Display

When a user pops up the VP/ix Interface Menu, a secondary display box also appears to the right.

VP/ix Version 1.2 (ESC)ape menu (F)loppy release (P)rinter flush (Q)uit VP/ix (R)eset VP/ix (S)ound OFF (C)lose serial (E)nter shell	<table border="1"> <tr> <td>Device</td> <td>Assignment</td> </tr> <tr> <td>A:</td> <td>/dev/rdisk/f0q15dt</td> </tr> <tr> <td>C:</td> <td>/usr/vpix/defaults/C:</td> </tr> <tr> <td>Terminal</td> <td>Assignment</td> </tr> <tr> <td>AT386</td> <td>/dev/console</td> </tr> </table>	Device	Assignment	A:	/dev/rdisk/f0q15dt	C:	/usr/vpix/defaults/C:	Terminal	Assignment	AT386	/dev/console
Device	Assignment										
A:	/dev/rdisk/f0q15dt										
C:	/usr/vpix/defaults/C:										
Terminal	Assignment										
AT386	/dev/console										

This is the simplest way to check precisely what devices are configured for the user, and that the user's `vpix.cnf` file has been set up properly. The left-hand section of the display lists all of the DOS devices that have been activated in the user's configuration file and the actual UNIX System device to which each DOS device has been mapped. If the user has released the device, for example, using the **(F)loppy release** or **(C)lose serial** options on the VP/ix Interface Menu, there will be no UNIX System device name listed and the word `released` will appear instead. Should the user reacquire the device, for example by referring to the A drive in the case of the diskette, the UNIX System device name will once again be displayed when the VP/ix Interface Menu is popped up.

The right-hand section of the display indicates the value of the user's `TERM` environment variable and the UNIX System terminal device on which he or she is logged in.

3. CROSS-EXECUTION ENVIRONMENT

The environment that allows a user to run a DOS program from the UNIX System and a UNIX System program from DOS is called a *cross-execution* environment. Cross-execution is available to all VP/ix users. This section describes the VP/ix cross-execution environment and discusses the problems a user may encounter. In this section you will learn:

- The importance of PATHS.
- How to run UNIX System programs from DOS.
- How to run DOS programs from the UNIX System.
- How to set up login environments.

3.1 The Importance of PATHS

A user can run a DOS program from the UNIX System environment by changing to the directory in which the DOS program is located and executing the program. To execute a DOS program directly from the UNIX System shell prompt *without* having to explicitly change to the directory in which it is located, the program must be located in a directory specified in a *search path*. A search path specifies a set of directories to be searched for an executable file or command. It is defined in the shell `.profile` file. (Refer to section 8 of the “System Administration for New Users of the INTERACTIVE UNIX Operating System” for more information about the `.profile` file.) Two path variables are required in the VP/ix Environment, `PATH` and `DOSPATH`. `PATH` defines the directories to be searched for a UNIX-executable file. `DOSPATH` defines the directories to be searched for a DOS-executable file if a UNIX-executable file of the specified name is not found. The directories specified in `DOSPATH` must be a subset of the directories specified in the shell variable `PATH`, which is set in the user’s `.profile` file. The `DOSPATH` variable is automatically added to the user’s `.profile` file when the VP/ix user account is installed.

If a user tries to execute the command `dosprog` from the UNIX System environment, for example, the shell searches for the file `dosprog` in the first directory specified in `PATH`. If the shell finds a UNIX System file by that name, it runs that file as an executable UNIX System program. If no UNIX System file by that name is found, the shell looks to see if that directory is also specified in `DOSPATH`. If it is not, the shell continues the process

with the next directory specified in `PATH`. However, if the directory *is* also specified in `DOSPATH`, then the shell searches for the equivalent DOS-executable file name in that directory. The shell searches for the files `dosprog.com`, `dosprog.exe`, and `dosprog.bat`, in that order. (The suffixes `.com`, `.exe`, and `.bat` indicate DOS-executable files.) If it finds a file that matches the above specification in the directory in `DOSPATH`, the VP/ix Environment is quickbooted and the DOS command is run.

For example, to execute DOS programs located in the directory `/usr/bob/dosbin` directly from the UNIX System shell prompt *without* having to change to that directory, modify the `DOSPATH` and `PATH` variables in the user's `.profile` file to look like this:

```
DOSPATH=$DOSPATH:/usr/bob/dosbin
PATH=$PATH:/usr/bob/dosbin
```

The user can now run the `dosprog` program at the UNIX System shell prompt from any directory.

3.2 Running UNIX System Programs From DOS (`rununix.exe`)

The VP/ix Environment is designed to allow users to execute any UNIX System command from the DOS under UNIX System environment. When VP/ix is initially installed, a DOS program called `rununix.exe` is put into the directory `/usr/vpix/unixbin`. This program allows a user to execute UNIX System commands (which are located in the UNIX System directories `/bin` and `/usr/bin`) while using the VP/ix Environment.

UNIX System conventions must be used to run a UNIX System command in the VP/ix Environment. Options must be specified with the UNIX System dash (`-`) character instead of the DOS slash (`/`), and directory and file names must be separated with the UNIX System slash (`/`) instead of the DOS backslash (`\`).

3.2.1 How `rununix.exe` Works

Each UNIX System command located in the directories `/bin` and `/usr/bin` must have an equivalent DOS-executable file name associated with the program `rununix.exe` in order to run in the VP/ix Environment. DOS-executable file names are automatically created in the directory `/usr/vpix/unixbin` and linked to `rununix.exe` when VP/ix is first installed.

Each DOS-executable file name must end in the suffix `.exe`. For example, the DOS-executable name for the UNIX System command `ls` is `ls.exe`. Since they are linked, there is only one actual copy of the file residing on the disk. The DOS-executable file name `ls.exe` is simply a link to the `rununix.exe` file. A link is a pointer to the original file; it is not a copy of the contents of the original file.

The system file `/usr/vpix/unixbin/rununix.exe` is available to all VP/ix users. Each VP/ix user is also provided with an individual copy of `rununix.exe`. When an individual VP/ix account is established, the system automatically creates the directory `vpix` and the subdirectory `vpix/unixbin` in the user's login (`$HOME`) directory. The system file `/usr/vpix/unixbin/rununix.exe` is copied to the user's `$HOME/vpix/unixbin` directory. The user has access to all of the UNIX System commands linked to the system's `rununix.exe` file in the directory `/usr/vpix/unixbin` and can also link personal UNIX System programs and commands to an individual `rununix.exe` file.

For example, Bob's UNIX System program `unixprog` is located in his `bin` directory, `/usr/bob/bin`. To use his `unixprog` program in the VP/ix Environment, an equivalent DOS-executable file name must be linked to the file `/usr/bob/vpix/unixbin/rununix.exe`. This can be done from either the VP/ix Environment or the UNIX System shell with this command:

```
$ ln /usr/bob/vpix/unixbin/rununix.exe /usr/bob/vpix/unixbin/unixprog.exe
```

The `unixprog` command is now available in Bob's environment.

Note that in order for a UNIX System program to be executable in the VP/ix Environment, the DOS-executable name of the program must be located in a directory specified in the user's DOS path (set in the `autoexec.bat` file) and the actual UNIX System command must be located in a directory specified in the user's UNIX System path (set in the `PATH` variable in `.profile`). Thus, in this example, `/usr/bob/vpix/unixbin` has been placed in Bob's DOS path when he started up VP/ix, and `/usr/bob/bin` was, presumably, put in his UNIX System path when he logged into the UNIX System. Also note that the user must give his UNIX System command a name that is a legal DOS file name (without an extension) in order to be able to access it while under VP/ix. Thus

the command name must have eight or fewer characters and contain no uppercase letters or dots (.).

3.3 Running DOS Programs From the UNIX System

It is not necessary to enter the VP/ix Environment in order to run a DOS command. The VP/ix installation procedures automatically create an environment in which a user can run a DOS program, located in the user's current directory, from the UNIX System shell prompt. For example, if the DOS application Lotus 1-2-3 is located in the directory `/usr/dos/123`, it can be run directly from the UNIX System shell prompt in this way:

```
$ cd /usr/dos/123
$ 123
```

A user can run DOS programs from other directories if the `DOSPATH` and `PATH` shell variables located in the user's `$HOME/.profile` file are correctly modified and exported. Refer to section 3.1.

3.4 Setting Up Login Environments

Under VP/ix, a user normally logs into the UNIX System environment and then executes the `vpix` program to access the VP/ix Environment. However, it is possible for a user to instead be placed directly in the VP/ix Environment or in a DOS application at login.

3.4.1 Setting Up a VP/ix Login Shell

You can set the `vpix` program to be a user's login shell by modifying the UNIX System file `/etc/passwd`. Use the `moduser` option from the `sysadm` User Management submenu to change the user's login shell. Select the option called `chgshell` and change the user's login program to `/usr/bin/vpix`. A user whose login shell is `vpix` is placed immediately in the VP/ix Environment at login. Upon exiting the VP/ix Environment, the user also exits the UNIX System and a new login prompt appears on the screen.

3.4.2 Setting Up a DOS Application as the Login Environment

Some users may want to be placed in a DOS application environment, such as Lotus 1-2-3, when they log in. You cannot make this application the user's login shell because the DOS application is not an executable UNIX System program. If you wish to place a VP/ix user directly into a DOS application upon login to the UNIX System, you must modify the user's `.profile` file to execute the DOS application from the UNIX System and then log out. For

example, to have a user log in directly to Lotus 1-2-3, which is installed in the UNIX System directory `/usr/dosapp/123`, add the following lines to the end of the user's `.profile` file:

```
cd /usr/dosapp/123
123
exit
```

When the user logs in, the DOS application in the file `123` will be executed. When the user exits this application, the UNIX System is also exited and a new login prompt appears on the screen.

4. TROUBLESHOOTING

The VP/ix Environment allows you to run DOS and the UNIX Operating System concurrently. Each system operates under a unique set of rules and conventions, although there is some overlap between the two systems.

Common problem areas include:

- The use of special characters (\, *, ?).
- The use of upper- and lowercase characters.
- Attempts to access programs from the wrong directory.
- Execution errors while running a DOS application.
- Attempts to list the C drive.
- Problems with the mouse device.
- Problems accessing the diskette drive.
- Problems with the VP/ix Interface Menu.

This section describes some of the most common problems areas and suggests some corrective actions.

4.1 Special Characters and VP/ix

The UNIX System shell interprets commands and performs expansion differently from the DOS command processor. The UNIX System directory separator / is the DOS switch character. A user might type the following in an attempt to execute the DOS command `chkdsk.com` from the UNIX System:

```
$ chkdsk/f a:
```

Instead of executing the command, however, the UNIX System shell interprets the / character as a directory separator and searches for a file named `f` in the directory `chkdsk`.

The UNIX System shell also expands wildcard characters, such as * and ?, and uses the \ as an escape character. However, the DOS command processor uses the \ as a directory separator and leaves the expansion of the wildcard characters to the DOS system itself.

To avoid the problems inherent in these differences, arguments to DOS commands that are running from the UNIX System shell should be enclosed in single quotes. For example:

```
$ chkdsk '/f a:'
```

or

```
$ copy '*.c a:\srcdir\cdir'
```

The UNIX System shell does not interpret arguments within single quotes, so they are passed on correctly to the VP/ix Environment when it quickboots.

4.2 Problems With Upper- and Lowercase Letters

The UNIX System shell differentiates between lower- and uppercase letters, but the DOS command processor maps everything to uppercase letters before passing it to the DOS system. UNIX System users usually use lowercase letters to name files, and DOS files located in the UNIX System file system are stored with their names in lowercase. As a general rule, therefore, it is simpler to use lowercase letters when in the VP/ix Environment. However, if a UNIX System command requires that an argument or option be specified in uppercase letters, then the argument or option must be typed in uppercase at the VP/ix command line.

4.3 Accessing a Program From the Wrong Directory

Some DOS programs, such as WordStar*, search for files in the current directory. Attempting to execute these programs from a directory other than the directory in which they are installed will result in error messages. While running such DOS programs from the UNIX System, the user must be in the directory in which these programs are installed.

4.4 Execution Errors While Running a DOS Application

VP/ix will run most DOS applications. However, there are some “ill-behaved” DOS programs that use nonstandard devices or port addresses. These programs may cause problems in the VP/ix Environment. If the program you are attempting to run does not function correctly, check the file `/usr/vpix/README`. This file contains a list of problems that are generated when some DOS applications are run under VP/ix. If the problem does not appear in this list, try to determine whether the application runs properly under native DOS. If it does, notify your VP/ix vendor of the problem.

4.5 Attempting to List the C Drive

The DOS image file `/usr/vpix/defaults/C:` (or `$HOME/vpix/C:`, if the user has a personal C drive image) is a

single UNIX System file containing the physical image of a DOS file system. A user cannot obtain a UNIX-style listing of the files on the C drive by using the `ls` command because the command cannot “see” the files inside C. To list the C drive from within the VP/ix Environment, use the DOS `dir` command.

4.6 Problems With Mouse Response

The mouse pointing device is not currently supported on MULTIBUS installations. If have a MULTIBUS system, you may skip this section.

If a user has installed an application that uses a mouse as the pointing device and the mouse does not work with the application, follow these steps:

1. Make sure that the mouse was specified as the pointing device during the DOS application installation.
2. Make sure that a driver for the mouse has been configured into your UNIX System (refer to the “INTERACTIVE UNIX Operating System Maintenance Procedures” for information about configuring the kernel to support new devices).
3. Make sure that an entry for the mouse appears and is not `released` in the display section of the VP/ix Interface Menu. If you are using a Microsoft Inport Bus Mouse or a LOGITECH 3-Button Bus Mouse, there should be a `MOUSE` entry; if you are using a serial mouse, there should be a `COM1` entry. If no entry is present, check the user’s `vpix.cnf` file and alter it as described in section 2.2.
4. Make sure that the mouse is properly plugged into the correct device.

4.7 Problems Accessing the Diskette Drive

If a user is not able to access the diskette drive from the VP/ix Environment, follow these steps:

1. Pop up the VP/ix Interface Menu and check the existence of an entry for the drive the user is trying to access (A or B). If the entry does not exist, alter the user’s `vpix.cnf` file as described in section 2.2.
2. If the entry is there but is marked `released`, and this error message is generated when trying to access the device, it

means that `vpix` could not obtain exclusive access to the diskette device:

```
General Failure error reading drive A
Abort, Retry, Ignore?
```

This can be caused by:

- Another user who has accessed the diskette drive from VP/ix and has not released it. The other user can release it by using the (F)loppy Release option on the VP/ix Interface Menu or by exiting the VP/ix Environment.
- The user does not have access permission for the diskette drive. Make sure the diskette device file to which you mapped the DOS A or B drive has read and write permissions for all users.

4.8 Problems With the VP/ix Interface Menu

If a user invokes the VP/ix Interface Menu from the VP/ix Environment while in a DOS graphics application, the menu is not visible. However, the keystrokes used to select the menu options will continue to operate. To return to the VP/ix Environment, use **ESC**. To exit the VP/ix Environment, type `q`. (Note: the VP/ix menu only responds to keys that are in the same key location as those labeled on a U.S. English Keyboard. When using European Keyboards, care must be taken to select the correct key by its position, not necessarily by the label. For example, on a French Keyboard, the key labeled `a` is in the same key location as the `q` key on a U.S. keyboard. The user should then press `a` in order to exit the VP/ix Environment.)

4.9 Problems With Printer Output

It is possible for a VP/ix user to request printer output using either an application's printing facilities or the DOS `print` command, to flush the printer from the VP/ix Interface Menu, and to have only part of the job appear when printed. When the user flushes the printer again, the rest of the job will appear. The problem is that the user does not know when the application or DOS has completed "printing," thus making the full job available for the real UNIX System print spooler.

Instruct the user to simply wait longer before flushing the printer (or exiting the VP/ix Environment). A rule of thumb is that he or

she should wait 1 minute per 15,000 bytes (about two pages) of print job before flushing the printer.

Alternatively, the user may enable the AUTOFLUSH option in their `vpix.cnf` file. This automatically flushes the printer buffer whenever the printer has been idle for the number of seconds specified on the AUTOFLUSH line in the `vpix.cnf` file.

5. DOS AND DOS FILES IN THE VP/ix ENVIRONMENT

The VP/ix Environment is distributed with an enhanced version of MS-DOS that is designed to work specifically with VP/ix. This section describes:

- The enhanced version of DOS distributed with VP/ix.
- The DOS files that are specific to the VP/ix Environment.

It is intended for system administrators who are very familiar with the standard DOS environment.

5.1 DOS in the VP/ix Environment

The version of DOS supplied with the VP/ix installation has been modified to run more efficiently in the VP/ix Environment. Changes were made to improve performance and do not alter the functionality of DOS in any way. Note that *any* “off the shelf” version of DOS will run under VP/ix, although not as efficiently as the version supplied with the VP/ix Environment. For more information about the modifications made to this version of DOS, refer to the *VP/ix Technical Guide*.

5.2 DOS Files in the VP/ix Environment

Some of the DOS files that are distributed as part of the installation are specific to the VP/ix Environment. These files are used by the system and should not be altered by the user. Each VP/ix-specific file is described in the following subsections.

5.2.1 The Redirector Program

`redir.exe` is a DOS program that allows other DOS programs to access the UNIX System file system. It is also called *the redirector*. `redir.exe` should be run from `autoexec.bat`.

5.2.2 Running DOS Commands From the UNIX System

`rundos.com` is a DOS program that facilitates the execution of other DOS programs from the UNIX System. One of its functions is to help save and fast load a DOS image. This should be the last program executed in the `autoexec.bat` file; any programs executed after this program will not be included in the DOS environment after a fast load operation.

5.2.3 Running UNIX System Commands From DOS

`rununix.exe` is a DOS program that facilitates the execution of UNIX System programs from DOS. This file is described in greater detail in section 3.

5.2.4 Expanded Memory Standard Driver

`ems.sys` and `pemm.sys` are DOS-loadable device drivers that allows access to the “Expanded Memory Specification” (EMS) as defined by Lotus-Intel*-Microsoft (LIM), version 3.2 and 4.0 respectively. They can be used by certain DOS applications to partially circumnavigate the limitation known as the “640K barrier.” (DOS normally prevents programs from accessing memory addresses beyond 640K.) If the EMS.SYS or the PEMM.SYS drivers have been loaded, part of the virtual address space available to the UNIX System will be dedicated to the VP/ix process. Only a small number of DOS applications are written to take advantage of EMS.

Up to 2 MB of LIM expanded memory (EMS) are available under VP/ix. The LIM 4.0 is backward compatible with the old LIM 3.2. The DOS driver is named `pemm.sys`. It uses the same command line syntax as the previous 3.2 driver, `ems.sys`. The driver must be placed in the `config.sys` file on the C: drive if it is to be used.

In a multi-user environment it is *not* recommended that EMS be configured for the communal C: drive in `/usr/vpix/defaults`. This is because unnecessary use of EMS greatly reduces system performance, and reduces the maximum number of VP/ix processes that can be run at one time.

To install this driver, a user must add the following line to his or her `config.sys` file:

```
device=ems.sys Sx
```

Or:

```
device=pemm.sys Sx
```

The `S` argument modifies the amount of EMS assigned to the particular user, and `x` is the number of 16K memory pages to be assigned. The maximum of value of `x` is 128, which dedicates 2 MB of EMS to the user.

Here is the procedure for using EMS:

1. If the person who needs access to EMS memory has already been added as a VP/ix user with his own copy of the C drive, go to step 6.
2. Login as the user who needs EMS (who must already have been added as a VP/ix user).
3. Type `cp /usr/vpix/defaults/C: $HOME/vpix`
4. Type `chmod 644 $HOME/vpix/C:`
5. Edit the file `$HOME/vpix/vpix.cnf`, replacing `/usr/vpix/defaults/C:` with `$HOME/vpix/C:`
6. Boot VP/ix.
7. Type `COPY Z:\USR\VPPIX\DOSBIN\EMS.SYS C:\`
8. Type `COPY C:\CONFIG.SYS + CON: C:\CONFIG.SYS`
9. Type either `Device = ems.sys s30` or `Device = pemm.sys s30`
10. Press **CTRL** **Z**.
11. Press **CTRL** **ALT** **DEL** to reboot VP/ix.
12. The EMS driver will print a herald as you reboot, and the installation is complete.

In step 9, the amount of expanded memory assigned to this user is configurable. After the `s` parameter, a value from 1 to 128 will specify the number of 16K EMS memory pages to assign. For example, `s30` will provide for 480K of expanded memory. This is the maximum value recommended on systems with only 2 MB of physical (real) memory.

The EMS memory space is used only for program data, and only by those few DOS applications that were specifically designed to take advantage of EMS.

5.2.5 Facilitating Redirector Access

`dosmount.exe` is a DOS file that works in conjunction with the redirector and allows the mapping of a DOS drive name to any directory in the UNIX System. It also allows you to deactivate or reactivate the redirector and change the DOS host name. For usage details, type `dosmount` at the VP/ix prompt.

6. ACCESSING NONSTANDARD DEVICES IN THE VP/ix ENVIRONMENT

VP/ix is pre-configured to support certain “standard devices,” such as diskette drives and fixed disks, serial ports (COM1 and COM2), and printer ports (LPT1 and LPT2). Other devices are classified as “nonstandard.” Examples of nonstandard devices that you may want to install at some point include networking cards, a tape drive, or an unsupported type of mouse.

☛ If you have a standard hardware configuration, you will not need any of the information in this section. You should not attempt to install a nonstandard device unless you have some understanding of hardware and system configurations. This section does not attempt to explain all the hardware-related concepts necessary to fully understand the procedures discussed here.

This section describes how VP/ix accesses nonstandard devices and explains how to make nonstandard devices accessible to the VP/ix user. Specifically, it describes:

- The difference between IEM and DDA devices.
- The function of the `vpixdevs` file.
- How to configure nonstandard devices into VP/ix.
- How to make nonstandard devices accessible to VP/ix users.
- How to modify the configuration parameters for nonstandard devices.
- How to remove nonstandard devices from the VP/ix Environment.

6.1 IEM and DDA Devices

VP/ix supports two different types of nonstandard devices: Direct Device Attachment (DDA) devices and Installable Emulation Module (IEM) devices. DDA devices are accessed directly by VP/ix in exactly the same manner as under native DOS. IEM devices are *emulated*; that is, VP/ix does not access them directly, but does so through a piece of software called an Emulation Module that is configured into VP/ix, and a corresponding UNIX System device driver.

DDA devices are typically supplied with a DOS device driver (often called *name.sys*) that must be installed into users' *config.sys* files. DOS application programs access the device through the device driver, which accesses the hardware directly.

Any device that functions in a standard PC may potentially be installed as a DDA device. The principal restriction is that devices that use DMA (for example, many network cards) may not be configured into the VP/ix Environment as DDA devices. Consult your INTERACTIVE UNIX System vendor for further information.

IEM devices are more complex, but easier to install. They are typically supplied by the hardware manufacturer or INTERACTIVE UNIX System vendor with Emulation Module software and a UNIX System device driver specifically tailored for the VP/ix Environment. The *device emulation software* used by IEM devices must be configured into the *vpix* program, and the additional UNIX System device driver must be configured into the UNIX System kernel. (Note that VP/ix's support for the Microsoft Bus Mouse is provided by means of an IEM called *bmouse*.) As with DDA devices, DOS application programs access the device through the DOS device driver, but this driver does not access the hardware directly. Instead, its actions are intercepted and interpreted by the Emulation Module, which communicates with the UNIX System device driver responsible for accessing the hardware.

While DDA devices require less software support than IEM devices, they are not as powerful, because they may be accessed by only one user at a time. IEM devices, because their control software is written specifically for VP/ix, may be shared by several users.

6.2 The *vpixdevs* File

All nonstandard devices known to VP/ix are defined in the file */usr/vpix/etc/vpixdevs*. Although this is a text file, it should never be edited directly; instead it should be modified using the options available on the *sysadm* VP/ix Management submenu.

The *vpixdevs* file contains eight fields, but not all fields are necessarily used for each device. Consult your hardware manufacturer's documentation to determine which fields are necessary for your particular device. The *vpixdevs* file contains the following fields:

- The device's logical name (the name used by VP/ix users in their *vpix.cnf* files).

- The name of the IEM associated with the device.
- The name of the UNIX System device special file associated with the device.
- The Interrupt ReQuest level (IRQ) used by the device.
- The range of I/O addresses used by the device.
- The range of controller memory used by the device.
- The DMA channel used by the device.
- A configuration flag specifying whether the device has been configured into `vpix` (and hence, whether it is accessible by VP/ix users).

Devices that are defined in `vpixdevs` are said to be “registered” with VP/ix. Note that it is possible for a device to be registered but still inaccessible, if the value of its configuration flag is N. The values of this and other fields are modified using the `vconfig` option on the VP/ix Management submenu of the System Administration menu described in section 6.5.

6.3 Configuring Nonstandard Devices Into the VP/ix Environment

All devices must be physically installed into the computer before they can be configured into the VP/ix Environment. In order to physically install a device, you should shut down the system and then follow the installation instructions provided by the device’s manufacturer. Once the system has been rebooted, you will follow one of the procedures described below, depending on whether you have an IEM or a DDA device.

6.3.1 Configuring IEM Devices Into the VP/ix Environment

The software support for IEM devices is provided on a diskette that should be installed using the `installpkg` option of the `sysadm` Software Management submenu. The installation procedure is completely automatic and will require no intervention unless some of the device configuration parameters (for example, the IRQ level) conflict with those used by one of VP/ix’s standard devices. If such a conflict occurs, the `installpkg` program will display a warning message. If this occurs, you will need to reconfigure the device’s board according to the instructions provided by the manufacturer and then rerun the installation procedure.

It is also possible that the device's configuration parameters could conflict with those of another IEM or DDA device that you have already installed. In this case you might want to reconfigure the board if possible, in order to remove the conflict. If you do so, you will need to select the `vconfig` option from the `sysadm` VP/ix Management submenu in order to specify to VP/ix the new parameters for that device. Refer to section 6.5 for more information on how to do this.

6.3.2 Configuring DDA Devices Into the VP/ix Environment

DDA devices are more difficult to configure into the VP/ix Environment than IEM devices, and there is more opportunity for error. Do not attempt to add a DDA device unless you fully understand the device configuration parameters discussed below.

Once the DDA device has been physically installed in the system, you must tell VP/ix how it is configured. Depending on the device, you will need to specify one or more of the configuration parameters listed below. (Note that devices that use DMA, such as many networking cards, *cannot* be accessed under VP/ix as DDA devices.) Configuration parameters that may be necessary are:

- The Interrupt ReQuest level (IRQ) used by the device.
- The range of I/O addresses used by the device.
- The range of controller memory used by the device.
- The name of the UNIX System special device file associated with the device.

Refer to the documentation supplied with the device for the necessary information for specifying these parameters. If insufficient information is available, consult your hardware manufacturer.

Access the System Administration main menu and select option 4, `packagemgmt`. When the Package Management menu appears, select the option called `vpixmgmt`. Your screen will look similar to this:

VP/ix MANAGEMENT

```

1  addvpixuser      enable users to use the VP/ix Environment
2  delvpixuser      disable users from using the VP/ix Environment
3  ddainstall       install a DDA device into the VP/ix Environment
4  ddaremove        remove a DDA device from the VP/ix Environment
5  vconfig          change device parameters and reconfigure VP/ix

```

Enter a number, a name, the initial part of a name, or
? or <number>? for HELP, q to QUIT:

Select option 3, `ddainstall`. This program prompts for a logical name for the device. It then displays a menu that allows you to specify the IRQ level, I/O range, and controller memory range used by the device, and the name of the UNIX System special device file associated with the device. If the device uses interrupts, I/O ports, or contains controller memory, you *must* specify this information here, otherwise you will not be able to access the device in the VP/ix Environment.

Once you have specified the device's configuration parameters and have finished going step-by-step through the procedure, the device may be accessed by VP/ix users. See section 6.4 for further information.

6.4 Making Nonstandard Devices Accessible to VP/ix Users

Once an IEM or DDA device has been configured into the VP/ix Environment as described above, individual users may access it by placing its logical name in their `vpix.cnf` files. In the case of DDA devices, only the logical name is required. For example, if you have added a networking card to your computer, with the logical name `NETCARD`, add the bottom line to your `vpix.cnf` file:

```

;
; VP/ix Configuration File
;
;
; DOS DEVICE                UNIX DEVICE PATH NAME
;
C                            /usr/vpix/defaults/C:
ROM                          /usr/vpix/defaults/rom
EGAROM                        /usr/vpix/defaults/romega
VGAROM                        /usr/vpix/defaults/romvga
CMOS                          /usr/vpix/defaults/cmos
BOOTIMAGE                    $HOME/vpix/vpix.img
NETCARD

```

- Certain IEM devices may require an additional parameter to be specified; consult the documentation supplied with the device for details.

6.5 Modifying the Configuration Parameters for Nonstandard Devices

Sometimes it is necessary to modify the configuration parameters of nonstandard devices (perhaps because they conflict with those of a new device, and it is more convenient to change the old device). Under these circumstances the `vconfig` command, which may be accessed via the `sysadm vpixmgmt` option, may be used to redefine the device's parameters. See the `vconfig(1)` manual entry for full details of this command.

6.6 Removing Nonstandard Devices From the VP/ix Environment

You may remove IEM or DDA devices temporarily or permanently. If you want to make a device temporarily unavailable to VP/ix users, it is not necessary to remove it completely from the VP/ix Environment. Use the `vconfig` option from the VP/ix Management submenu (see section 6.5) to set the value of the Configuration Flag to N. Devices that have been “disabled” in this way may be re-enabled in the future simply by using `vconfig` to set the value of the Configuration Flag back to Y.

To permanently remove an IEM device from the VP/ix Environment, use the `removepkg` option of the `sysadm` Software Management submenu. It will guide you step-by-step through the procedure. Refer to “INTERACTIVE UNIX Operating System Maintenance Procedures” for more information.

To permanently remove a DDA device, use the `ddaremove` menu option from the `sysadm` VP/ix Management submenu. Enter the logical name of the device you wish to remove and the removal will proceed automatically.

Once either type of device has been removed from the VP/ix Environment, it may be physically removed from the computer by following the manufacturer's instructions.

Appendix A: USING VP/ix ON SERIAL TERMINALS

1. INTRODUCTION

The VP/ix Environment runs on many serial terminals. However, if you want to add an unsupported serial terminal to the VP/ix Environment or alter the default mappings in the terminal configuration file, you should read this appendix; otherwise, you may skip this section.

To find out whether your terminal is supported by the VP/ix Environment, check the `/usr/vpix/term` directory. This directory contains a file for each supported terminal. For example, the file for the VT100* terminal is `/usr/vpix/term/vt100`.

1.1 Terminal Requirements

Any terminal that you add to the VP/ix Environment *must* have these basic features:

- A display of at least 24 rows and 80 columns.
- Direct cursor addressing, that is, there must be an escape sequence that allows the cursor to be moved to an arbitrary row and column on the screen.
- A non-autowrap mode in which writing a character in column 80 does *not* automatically advance the cursor to column 1 of the next row.
- Escape sequences for line insertion and deletion *or* for setting the scrolling region and reverse scrolling (the VT100 terminal, for example, has these escape sequences).
- An escape sequence for clearing the screen.

Note that most terminals are equipped with these features.

The following features are not necessary, but are highly desirable for running VP/ix because they add significantly to the illusion of being logged into a PC console:

- A PC-compatible mode that can be entered and exited via escape sequences sent from the computer. This feature allows the entire PC character set to be displayed. A terminal with a PC-compatible mode also provides a 25-line display, a keyboard that is identical to a PC keyboard, and all of the screen attributes available on a PC.

- A number of “extra” keys, such as function keys and cursor arrow keys, in addition to the basic ASCII set. Extra keys are particularly important in the absence of a PC-compatible mode because they allow convenient mapping to PC keyboard keys.
- A 25-line display. A standard PC display has 25 lines, and many PC applications use this feature. On standard terminals with only 24 lines, support is provided for “rocking” the terminal screen up and down to show either the first or the last 24 lines of the true screen.
- Screen attributes, such as bold intensity, blinking, reverse, underline, and invisible modes. These allow a variety of applications that use various screen attributes to be displayed correctly.

1.2 Before You Begin

This appendix assumes that you have a thorough understanding of the following:

- The ASCII character set (refer to the *INTERACTIVE SDS Guide and Programmer's Reference Manual*).
- The `terminfo` database (refer to *terminfo(4)* in the *INTERACTIVE SDS Guide and Programmer's Reference Manual*).
- Terminal escape sequences (refer to *terminfo(4)* and the terminal hardware manual provided with your terminal).

Be sure that you have read and understood the appropriate documentation before you attempt to add a serial terminal to your VP/ix Environment.

2. WHERE TERMINAL DESCRIPTIONS ARE LOCATED

The VP/ix Environment uses the standard UNIX System shell variable `TERM` to determine a user's terminal type. Four sources are scanned for the necessary terminal information. Information specified in a later source overrides information collected from an earlier source. The sources are given below, in the order in which they are scanned.

1. The first source scanned is the standard UNIX System `terminfo` database. This is a UNIX System database that describes what a terminal is and contains files for each supported terminal.
2. The second source scanned is the VP/ix terminal-independent configuration file, `/usr/vpix/term/sys.default`. This file provides some default, VP/ix-specific terminal information.
3. The third source scanned is the VP/ix terminal-specific configuration file located in the directory `/usr/vpix/term`. The name of this file is the same as the value of the user's `TERM` variable. This file provides a specific terminal description. If no such file is found, the file `generic` is used. This file provides a set of default keyboard mappings, output character mappings, and special key definitions usable on even the simplest terminals. These mappings are described in "Appendix C: USING ASCII TERMINALS" of the "VP/ix and MS-DOS Primer." No special capabilities of the user's terminal will be taken advantage of.
4. The fourth source scanned is an optional, individual override file, located in the directory specified in the shell variable `VPIXTERMDIR`. The name of the override file must also be the same as the value of the user's `TERM` variable. This file allows a user to override the specifications in the system-wide, terminal-specific configuration file located in `/usr/vpix/term`.

For example, if Bob's `TERM` variable has the value `vt100` and his `VPIXTERMDIR` variable has the value `/usr/bob/vpixterm`, VP/ix scans these files in order:

- /usr/lib/terminfo/v/vt100
- /usr/vpix/term/sys.default
- /usr/vpix/term/vt100
- /usr/bob/vpixterm/vt100

The last three files listed have a similar format, which is described in section 3 of this appendix.

The UNIX System file in the `terminfo` directory is scanned using the standard UNIX System curses library. (Refer to *curses(3X)* for more information.) The `terminfo` database contains information about various types of terminal capabilities (refer to *terminfo(4)* for details). If a file for the user's terminal is not contained in this database or if the `terminfo` database is not installed on the system, you must define these capabilities either in the VP/ix terminal-specific configuration file or in the individual's override file. section 3 of this appendix lists the capabilities that must be defined for a terminal in the VP/ix Environment.

3. HOW TO DESCRIBE TERMINALS

The VP/ix Environment must be provided with three types of information about a terminal:

- Escape sequences or other information concerning terminal capabilities.
- Mappings from ASCII input sequences to scan code sequences. (Scan codes are discussed in section 3.2 of this appendix.)
- Mappings from PC output characters to ASCII sequences.

Information on escape sequences is contained in the UNIX System `terminfo` files, which is why the appropriate `terminfo` file is scanned by VP/ix. However, it is possible to specify all of the information needed by VP/ix in the VP/ix terminal configuration file format. Actual examples of configuration files can be found in the directory `/usr/vpix/term`. This is an example of a simple configuration file, illustrating each of the three types of terminal information listed above:

```
*      Comment lines begin with an asterisk.
*      Tokens are separated by white space.
*      The following line defines a terminal
*      capability; it has the form "capname: definition",
*      where the definition is terminfo-compatible.
clear:  \E[H<10>      * ANSI clear screen with 10 millisecond delay

*      The following line is an input ASCII to scancode
*      mapping definition; it maps the VT100 up-arrow key
*      to the PC keyboard up-arrow scancodes.
input:  \033[-A      48-C8

*      The following line is a PC output to ASCII sequence
*      mapping.
output: \001      0      * map smiley-face to an O
```

The following sections discuss the three types of terminal information.

3.1 Terminal Capabilities

The following standard UNIX System `terminfo` capabilities are used by VP/ix. Note that only the required capabilities discussed in section 1 of this appendix *must* be specified. (These are preceded by a “+” in the following list.) However, if you wish to increase display efficiency or increase the accuracy of the PC emulation, you should specify *all* terminal capabilities in the appropriate configuration file located in the `/usr/vpix/term` directory.

	clear	clear screen and home cursor
	ed	clear to end of screen
	el	clear to end of line
+	csr	change scrolling region (vt100 only)
	ri	scroll screen down one line
	ind	scroll screen up one line
	indn	scroll screen up n lines
+	ill	insert one line (non-vt100 only)
	il	insert lines (non-vt100 only)
+	dll	delete one line (non-vt100 only)
	dl	delete lines (non-vt100 only)
	ich1	insert one character
	ich	insert characters
	dch1	delete one character
	dch	delete characters
+	cup	position cursor
	home	move cursor home
	cubl	move cursor left
	cuf1	move cursor right
	cuul	move cursor up
	cud1	move cursor down

In addition, the following capabilities are specific to VP/ix. Note that although the first six of these have the same name as a corresponding UNIX System `terminfo` capability, the definitions are slightly different. Each of these capabilities are described more fully below.

Screen Attribute Definitions

sgr0	normal mode
blink	blink mode
bold	bold mode
rev	reverse video mode
invis	invisible mode
smul	underline mode
blbo	blink+bold mode
blr	blink+reverse mode
blu	blink+underline mode
bor	bold+reverse mode
bou	bold+underline mode
blbor	blink+bold+reverse mode
blbou	blink+bold+underline mode

Svpix and Fvpix Sequences

svpix	VP/ix startup string
fvpix	VP/ix shutdown string

Special Key Sequences

k_up	up “rocker” sequence for 24-line terminals
k_down	down “rocker” sequence for 24-line terminals
k_refr	refresh sequence
pccompat	PC-compatible terminal
c_xon	XON character for PC-compatible terminal
c_xoff	XOFF character for PC-compatible terminal

3.1.1 Screen Attribute Definitions

All of the screen attributes listed may be turned on. Although the first six attributes have the same name as the corresponding UNIX System `terminfo` capability, the definitions are different. The remaining seven definitions do not exist in the UNIX System `terminfo` information. Therefore, if your terminal supports any of these 13 attributes, you should define them in the appropriate VP/ix-specific configuration file in the `/usr/vpix/term` directory.

The *blink* attribute illustrates the difference between the `terminfo` and the VP/ix way of defining attributes. The `terminfo blink` means “turn on blinking without changing any existing attributes.” Therefore, if reverse video is already on, it will remain on. However, the VP/ix *blink* means “turn on blinking and turn off all other attributes.” Therefore, if reverse video is on, it will be turned off. (On an ANSI standard terminal, the `terminfo blink` is defined as `\E[5m`, and the VP/ix *blink* is defined as `\E[0;5m`). The other 12 definitions listed operate in the same way, that is, they turn on the specific attribute(s) while turning off all other attributes.

3.1.2 The svpix and fvpix Sequences

When you enter the VP/ix Environment or run a UNIX System command from VP/ix, the *svpix* (start VP/ix) sequence is output to the terminal. When you leave the VP/ix Environment or leave a UNIX System subshell from VP/ix, the *fvpix* (finish VP/ix) sequence is output to the terminal. This allows you to specify an arbitrary set of escape sequences for turning on and off any terminal modes that are appropriate under the UNIX System but not under

VP/ix. It is suggested that the following escape sequences (or similar escape sequences) be specified in the *svpix* and *fvpix* lines of the configuration file:

- Turn off autowrap mode when entering VP/ix and turn on autowrap mode when exiting VP/ix: under VP/ix, a screen emulating a PC must not automatically wrap and/or scroll when a character is written in column 80. It is only necessary to specify this escape sequence if the autowrap mode on your terminal is normally on.
- Make the cursor a blinking underscore upon entering VP/ix (if you wish to emulate the cursor on a PC). Make the cursor a solid block when exiting VP/ix.
- Upon entering VP/ix, turn on PC-compatible mode if you are using a PC-compatible terminal. (This escape sequence must be set if you are using a PC-compatible terminal.) Turn off PC-compatible mode when exiting VP/ix.
- Reset all attributes to your normal UNIX System setup upon exiting VP/ix.

Here is an example of the *svpix* and *fvpix* lines contained in the *wyse* configuration file:

```
svpix:          \E'7\E'5$<3000>\EvI\E.3
fvpix:          \Ev4$<6>\Ed*\EO$<6>\Ee(\E'5
```

The *svpix* string specifies “make the default attribute dim, go into PC-compatible mode, wait 3 seconds, turn enhance mode on, make cursor a blinking line.” The *fvpix* string specifies “go out of PC-compatible mode, wait 6 milliseconds, set autopage off, set autoscroll off, wait 6 milliseconds, set screen to 24 lines, make cursor a blinking block.” Note that in this case there is no need to specify autowrap mode off in the *svpix* line, because turning on PC-compatible mode does this automatically.

3.1.3 Special Key Definitions

Three special ASCII sequences may be defined for terminals that are not PC-compatible. Each special ASCII sequence specifies a particular function to be performed when VP/ix receives the corresponding sequence. For the sake of convenience, the special sequences should correspond to the sequences sent from the terminal when a single special key on the keyboard is pressed. (A special key is any key other than those on the basic ASCII keyboard.)

The sequences *k_up* and *k_down* are used to support 24-line terminals. Defining either of these capabilities tells VP/ix that the terminal has only 24 lines and defines a key or keys to “rock” the screen up or down, showing either the first or last 24 lines of the true 25-line display. Note that *k_up* and *k_down* may have identical definitions (or only one of them may be defined); this means that the same key may be used to rock the screen up or down. Alternatively, *k_up* and *k_down* may be mapped to two different keys. One key will be used to rock the screen up, and another key will be used to rock the screen down. In this case, the bell will ring if the key mapped to *k_up* is used when the screen is already up or the key mapped to *k_down* is used when the screen is already down.

Note that normal scrolling operations on a 24-line terminal should be performed with the screen rocked up, that is, with the 25th line showing. Having the screen rocked down slows scrolling and leaves the 25th line, which will contain the prompt when scrolling is completed, invisible.

The *k_refr* capability allows the terminal screen to be refreshed if it is disrupted due to output from extraneous processes (for example, a system message) or buffer overflows on a busy system. If the *k_refr* capability is defined, the screen can be refreshed by pressing the key to which *k_refr* has been mapped.

3.1.4 *pccompat*

The *pccompat* capability tells VP/ix whether your terminal is PC-compatible so that VP/ix knows whether to expect *scan codes* or ASCII characters from the keyboard. (Scan codes are numerical values that correspond to the position of the key on the keyboard.) Give *pccompat* the value *y* for PC-compatible terminals; otherwise, do not specify the *pccompat* capability.

3.1.5 *c_xon* and *c_xoff*

PC-compatible terminals may have alternate characters defined to act as the XON/XOFF handshaking characters, since sending the normal ASCII characters would conflict with the scan codes that coincide with those characters. If your terminal has alternate characters defined for handshaking, you must set *c_xon* and *c_xoff* to these one-character values.

3.2 ASCII Input to Scan Code Mappings

On terminals that are not PC-compatible, the keyboard transmits the ASCII values of the keys pressed. In the case of special keys, sequences of ASCII characters may be transmitted. PCs and PC-compatible terminals, on the other hand, transmit scan codes. (One scan code, called a “make” code, is transmitted when the key is depressed, and another scan code, called a “break” code, is transmitted when the key is released.) For example, the ASCII a key, when shifted, transmits (octal) 101, the ASCII code for A. A PC application expects to see the (hex) scan codes 2A, 1E, 9E, and AA in that order, corresponding to the make code for **SHIFT**, the make and break codes for the a key, and the break code for **SHIFT**. Therefore, there must be a mapping between ASCII input characters and the sequence of scan codes that a PC application expects to receive.

`/usr/vpix/term/sys.default`, the VP/ix terminal-independent configuration file, contains the ASCII-to-scan code mappings for all single ASCII characters, so these standard mappings need not be defined elsewhere. Note that if your terminal is PC-compatible, these input mapping specifications have no meaning (but do no harm) since VP/ix assumes that it is already receiving scan codes from the keyboard.

There are, however, a number of keys on a PC keyboard that do not correspond to any ASCII character: the function keys, the three toggle keys (**CAPS-LOCK**, **NUM-LOCK**, **SCROLL-LOCK**), the nine-key numeric keypad, **INS**, **DEL**, **SYS-REQ**, **PRT-SC**, **-**, and **+** on the numeric keypad, **ALT**, **CTRL**, **LEFT-SHIFT**, and **RIGHT-SHIFT**. The last three do not transmit a character but, instead, modify the character transmitted by other keys. In order to communicate with all PC applications, it must be possible to “tell” VP/ix 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 these keys have been pressed (make code) and when they have been released (break code).

The `input:` lines in the terminal configuration file allow arbitrary mappings between sequences of input ASCII characters and sequences of scan codes. The syntax of an input mapping line is:

```
input:    ASCII-sequence    scancode-sequence
```

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 three octal digits). The scan code sequence is a series of two-digit hex scan codes separated by dashes (-).

If, for example, your terminal is extremely simple and has no extra keys whatsoever, you may decide to map particular PC keys to an ASCII sequence that begins with a **CTRL** z and has a second character that is a mnemonic for the PC key. Part of this mapping might look like this:

ASCII Sequence	Scan Codes
input: \032-1	3B-BB
input: \032-2	3C-BC
input: \032-3	3D-BD
input: \032->	4B-CB
input: \032-\032	1D-2C-AC-9D

The first line specifies that when a user types **CTRL** z (octal 032) followed by a 1 on an ASCII terminal, it will have 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. The next two lines allow a user to emulate F2 and F3, respectively. The fourth line maps the cursor-right key on the PC keyboard (the 6 on the numeric keypad) to **CTRL** z followed by > 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 “normal” scan code sequence, it is necessary to provide a sequence that does transmit its normal scan codes. This is illustrated in the fifth line above, which specifies that typing **CTRL** z twice will send the scan codes for the PC **CTRL** make code, the z make and break codes, and the **CTRL** break code.

It is important to note that the ASCII sequences above are completely arbitrary, whereas each scan code corresponds to a particular key on the PC keyboard that is to be emulated. If your ASCII terminal has a number of extra keys that normally transmit a sequence of ASCII characters (typically starting with ESC, octal 033), you can map that ASCII sequence to a corresponding key on the PC keyboard. In this case, you must also provide a sequence for actually typing ESC, which is used by many PC applications.

Note that since the characters -, \, and * are special characters in a terminal configuration file, you must enter them using their octal representation if they are to be used as actual characters in an ASCII input sequence. Sample mappings may be found in the VP/ix terminal configuration files located in `/usr/vpixon/term`.

3.3 Output Character Mappings

In order to support serial terminals, it is necessary to allow mappings from each of the 256 characters in the PC character set to the output character set of your terminal. In addition, even PC-compatible terminals require that control characters, namely octal 000 thru 037 and 177, be somehow escaped if one desires to display the actual PC character for that code.

In a VP/ix terminal configuration file, `output:` lines are used to perform this mapping. The syntax of an `output:` mapping line is:

```
output:  character character-sequence
```

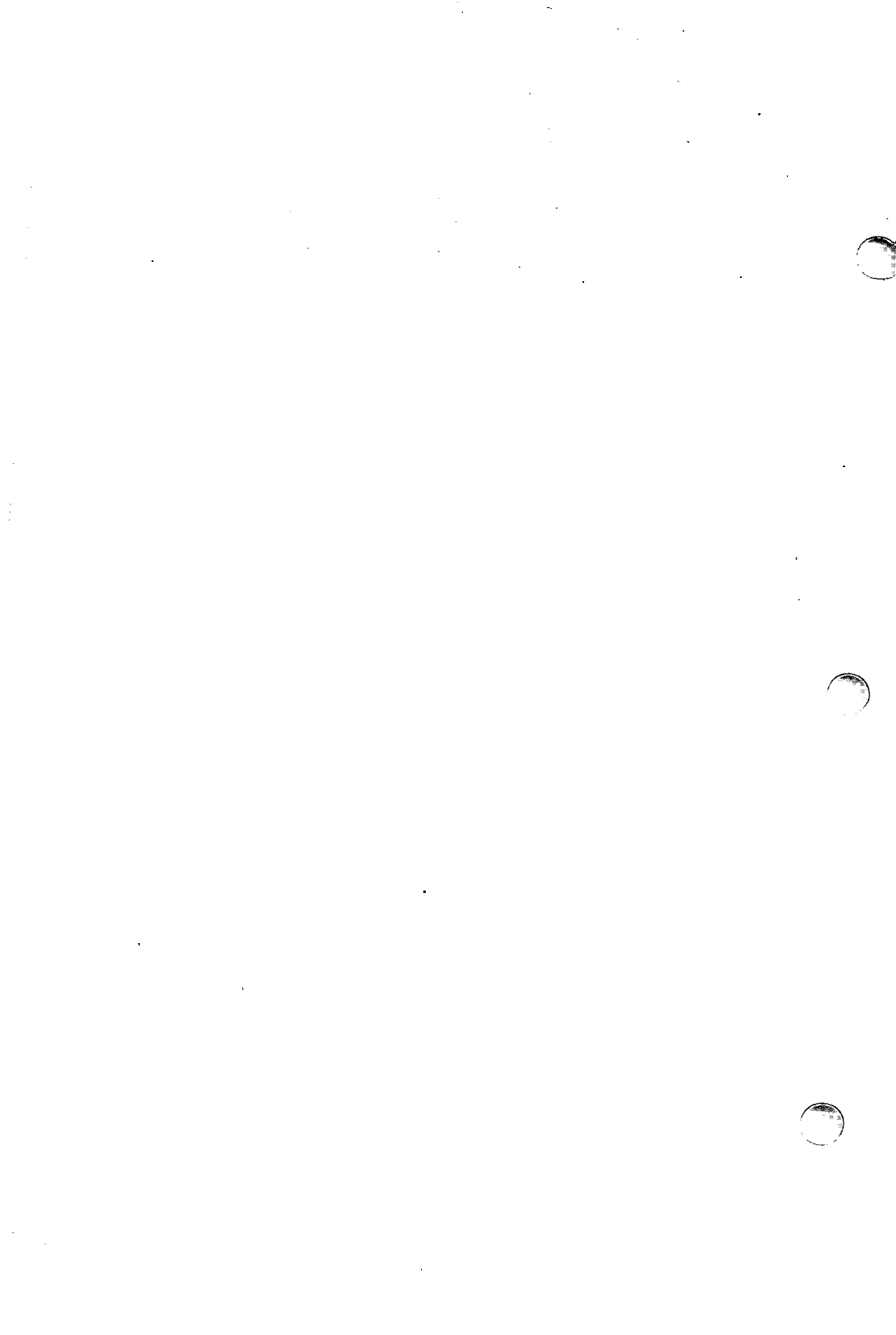
The character sequence has the same syntax as an ASCII character sequence in an `input:` mapping line, except that the characters \200 thru \377 make sense in this context, but not in an input context. The character is also the same as the ASCII sequence in an `input:` line but consists of a single character rather than a character sequence.

The default output mappings contained in `/usr/vpixon/term/sys.default` simply map every character to itself, except for the control characters mentioned above. VP/ix maps each of these control characters to ESC (octal 033) followed by the octal equivalent of the control character 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.

If your terminal is not PC-compatible, you must decide what to map each of the characters 001 thru 037 and 177 thru 377 to, since they are typically not present in the character set of an ASCII terminal.

As with `input:` lines, a great deal of flexibility is allowed in specifying `output:` lines. For example, you might map all of the above characters to a period (.), or you might try to find the ASCII character that looks most like the PC character (see a PC reference manual for pictures of the PC character set). If the ASCII terminal

has a graphics mode that supplies some of these characters, you can map the PC character to the escape sequence for getting into graphics mode, followed by the graphics character, followed by the escape sequence for getting out of graphics mode. This is a more costly procedure, in terms of time and processing, but since most of the characters output by a PC application are regular ASCII characters, the cost may be acceptable.



Appendix B: VP/ix ERROR MESSAGES

1. INTRODUCTION

This section describes the three categories of messages displayed by the VP/ix Environment when an error occurs. The possible causes of each error are indicated (cause) and recommendations for resolution are made (action). Messages having the same cause and action are grouped together. Otherwise, messages are presented alphabetically within categories. The VP/ix errors can be loosely grouped into three categories:

- **Configuration errors**, which result from an improper environment or a lack of system resources. These errors may be resolved by making the appropriate changes in the VP/ix user or system environment or in the surrounding UNIX System environment.
- **Faulty program errors**, which result from user programs that attempt to access or control reserved parts of the system or to do something that is illegal even in a native DOS environment. Fixing the offending program typically solves these problems.
- **VP/ix internal errors**, which usually occur when the VP/ix program is being executed with a UNIX System kernel that does not support the VP/ix Environment. Contact your VP/ix vendor if you encounter one of these messages.

If the VP/ix Environment displays an error message that is not listed in this appendix and that is not self-explanatory, contact your VP/ix vendor.

2. CONFIGURATION ERRORS

A configuration error indicates a problem in the usage of VP/ix or a problem in the surrounding UNIX System environment. The VP/ix program interfaces with the UNIX System environment by making a set of requests (called system calls) to the UNIX System kernel. Most of the error messages listed in this section are a result of failed system calls. Under normal circumstances, system calls should succeed. However, if the UNIX System environment is improperly configured, these error messages are likely to occur.

Can't execute *progname*: *unix_error*
Exec failed *unix_error*

Cause: An attempt to execute a UNIX System program from within the VP/ix Environment has failed. This error message indicates that there may be a problem with the UNIX System variable SHELL or the program `/bin/sh`, which is used to execute the program.

Action: Exit VP/ix and make sure that the `/bin/sh` file exists and that the variable SHELL is set to the name of a UNIX System shell program or that it is not set at all. Other possible causes for this error message are indicated by the value *unix_error*, which indicates the UNIX System error number of the failed `exec` system call. Refer to *exec(2)* for a description of these values. If the *unix_error* is Not enough space, you may need to increase the amount of swap space on the UNIX System. Refer to the appendix in the "INTERACTIVE UNIX Operating System Maintenance Procedures."

Cannot open file for saving virtual PC state

Cause: The user has attempted to run a DOS command from the UNIX System or to quickboot VP/ix using the `vpix -r` command. The VP/ix program has attempted to create the BOOTIMAGE file with the name specified in the user's configuration file but the *open* system call has failed. This error message usually indicates a problem with access to or permissions on the file.

Action: Make sure that the directory in which the BOOTIMAGE file is to be created exists and that it

has *write* permission for the user. Refer to *open(2)* for more information about the possible causes of this error message.

Cannot open /usr/vpix/etc/vpixdevs file

Cause: The user has a nonstandard device activated in his or her *vpix.cnf* file, but VP/ix is unable to open the *vpixdevs* file to obtain information about it.

Action: Check to see that */usr/vpix/etc/vpixdevs* exists and has *read* permission for the user.

Cannot open virtual terminal *filename*

Cause: The VP/ix program has attempted to access another virtual terminal on the console screen, but the system call has failed. Typically, this error message occurs when the permissions for the UNIX System virtual terminal device file *filename* are incorrect or the file has been deleted.

Action: If the file with the given *filename* does not exist, contact your INTERACTIVE UNIX System vendor for information on how to recreate it. If the file does exist, make sure it has the permissions 666 (read and write for all users).

NOTE: This error does not occur on MULTIBUS systems.

Cannot write file for saving virtual PC state

Cause: The VP/ix program has attempted to save a *BOOTIMAGE* file with the name specified in the user's configuration file. A *write* error occurred while writing out the VP/ix image.

Action: Make sure that the file system to which this file is being written contains enough space (the VP/ix image is over 100K bytes long). Also, make sure that the user's *ulimit* value is large enough to handle a file of this size. Refer to *ulimit(2)* for information about setting the *ulimit* value. Refer to *write(2)* for more information about other causes of this error message.

Couldn't get info on device *name* from /usr/vpix/etc/vpixdevs file

Cause: The user has specified a logical device *name* in their VP/ix configuration file, *vpix.cnf*, which was not found in the VP/ix device definition file.

Action: Check for typographical errors and ensure that the device specified has been correctly configured into the VP/ix Environment with the *vconfig* utility.

DMA conflict between device *name1* and *name2*

Cause: An attempt has been made to use device *name1*. VP/ix has detected a potential DMA conflict between this attempt and the use of the existing device *name2*.

Action: Resolve conflict by using only one device at a time or reconfiguring the hardware to avoid the conflicts.

Error *unix_error* encountered opening file *filename* as drive *drive_letter*

Cause: While attempting to open the UNIX System file *filename*, which is mapped to the DOS drive *drive_letter* in the user's configuration file, VP/ix encountered the *unix_error* named in the error message. The entry for the *drive_letter* in the user's configuration file may be set to a file that does not exist, or the permissions on the specified file may not allow *read* access. This error message may also be generated if the user attempts to access the file when it is locked by another user, since only one user at a time may open certain files.

Action: Make sure that the file exists and that the *read* permission on it is set appropriately. Refer to *open(2)* for other possible causes of this error message.

Error *unix_error* opening ROM file *filename*

Cause: The ROM or EGAROM file *filename* specified in the user's configuration file could not be opened. The user must have *read* permission on these files.

Action: Make sure the file exists and that *read* permission on it is set appropriately. Refer to *open(2)* for other possible causes of this error message.

Error *unix_error* opening configuration file *filename*

Cause: The VP/ix configuration file could not be opened. It is also possible that the configuration file was found but did not have the necessary *read* permission set for the user.

Action: Make sure that the user's home directory contains a *vpix/vpix.cnf* file or that the file named in the user's *VPIXCNF* variable exists. Also make sure that the *read* permission on the configuration file is set appropriately. Refer to *open(2)* for other possible causes of this error message.

Error *unix_error* opening mouse file *filename*

Cause: The VP/ix program is unable to access the mouse device as specified in the user's configuration file. This error message may also be generated if the user attempts to access the device when it is already locked by another user, since only one user at a time may access the device.

Action: Make sure that the *MOUSE* entry in the VP/ix configuration file is set to an existing UNIX System device file and that the permissions on the specified UNIX System device file provide *read* access to the user. If another user has accessed the *MOUSE* device, wait until the device is released before attempting to access it again. Refer to *open(2)* for other possible causes of this error message.

NOTE: Mouse entries should not be defined on *MULTIBUS* systems.

Error *unix_error* opening terminal configuration file *filename*

Cause: The terminal configuration file *filename* specified by the user's *TERM* environment variable could not be opened. If the *unix_error* is *No such file or directory*, then the terminal configuration file *filename* does not exist. This error message may be generated when the *TERM* environment variable is set

to a terminal type that is not supported by the VP/ix Environment.

Action: You must create a terminal configuration file for this terminal before VP/ix can be run on it (refer to Appendix A). If the *unix_error* is Permission denied, then the permissions on the terminal configuration file *filename* do not allow the user to read it. Make sure that the permissions on the configuration file provide *read* access to the user. Refer to *open(2)* for other possible causes of this error message.

Error attempting to open serial device

Cause: The serial device entry (COM1, COM2, COM1MOUSE, or COM2MOUSE) specified in the user's configuration file could not be opened. This error message may occur when one of the entries in the VP/ix configuration file is set to a file that does not exist or when the file exists but does not have the necessary *read* or *write* permissions set for the user. This error message may also occur if the device is already being accessed by another user, since only one user at a time may access a COM1 or COM2 device.

Action: Make sure that the serial device entry in the configuration file is set to an existing file and that the file permissions allow the user *read* and *write* access. If the device is already being accessed by another user, wait until the device is released before attempting to access it again. Refer to *open(2)* for other possible causes of this error message.

NOTE: Serial device entries should not be defined on MULTIBUS systems.

Fork failed *unix_error*

Unable to fork new process: *unix_error*

Cause: An error occurred when an attempt was made to execute a shell. The value *unix_error* indicates the UNIX System error number of the failed *fork* system call.

Action: If the *unix_error* is No more processes, you may need to increase the number of UNIX System processes allowed on your system. If the *unix_error* is Not enough space, you may need to increase the amount of swap space on your UNIX System. Refer to the appendix in the “INTERACTIVE UNIX Operating System Maintenance Procedures.” Refer to *fork(2)* for other possible causes of this error message.

Interrupt conflict between device *name1* and *name2*

Cause: An attempt has been made to use device *name1*. VP/ix has detected a potential interrupt conflict between this attempt and the use of an existing device *name2*.

Action: Resolve conflict by using only one device at a time or reconfiguring the hardware to avoid the conflicts.

Invalid command line parameter: *arg*

Cause: The user has invoked the *vpix* program with an invalid argument *arg*.

Action: Refer to *vpix(1)* for information about valid arguments to the *vpix* program.

line *line_number*: input sequence too long
(max: *max_num*)

Cause: An input key mapping in the terminal configuration file has too many input characters.

Action: You must shorten the number of input characters on the specified *line_number*. If this input character sequence corresponds to what a single key on the terminal sends, it will not be possible to provide a mapping for this key on the terminal. The maximum number of input key mappings is indicated in *max_num*. Refer to Appendix A for more information about mapping input sequences.

I/O register conflict between device *name1* and *name2*

Cause: An attempt has been made to use device *name1*. VP/ix has detected a potential I/O register conflict between this attempt and the use of an existing device *name2*.

Action: Resolve conflict by using only one device at a time or reconfiguring the hardware to avoid the conflicts.

**line *line_number*: scancode sequence too long
(max: *max_num*)**

Cause: An input key mapping in the terminal configuration file has too many scan codes mapped to it.

Action: You must shorten the number of scan codes on the specified *line_number*. The maximum number of scan code mappings is indicated in *max_num*. Refer to Appendix A for more information about mapping input sequences.

**line *line_number*: too many mappings
(max: *max_num*)**

Cause: The terminal configuration file has too many input key mappings. These are lines that begin with the string **input:** and indicate key sequences to be mapped to scan code sequences.

Action: You must delete some of these input key mappings in order to run VP/ix on this terminal. The maximum number of input key mappings is indicated in (max: *max_num*). Refer to Appendix A for more information about mapping input sequences.

**line *line_number*: unknown terminal capability
*capname***

Cause: A line in the terminal configuration file begins with an unknown keyword *capname*. Only the capabilities listed in Appendix A may be specified.

Action: Change the entry in the configuration file to match one of the capabilities listed in Appendix A.

Memory conflict between device *name1* and *name2*

Cause: An attempt has been made to use device *name1*. VP/ix has detected a potential memory conflict between this attempt and the use of an existing device *name2*.

Action: Resolve conflict by using only one device at a time or reconfiguring the hardware to avoid the conflicts.

No virtual terminals available

Cause: The VP/ix program has attempted to access a new virtual terminal but the attempt has failed.

Action: You must terminate one or more processes running on the virtual terminals before additional virtual terminals can be used. Log out of a virtual terminal, or use the `sysadm virterm` command to decrease the number of active virtual terminals.

NOTE: This error does not occur on MULTIBUS systems.

Printer put error

Cause: An error occurred while data was being sent to the printer process specified in the LPT1, LPT2, or LPT3 entry of the user's configuration file. This error message may be generated by an unrecoverable disk error.

Action: Refer to `write(2)` for more information about `write` errors.

TERM variable is not set

Cause: The TERM variable has not been set. In order for VP/ix to run on a terminal other than the console, the environment variable TERM must be set.

Action: Make sure that the user's TERM variable is set to the type of terminal being used and that the terminal has been exported. This is typically done in the user's `.profile` file.

Terminal configuration file *filename* has errors:

Cause: A problem has been detected with the structure of the terminal configuration file *filename*. The subsequent error messages will specify the problem.

Action: The action to be taken will depend on the specific error message.

Unable to complete ioctl

Cause: The built-in Direct Device Attachment (DDA) mechanism is attempting to gain direct access to a device specified by the user. However, the UNIX System has found that permitting this action will cause a conflict with another device in the system. VP/ix will be denied its request and the device will not be accessible.

Action: The administrator should ensure that the DDA device is not being accessed by another program. All attached peripherals on the system should then be checked for conflicts and those conflicts should be resolved.

Unable to open generic device *name*

Cause: The user has an entry with a given *name* in his or her VP/ix configuration file, which VP/ix is unable to open. This entry corresponds to a Direct Device Attachment (DDA) device, which requires a UNIX System special file in the */dev* directory.

Action: Generally, the device cannot be opened either because the special file in the */dev* directory does not exist or because the permissions on that entry do not allow access to the device. Also, if the device is being used by another user, this error message will appear and is considered a normal consequence of attempting to share a DDA-type device. The other user must exit VP/ix before this user will be able to enter VP/ix.

v86init() failed: *unix_error*

Cause: The system call used to initialize the VP/ix Environment has failed. If the *unix_error* is Not owner, the failure occurred because the *vpix* program

(`/usr/bin/vpix`) is not owned by `root` or does not have the permissions `4755`.

Action: Use the UNIX System `chown` and `chroot` commands to change the owner or the permissions, as appropriate. If the `unix_error` is No more processes, you must allocate more swap space for your UNIX System. Refer to the appendix in the “INTERACTIVE UNIX Operating System Maintenance Procedures.”

VPIX not linked to include IEM module for device *name*

Cause: The user has specified a logical device *name* in the VP/ix configuration file, `vpix.cnf`, which contains an entry in the VP/ix device file. However, VP/ix has not been correctly configured (built) to include support for the specified device.

Action: Check that the device has been correctly configured into the VP/ix Environment with the `vconfig` utility.

3. FAULTY PROGRAM ERRORS

The VP/ix Environment provides compatibility with standard DOS devices and functions. However, there are certain “ill-behaved” applications that use nonstandard devices or port addresses. These applications will usually cause errors in the VP/ix Environment and should be avoided. A user-written DOS program that is faulty (for example, one that attempts to divide by zero) and would normally cause a problem under native DOS will also cause one of the messages listed below to be generated. In this case, the user should fix the offending program.

Application attempted to enter protected mode

Cause: This DOS application attempted to enter 80286 protected mode.

Action: This is not a true 8086 DOS application, and it is not supported in the VP/ix Environment.

Error reading serial device

Cause: An error occurred while data was being read from the serial device. This error may have been caused by a lost carrier (for example, a disconnected phone line). Another possible cause is that the DOS application has attempted to read from a serial port with a port number that is not valid.

Action: Make sure that the appropriate phone lines are connected, and make sure that the serial port is assigned a valid port number.

NOTE: COM1 and COM2 should not be defined on MULTIBUS systems.

Error writing serial device

Cause: An error occurred while data sent to the serial device was being written. This error may have been caused by a lost carrier. Another possible cause is that the DOS application has attempted to write to a serial port with a port number that is not valid.

Action: Make sure that the appropriate phone lines are connected, and make sure that the serial port is assigned a valid port number.

NOTE: COM1 and COM2 should not be defined on MULTIBUS systems.

General protection fault, cannot emulate instruction

Cause: The DOS program has caused a General Protection Fault. VP/ix is unable to emulate the specified privileged instructions. Possible causes of this error are:

- An instruction fetching or storing a word at location 0xffff (65535), which would cause the next address to become too large.
- An instruction with an “addr32” (0x67) prefix in front of it, causing an address that is bigger than 0xffff (65535).
- An attempt to execute privileged instructions.
- An attempt to execute an 8086 “pop cs” instruction. This instruction was removed from Intel processors after the 8086, and cannot be emulated in VP/ix.
- Too many instruction prefixes in front of an instruction.
- Any other reason that would cause a General Protection Fault in a Virtual 8086 Mode process.

Action: Make sure that the offending program runs correctly under native DOS. If it does not, then fix the program. Refer to the *INTERACTIVE SDS Guide and Programmer's Reference Manual* for more information about the possible causes of this error.

Int13(): Function = *functioncode*

Cause: An attempt has been made to read or write input or output to a DOS file system image with the `int 13` instruction in a way that is not supported by VP/ix. The value of *functioncode* gives the hexadecimal value of the AH register for the `int 13` call.

Action: This program cannot be run under VP/ix.

Invalid interrupt service routine number

Cause: The DOS application has attempted an `int 1` (trace trap), `int 3` (breakpoint), or `int 4` (overflow trap) instruction that VP/ix has been unable to emulate correctly.

Action: Make sure that the offending DOS program runs correctly under native DOS. If it does not, then fix the program.

Invalid magic request *data*

Cause: The DOS process has attempted to write to a VP/ix reserved port. The hexadecimal value *data* indicates the invalid data that is being output.

Action: The application must be modified so that it does not use this port.

PR_CMOS: Bad read from *port***PR_CMOS: Bad write of data to *port***

Cause: The DOS program has attempted to execute an `in` or `out` instruction at the hexadecimal value *port*. This value is in the range of the CMOS ports (0x70 to 0x7f), but it is not a valid or emulated VP/ix port.

Action: You must correct the invalid instruction.

Printer general error

Cause: An attempt has been made to read from or write to a printer or parallel port with a port number that is not implemented.

Action: You must correct the invalid input or output instruction.

Unable to emulate two-byte instruction

Cause: An 80386* two-byte opcode (one beginning with the byte 0xf) cannot be emulated. This implies that the DOS application is doing something abnormal.

Action: Verify that the program that generated the error message runs correctly under native DOS. If it does not, correct the program. If it does, report the problem to your VP/ix vendor.

4. VP/ix INTERNAL ERRORS

The VP/ix program requires special support from a customized UNIX System kernel that is supplied with your INTERACTIVE UNIX System Operating System. Most of the errors listed in this section result from using the VP/ix program with a nonstandard kernel that is not enhanced to support the VP/ix Environment. Other VP/ix internal errors are indicative of serious problems with the VP/ix program itself. These errors should be reported to your VP/ix vendor.

Can't map memory

Can't track memory: *unix_error*

Couldn't Re-track VRAM

Cause: An error was encountered while VP/ix was trying to set up access to the screen memory.

Action: Report this error to your VP/ix vendor.

Can't query VT manager: *unix_error*

Can't set scan mode for AT kbd

Can't set scan mode in kbd

Can't set virtual terminal mode: *unix_error*

Can't set virtual terminal text mode: *unix_error*

Couldn't set kbd INTTYPE

Couldn't set VT mode: *unix_error*

Couldn't get VT mode: *unix_error*

Cause: An attempt to access the UNIX System console driver has failed. A possible cause is that the UNIX System kernel does not have the proper console driver support for VP/ix.

Action: You must use a standard INTERACTIVE UNIX System kernel.

Couldn't map EGA ROM: *unix_error*

Couldn't unmap EGA ROM: *unix_error*

Cause: An error was encountered trying to set up or release access to the EGA ROM.

Action: Report this error to your VP/ix vendor.

Internal error: ect_main returned

Cause: An unexpected internal error in VP/ix has occurred.

Action: This error should be reported to your VP/ix vendor.

Error attempting to initialize mouse

Cause: An attempt to initialize the mouse device for reading has failed. A possible cause is that the mouse driver in the UNIX System kernel does not have the proper support for the mouse device.

Action: You must use a standard INTERACTIVE UNIX System kernel.

NOTE: Mouse devices are not currently supported on MULTIBUS systems.

Error attempting to send out to serial

Cause: An attempt to emulate the serial communications port on the terminal represented by the COM device has failed. A possible cause is that modifications necessary for VP/ix to run on the terminal have not been configured into the UNIX System kernel.

Action: You must use a standard INTERACTIVE UNIX System kernel.

NOTE: COM devices are not currently supported on MULTIBUS systems.

Error attempting to send serial info**Error attempting to set DOS mode****Error attempting to set non-DOS mode****Error sending serial int type**

Cause: An attempt to set the terminal representing the COM device into a special mode for VP/ix has failed. The probable cause of this error message is that modifications necessary for VP/ix to run on the terminal have not been configured into the UNIX System kernel.

Action: You must use a standard INTERACTIVE UNIX System kernel.

NOTE: COM devices are not currently supported on MULTIBUS systems.

Error setting line for serial device

Cause: An error has occurred when VP/ix attempted to get or set the terminal characteristics, such as the baud rate, from the serial device. This error may have been caused by a lost carrier.

Action: Make sure that the phone line has not been disconnected.

Map to NULL failed

Cause: An error was encountered when VP/ix attempted to set up access to a null page of memory.

Action: This error should be reported to your VP/ix vendor.

Seek(): lseek() failed, errno = *unix_error*

Cause: An error has occurred when VP/ix attempted to position to *read* or *write* a DOS file system image.

Action: Report this error to your VP/ix vendor. Refer to *lseek(2)* for other possible causes of this error message.

Segmentation violation

Cause: The VP/ix process has received a signal 11 from the UNIX System. This is either because another UNIX System process has sent this signal (for example, the *kill* command), or there is an internal error in VP/ix.

Action: First, check whether the problem is being caused intentionally by another process, by repeating the conditions that resulted in the message. If it is repeatable, contact your VP/ix vendor.

Appendix C: INCOMPATIBILITIES BETWEEN DOS AND VP/ix

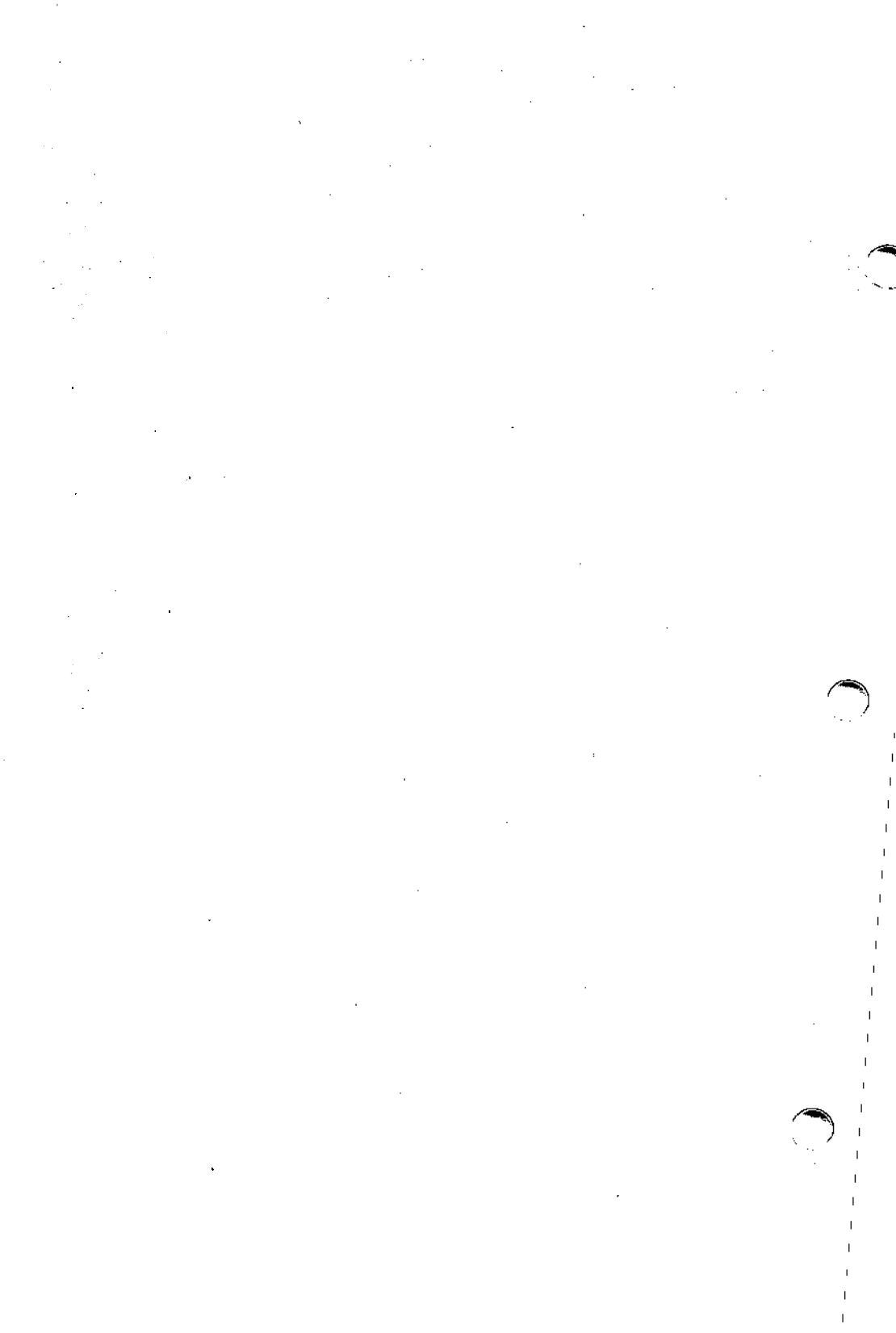
While the majority of DOS applications will run successfully under VP/ix, a small number of applications rely on functions, such as tight control of the hardware interface, that are impossible to emulate within the VP/ix Environment. Such known areas include:

- *Real-time timing dependencies* (e.g., games, music). Such applications will run properly, but due to the fact that in reality they are running under a multi-tasking environment (i.e., the UNIX System), they may appear discontinuous or slow if other users are busy on the system.
- *Heavy programming of the DMA controller chip* (as is done by Fastback*) and unusual or undocumented features of other chips that are emulated under VP/ix.
- *Access to nonstandard PC hardware* (e.g., tape drives, network adapters, nonstandard display adapters, synchronous communications or RS-422 adapters, scanners, or CD-ROM drives). If such devices are not configurable using Direct Device Access, explicit support in the form of an Installable Emulation Module must be provided by the hardware manufacturer. See section 6 for further explanation of nonstandard devices.
- *Applications that attempt to go into 80386 protected mode.* These include multi-user DOS systems such as PC-MOS and Concurrent DOS, as well as 386-specific versions of standard DOS applications such as MS Windows* 386 and Paradox* 386. Such applications truly take over the machine in a way which is incompatible with running under the INTERACTIVE UNIX System.
- *Applications or commands which attempt to directly manipulate parts of the DOS file system.* Such applications will work on the A, B, or C drives, but not on the Z drive, which is actually the UNIX file system. There are very few such applications (Norton Utilities “unerase” command is one), but there are a number of standard DOS commands that do this. Such commands should produce an error when run on the Z drive saying that they cannot work on a “networked” drive.
- *Direct access to the fixed disk controller registers;* all fixed disk access must be via BIOS or higher level calls.

- *Applications that require so much memory under DOS that they are unable to run if there are any terminate and stay resident DOS programs loaded into memory.* Under VP/ix, the file redirector is normally loaded and takes up about 20K of memory. If you encounter such an application, you may turn off the redirector using the `dosmount` command and run it, but of course you will not have access to the Z drive while the redirector is not running.
- *EGA-generated interrupts.*
- *The Softguard copy protection scheme.*
- *INT 15 extended memory calls.* Certain other BIOS INTs that deal directly with hardware (such as the real-time clock) are also not supported. There are no known applications that use these unsupported BIOS INTs.

INDEX

- A, B DOS devices 11
- altering terminal default mappings 35
- asterisk (*) 21
- AUTOFLUSH 14
- backslash 17, 21
- BIOS 10
- BOOTIMAGE 10
- characters, lowercase 21
- characters, uppercase 21
- CMOS 10
- COM1, COM2 12
- COM1MOUSE, COM2MOUSE 12
- comment lines 5
- cross-execution environment 16
- D drive 12
- dash (–) 17
- DDA 29
- device names 4
- devices, nonstandard 29
- /dev/tty00 12
- /dev/tty01 12
- Direct Device Access 29
- diskette drive, problem accessing 23
- DOS and VP/ix incompatibilities 67
- DOS application environment 19
- DOS devices A, B 11
- DOS, version of 26
- DOSPATH 16
- EGAROM 10
- environment, user 3
- Environment, VP/ix 3
- error messages, VP/ix 49
- file, .profile 16
- Flush Printer 14
- high-resolution adapter 10
- IEM 29
- incompatibilities between DOS and VP/ix 67
- Installable Emulation Module 29
- lock mechanism 3
- LOGITECH 13
- lowercase characters 21
- LPT1, LPT2, LPT3 13
- MOUSE 13
- mouse, Inport-2 Bus 13
- mouse, LOGITECH 13
- mouse, Microsoft 13
- mouse pointing device 13
- nonstandard devices 29
- PATH 16
- print spooler 13
- printer, buffer flushing 14
- printer, flushing 14
- printer problems 24
- question mark 21
- quickboot image file 10
- resource allocation 3
- run DOS programs from UNIX 19
- run UNIX programs from DOS 17
- rununix.exe 17
- VP/ix Interface Menu, problems 24
- search path 16
- semicolon (;) 5
- single quotes (") 21
- slash (/) 17
- special characters 21
- sysadm program 9
- terminal description 39
- troubleshooting 21
- uppercase characters 21
- /usr/bin/lp 13
- /usr/bin/vipx 19
- version of DOS 26
- VGAROM 10
- video adapter 10
- VP/ix and DOS incompatibilities 67
- VP/ix configuration file 4
- VP/ix Environment 3
- VP/ix Environment, configuring 3
- VP/ix error messages 49
- VP/ix Interface Menu 4
- vpix login shell 19
- VP/ix on serial terminals 35
- VP/ix serial terminal, adding 35
- vpix.cnf 4
- wildcard characters 21



NAME

csh — a shell with C-like syntax, VP/ix addendum

NEW ENVIRONMENT VARIABLE

The new environment variable described in this section has been added to the C shell.

DOSPATH The C shell will behave normally for those users who do not set **DOSPATH**. Users who wish to be able to execute DOS programs directly from the C shell, i.e., bypassing the normal DOS bootup that occurs when running *vpix*, should set **DOSPATH** to include those directories in **PATH** that contain DOS executables.

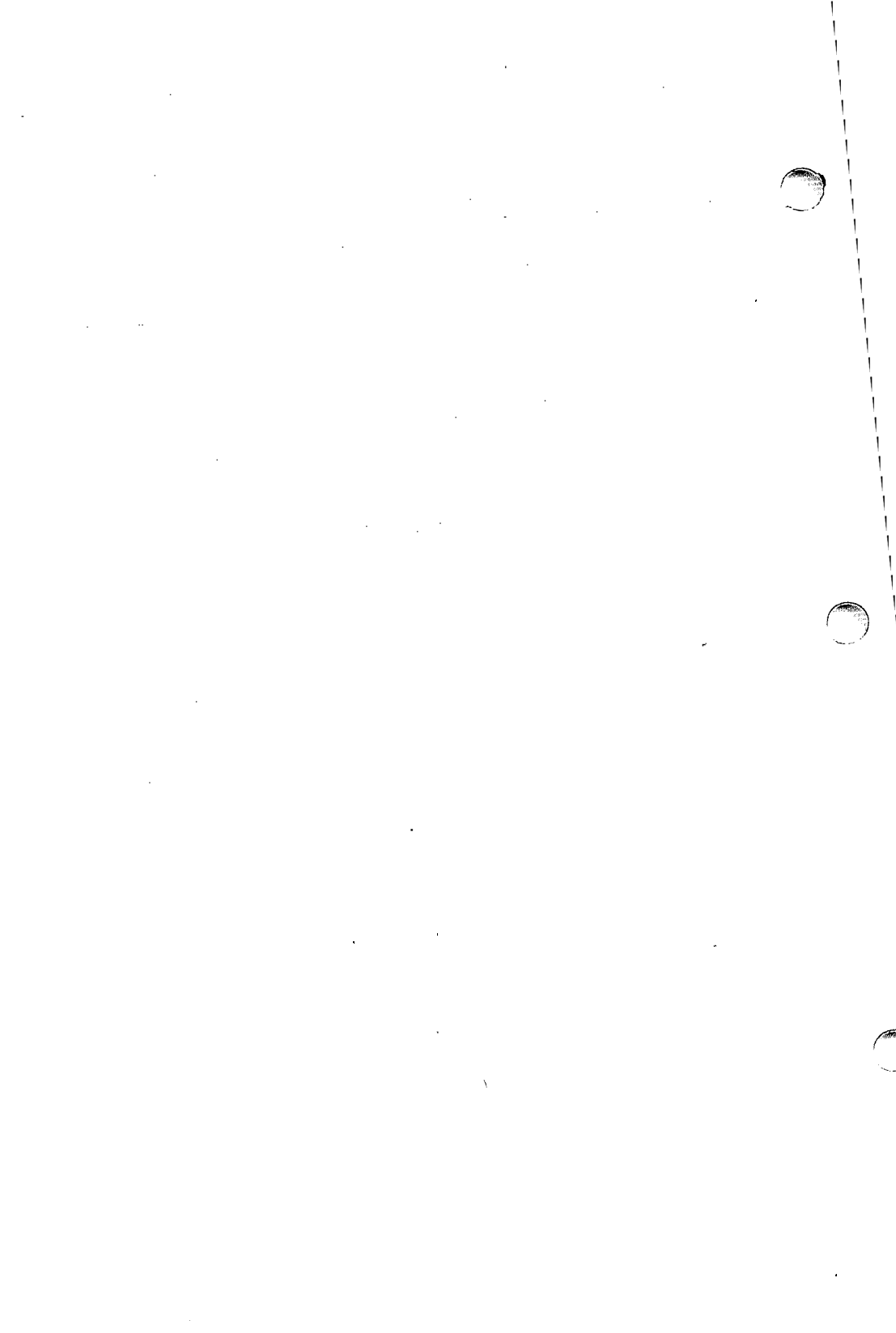
DOSPATH is a string with the same format as **PATH**; it contains a subset of the list of directories from **PATH**. When searching a directory in **PATH** for a program, the C shell determines whether that directory is also in **DOSPATH**. If it is not, the C shell acts as usual. If it is, the C shell looks first for the command with the suffix *.com*, then *.exe*, then *.bat*, and finally, for the command without any suffix. Whenever the result of a path search gives a file with one of these DOS suffixes, the shell runs the *vpix* program via the standard search path and adds arguments *-c* and the full path name of the DOS program (including the suffix).

EXAMPLE

If **PATH** is set to *./bin:/usr/bin*, **DOSPATH** is set to *.*, the current directory is */usr/john/dosbin*, and there is a DOS program named *abc.com* in the current directory, then typing *abc* to the C shell will cause the command *vpix -c /usr/john/dosbin/abc.com* to be executed, which will run the DOS program *abc.com* without the normal *vpix* DOS bootup.

SEE ALSO

vpix(1).



NAME

sh, rsh — standard or restricted shell, VP/ix addendum

NEW ENVIRONMENT VARIABLE

The new environment variable described in this section has been added to the shell.

DOSPATH The shell will behave normally for those users who do not set **DOSPATH**. Users who wish to be able to execute DOS programs directly from the shell, i.e., bypassing the normal DOS bootup that occurs when running *vpix*, should set **DOSPATH** to include those directories in **PATH** that contain DOS executables.

DOSPATH is a string with the same format as **PATH**; it contains a subset of the list of directories from **PATH**. When searching a directory in **PATH** for a program, the shell determines whether that directory is also in **DOSPATH**. If it is not, the shell acts as usual. If it is, the shell looks first for the command with the suffix *.com*, then *.exe*, then *.bat*, and finally, for the command without any suffix. Whenever the result of a path search gives a file with one of these DOS suffixes, the shell runs the *vpix* program via the standard search path and adds arguments *-c* and the full path name of the DOS program (including the suffix).

EXAMPLE

If **PATH** is set to *./bin:/usr/bin*, **DOSPATH** is set to *.*, the current directory is */usr/john/dosbin*, and there is a DOS program named *abc.com* in the current directory, then typing *abc* to the shell will cause the command *vpix -c /usr/john/dosbin/abc.com* to be executed, which will run the DOS program *abc.com* without the normal *vpix* DOS bootup.

SEE ALSO

vpix(1).



NAME

vconfig — configure, build, and install vpix

SYNOPSIS

vconfig

DESCRIPTION

The *vconfig* command provides a menu-driven mechanism for modifying the configuration parameters associated with nonstandard (IEM and DDA) devices that are defined in the file `/usr/vpix/etc/vpixdevs`. This program should be run only when no users are running the *vpix* program.

The program's top-level display lists all defined IEM and DDA devices and offers menu options to display an additional screenful of device names (in case more devices have been defined than may be displayed on a single screen), to modify the parameters associated with a device, to exit and update VP/ix, or to exit without changes.

If the user chooses to make changes, *vconfig* prompts for the logical name of the device to be modified; a second menu is then displayed, from which the user may choose one or more of the following device configuration parameters for modification:

- The range of I/O ports used by the device
- The range of controller memory used by the device
- The interrupt vector used by the device
- The DMA channel used by the device
- The UNIX System special device file associated with the device
- The device's "configuration flag" (when set to Y, the device is accessible by VP/ix users; when set to N, the device is not accessible)

Additional menus and forms then guide the user through the process of making changes to selected parameters. If the user attempts to enter parameters that conflict with VP/ix's built-in (**fixed**) devices, the changes are rejected. The program will not prohibit the entry of parameters that conflict with other DDA or IEM devices, although it does issue a warning. (VP/ix performs additional checks at runtime to ensure that a VP/ix user does not attempt to access two devices that conflict with each other.)

Ample opportunity is given to cancel or "undo" changes made in error.

When all the necessary changes have been made, the user may choose to exit without change, or to update VP/ix and exit. If the changes are retained, and modifications have been made to IEM device configuration parameters, *vconfig* rebuilds the *vpix* program, incorporating the changes as necessary.

FILES

`/usr/vpix/etc/vconfig`
`/usr/vpix/etc/vpixdevs`

SEE ALSO

"Accessing Nonstandard Devices in the VP/ix Environment" in "VP/ix Environment Maintenance Procedures."

WARNINGS

The *vconfig* command may not be used to “register” or “deregister” nonstandard devices with VP/ix, but only to modify the parameters of devices that are already registered. DDA devices are registered and deregistered by means of the *syadm vpixmgmt* menu options *ddinstall* and *ddaremove*; the corresponding menu options for IEM devices are *installpkg* and *removepkg* on the System Administration Software Management submenu. Note however that registered devices may be “installed” and “deinstalled” by using *vconfig* to toggle the value of the configuration flag.

vconfig does not prohibit the entry of parameters that conflict with other IEM or DDA devices, although it does warn of the situation. An individual user’s *vpix.cnf* file may define only one of any such conflicting devices.

NAME

vpix – create a virtual PC process

SYNOPSIS

`/usr/bin/vpix [-c command] [-r [image]] [-n] [-b]`

DESCRIPTION

vpix creates and monitors a VP/ix virtual PC, and also provides a user interface to it. The virtual PC is a special process that executes in both Virtual 8086 mode and Protected mode, and runs DOS and DOS applications in a virtual environment equivalent to an IBM PC AT running in Real mode. After finding the various files required for initialization, *vpix* creates a virtual PC memory region, loads a BIOS into it from the appropriate source, and begins execution in Virtual 8086 mode.

vpix can be invoked from any asynchronous serial terminal or from a virtual terminal on a PC AT-compatible memory-mapped system console device. DOS graphics applications can run only on the latter. The Monochrome Display Adapter is emulated on serial terminals, and only text-mode DOS applications are supported. A **TERM** variable in the environment, and a matching file in the `/usr/vpix/term` directory, are required for serial terminal support. If no matching file is found in `/usr/vpix/term`, then `/usr/vpix/term/generic` is used. If the environment variable **VPIXTERMDIR** is set, it is searched after `/usr/vpix/term`. See the “VP/ix Environment Maintenance Procedures” for the format of a VP/ix terminal description file.

Options

-c *command*

Execute the *command* as the first DOS program instead of the DOS shell (COMMAND.COM, which is the default), and terminate the virtual PC when the *command* terminates. This option is the method used by the Bourne and C shells to execute DOS commands directly from the UNIX System shell prompt.

-r [*image*]

Initialize *vpix* quickly from a file containing the image of a virtual PC. If *image* is not specified, the configuration file **BOOTIMAGE** specification (see below) is used. If the image file does not exist or is out-of-date with respect to the user's current hardware configuration, it is recreated automatically by VP/ix.

-n

Normally, if conditions such as the inability to open device files are encountered when *vpix* is invoked, a message is displayed and the user is prompted to press the **ENTER** key. If this option is specified, the warning messages will still be displayed, but the *vpix* boot will continue without user intervention. This is a useful option for running *vpix* in a noninteractive environment.

-b

Normally, *vpix* does not attempt to boot off of a DOS partition on the DOS A: device, even if it is specified in the configuration file A specification (see below). If this option

is given, it *will* attempt to boot off of A: if A is specified in the configuration file.

Configuration

vpix uses a configuration file to map virtual DOS devices to real UNIX System devices, files, or programs, and also to provide certain other information to the Emulation Control Task of the virtual PC. If the environment variable **VPIXCNF** is present when *vpix* starts up, it is taken as the full path name of the configuration file. If **VPIXCNF** is not present, then *vpix* searches for the file **vpix.cnf** first in the directory **\$HOME/vpix**, then in **/usr/vpix/defaults**.

The configuration file is an ASCII text file that consists of comment lines and configuration lines. A comment line is any line beginning with the semicolon (;) character. A configuration line consists of two fields separated by spaces or tabs. The first field is a keyword, usually the logical name of a DOS device. The second field is a corresponding value that is associated with the keyword, usually the name of a UNIX System device or file. The valid keywords are:

A, B, C, D

One of the uppercase set A–D corresponding to a DOS disk or diskette volume. No colon is required. The value may be a UNIX System device or the name of a UNIX System file containing an image of a DOS file system. These drive letters have no defaults and are emulated only if present in the configuration file. Normally, drive C is mapped to either **\$HOME/vpix/C:** or **/usr/vpix/defaults/C:**, depending on the user's requirements.

AUTOFLUSH *n*

Enable the print buffer to automatically be flushed when the printer has been idle for *n* seconds. If *n* is 0 or if the AUTOFLUSH entry is commented out, the autoflush mechanism is disabled. When AUTOFLUSH is enabled, 5 seconds is the minimum accepted value for *n*. Values specified that are less than this will default to 5. AUTOFLUSH will not be enabled if *n* cannot be read correctly.

COM*n* where *n* is 1 or 2. The value must be the name of a UNIX System asynchronous character device special file. COM2 should not be specified unless COM1 is also specified.

COM*n*MOUSE

where *n* is 1 or 2. The value must be the name of a UNIX System asynchronous character device special file to which a serial mouse is attached. COM2MOUSE should not be specified unless COM1MOUSE is also specified. If no COM*n* or COM*n*MOUSE entry is specified, the DOS serial port is not emulated.

LPT*n* where *n* is 1, 2 or 3. The value must be a string (which must be quoted if it contains spaces) that can be passed to the UNIX *system*(3) library routine. Buffered print data is passed to the resulting child process through a pipe. If there is no LPT entry in the configuration file, the DOS parallel port is not emulated. LPT2 should not be specified unless LPT1 is also

specified, and LPT3 should not be specified unless both LPT1 and LPT2 are also specified.

LMOUSE,MOUSE

The value following one of these keywords must be the name of a UNIX System character device special file, e.g., `/dev/mouse`. Use the `MOUSE` entry for a Microsoft InPort Bus Mouse; use the `LMOUSE` entry for the LOGITECH 3-Button Bus Mouse. If one of these keywords is not present, no mouse emulation is performed.

ROM The BIOS image file. This is normally the file `/usr/vpix/defaults/rom`.

EGAROM

The EGA BIOS image file. This is normally the file `/usr/vpix/defaults/romega`. On systems without an EGA, this keyword is ignored.

VGAROM

The VGA BIOS image file. This is normally the file `/usr/vpix/defaults/romvga`. On systems without a VGA, this keyword is ignored.

CMOS The CMOS emulation file. This is normally the file `/usr/vpix/defaults/cmos`.

BOOTIMAGE

The fast-load virtual PC image used when `vpix` is invoked with the `-r` option and no arguments. It is normally set to `$HOME/vpix/vpix.img`.

Any other keywords found in the configuration file are looked up in the file `/usr/vpix/etc/vpixdevs` for a description of their characteristics. See "VP/ix Environment Maintenance Procedures" for further information.

Implicit Invocation

`vpix` can be implicitly invoked when the name of a DOS command is typed at the UNIX System shell prompt. This facility permits users to execute DOS programs from within either DOS or the UNIX System. See `cs(1)`, *Addendum* and `sh(1)`, *Addendum*.

FILES

<code>/usr/vpix/defaults/vpix.cnf</code>	Default device configuration file
<code>\$HOME/vpix/vpix.cnf</code>	Device configuration file
<code>/usr/vpix/defaults/cmos</code>	CMOS emulation file
<code>/usr/vpix/defaults/C:</code>	Common bootable pseudo-disk
<code>\$HOME/vpix/C:</code>	Private bootable pseudo-disk
<code>\$HOME/vpix/vpix.img</code>	Fast-load virtual PC image
<code>/usr/vpix/defaults/rom</code>	BIOS image file
<code>/usr/vpix/defaults/romega</code>	EGA BIOS image file, if running on EGA display
<code>/usr/vpix/defaults/romvga</code>	VGA BIOS image file, if running on VGA display
<code>/usr/vpix/etc/vpixdevs</code>	Descriptions of nonstandard devices

/usr/vpix/term/sys.default

Terminal-independent terminal description file, if running on serial terminal

/usr/vpix/term/*

Terminal-dependent terminal description files, if running on serial terminal

SEE ALSO

csh(1) Addendum, sh(1) Addendum, vt(7).
“VP/ix and MS-DOS Primer,” “VP/ix Environment Maintenance Procedures.”



INTERACTIVE



A Kodak Company

DOC0115-2Z