

Administering SCO[®] Merge[™]



Administering SCO[®] Merge[™]

© 1983-1993 The Santa Cruz Operation, Inc.
© 1980-1993 Microsoft Corporation.
© 1989-1993 UNIX System Laboratories, Inc.
All Rights Reserved.

No part of this publication may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of the copyright owner, The Santa Cruz Operation, Inc., 400 Encinal, Santa Cruz, California, 95060, U.S.A. Copyright infringement is a serious matter under the United States and foreign Copyright Laws.

The copyrighted software that accompanies this manual is licensed to the End User only for use in strict accordance with the End User License Agreement, which should be read carefully before commencing use of the software. Information in this document is subject to change without notice and does not represent a commitment on the part of The Santa Cruz Operation, Inc.

SCO OPEN DESKTOP Software is commercial computer software and, together with any related documentation, is subject to the restrictions on U.S. Government use as set forth below.

If this procurement is for a DOD agency, the following DFAR Restricted Rights Legend applies:

RESTRICTED RIGHTS LEGEND: Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of rights in Technical Data and Computer Software Clause at DFARS 252.227-7013. Contractor/Manufacturer is The Santa Cruz Operation, Inc., 400 Encinal Street, Santa Cruz, CA 95060.

If this procurement is for a civilian government agency, the following FAR Restricted Rights Legend applies:

RESTRICTED RIGHTS LEGEND: This computer software is submitted with restricted rights under Government Contract No. _____ (and Subcontract No. _____, if appropriate). It may not be used, reproduced, or disclosed by the Government except as provided in Paragraph (g)(3)(i) of FAR Clause 52.227-14 or as otherwise expressly stated in the contract. Contractor/Manufacturer is The Santa Cruz Operation, Inc., 400 Encinal Street, Santa Cruz, CA 95060.

SCO, Open Desktop, The Santa Cruz Operation, the Open Desktop logo, and the SCO logo are registered trademarks of The Santa Cruz Operation, Inc. in the USA and other countries.

All other brand and product names are or may be trademarks of, and are used to identify products or services of, their respective owners.

Date: 2 July 1993
Document Version 3.1.0A



Chapter 1

Introduction

1

Who should use this guide	1
Conventions used in this guide	2
Organization of this guide	3
Other SCO Merge documentation	3
Installing SCO Merge	4
Release Notes	4

Chapter 2

Administering SCO Merge

5

Using the dosadmin program	6
Adding and deleting user accounts	6
Administering the system console	7
Setting up the console	7
SCO Merge and display modes	7
Changing console display adapter cards	9
Using extended video modes	9
Using the UNIX MultiScreen facility with DOS	9
Redefining the SCO Merge switch-screen key sequence	10
Administering terminals	11
Setting up serial terminals	11
Setting up PC scancode terminals	11
Administering com ports	12
Administering DOS printers	13
Configuring the default DOS printer	13
Selecting a printer interface program	13
Changing the default DOS printer	14
Redirecting DOS printer output	14
Administering printers directly attached to DOS	15
Changing the printer timeout	17
Using the DOS mode command	17
Backing up filesystems	18
Administering disk and diskette drives	18
Sharing diskette drives	18
Diskette drives on very fast machines	19

Administering physical DOS partitions	20
Creating and formatting physical DOS partitions	21
Default protection of DOS partitions	21
Changing the protection of DOS partitions	22
Administering virtual DOS partitions and virtual floppy disks	23
Using dosadmin to create a virtual DOS partition	24
Using virtual partitions	26
Administering virtual DOS partitions	26
Using dosadmin to create a virtual DOS floppy	27
Installing plug-in cards in your computer	29
Using directly attached DOS devices	29
Setting up /etc/dosdev	32
Mapping device names	33
Limitations of direct attachment	34
Making new DOS images	34
Hardware changes that require new DOS images	35
Standard ROMs and on-card ROMs	35
Changing special config.sys commands	36
Adding device drivers to DOS images	36
Using mkimg to make new DOS images	37
Using dosadmin to make new DOS images	37
System files affected by system administration	39

Chapter 3

Installing DOS applications

41

Tips on installing DOS applications to run under SCO Merge	42
Installing personal DOS applications	42
Installing public DOS applications	44
Defining the search path	45
Installing applications with advanced processor requirements	45
Configuring DOS applications for use from the UNIX shell	47
Using dosadmin to configure DOS applications	47
Manually configuring DOS applications	49
Installing copy-protected DOS applications	50
Using a key disk	51
Using special installation procedures	51
Installing DOS applications on a DOS partition	51
Installing DOS applications under SCO systems	51
Installing applications under "raw" DOS	52

Removing DOS applications	54
Removing the application from the fixed disk	54
Deleting UNIX links	54
Redefining the search path	54

Chapter 4

Tuning applications with the merge command **55**

fastclk option	56
pollsleep option	56
merge network control options	57
drive option	57
handle option	58
netbios option	58
5f02 option	59
lpt_binary option	59
display option	60
return option	60
Using batch files for merge commands	61

Chapter 5

Tailoring the operation of DOS **63**

DOS defaults	64
Using DOS options on the dos command line	64
Installing DOS options to change defaults	65
Installing DOS options to override system defaults	66
Using dosadmin to install and remove DOS options	67
Characteristics you can tailor using dosadmin	67
Startup memory	67
DOS startup file	67
DOS device file	68
Using dosadmin to tailor DOS	68
Selecting the configuration menu	69
Selecting the DOS command or application	70
Filling in the configuration menu	71
Using dosopt to install and remove DOS options	73
Creating a home directory options file	75
Installing DOS options	75

Displaying installed options	76
Removing installed options	77
Restoring DOS options files	77
Allocating memory for DOS	78
Running RAM-hungry applications in base memory	78
Preparing minimal startup files	79
Using minimal startup files	79
Allocating memory for Windows	80
Memory for Windows 3.0	80
Memory for Windows 3.1	81

Chapter 6

DOS options

83

Attach devices to DOS	86
Default devices	86
The keyboard	86
The fixed disk	86
Diskette drives	86
The mouse	87
Physical DOS partitions	87
Display adapters	87
Other standard DOS devices	88
com ports	88
Expanded Memory (EMS)	89
Printer ports	89
The game port	89
Attaching DOS drives	90
Specifying immediate, exclusive, or read-only access	91
Specifying custom devices	91
Canceling device specifications	92
Using -a to prevent DOS from accessing devices	92
Using -a to remove defaults from options files	93
Display-oriented and stream-oriented DOS programs	93
Pass command directly to DOS command interpreter	95
Set initial current drive	96
Interpret configuration file	96
Help text	97
Alternative DOS load image	98
DOS memory size	99

Run autoexec.bat	100
Change DOS printer timeout	101
Translate DOS switch characters and path separators	102
Dosopt "verbose" mode	103
Set X Client startup	104
dosopt null option	104
Remove assigned options	104

Chapter 7

Using SCO Merge NLS features **107**

Features	107
Prerequisites	108
Theory of National Language Support	109
DOS code pages and UNIX operating system code sets	110
Differences between DOS code pages and UNIX code sets	110
How MS-DOS handles NLS	111
Varieties of DOS code pages	111
Code page switching	112
Tailoring DOS for different languages	112
How the UNIX operating system handles NLS	112
How SCO Merge handles NLS	113
Display considerations	113
Setting up the SCO Merge NLS environment	114
Setting UNIX operating system environment variables	116
DOS National Language Support codes	116
NLS commands supported by SCO Merge	117
SCO Merge exceptions	118
Sample NLS setup procedures for VGA displays	118
Setting up the startup files	118
Using DOS images	119
Resetting the DOS image	120
Sample NLS setup procedures for ISO terminals	121
Naming DOS and UNIX operating system files	122
Printing in an NLS environment	123
Converting text files	124
Using dos2unix and unix2dos	124
Using charconv	124
charconv options	125
Using the CONVOPTS environment variable	126

Changing colors in DOS windows	128
How DOS and X Windows handle colors	128
Installing the DOS colors	129
X System resources for SCO Merge	129
Running SCO Merge on remote servers	131
Setting the XMERGE environment variable	131
SCO Merge fonts	132
Using expanded memory (EMS)	132
Using multiport serial cards	132
SCO Merge/DOS compatibility	133
DOS interrupts	133
DOS clock interrupt rate	134
Applications that continuously poll the keyboard	134
Configuring applications that buffer keystrokes	134
Local and remote drives and file handles	135
Limits on number of open files	135
Protected mode restrictions	135
Passing UNIX environment variables to DOS	136
Setting your DOS search path with DOSPATH	138
Setting other DOS environment variables with DOSENV	139
Running DOS or a DOS application as the default shell	140
Running DOS as the default shell	141
Running a DOS application as the default shell	141
Scheduling DOS programs	141

Characteristics of DOS on the console	143
Using EGA cards	144
Prerequisites to using EGA cards	144
Using EGA "fast" and "reliable" modes	145
DOS processes that suspend when their screens are not visible	146
Using VGA cards	146
Using PC scancode terminals	146

Using serial terminals	147
Serial terminal keyboard mapping	147
Console and terminal control codes	151

Appendix B

Terminfo guide

153

Basic concepts	154
terminfo	154
terminfo files	154
Standard and PC scancode terminals	155
Escape sequences	156
The SCO Merge lead-in sequence	156
Escape sequences used by SCO Merge	156
Output on standard terminals	157
Special terminfo descriptors	160
pctrm	160
dispc	161
How PC characters are displayed on terminals	162
How PC characters are displayed on ISO terminals	162
Combined ISO and PC character set terminals	163
Input on standard terminals	164
Default “key emulation” escape sequences	165
Default “special action” escape sequences	166
Using terminfo descriptors for key emulation	166
pctrm	167
Changing the lead-in character	168
About PC scancode terminals	168
Switching commands	169
Alternate flow control	169
Serial line parameters	169
pctrm	170
Sequence of use	171
SCO scancode terminal support	172
Output on PC scancode terminals	172
Parameterized strings	173
dispc as a parameterized string	173
Code pages	174

Input on PC scancode terminals	175
Changing the lead-in sequence	175
PC-character-to-ASCII conversions	177
terminfo escape sequences	178

Chapter 1

Introduction

This guide covers administrative and advanced topics of SCO® Merge™. Administering SCO Merge is no different in most respects from administering separate, conventional DOS and UNIX systems. The administrator's responsibilities include installing and maintaining system hardware and software, regularly backing up system data, assisting users, and informing them of changes to the system.

This guide supplements the system administration instructions in your DOS and UNIX documentation. You should be familiar with that documentation because this guide is not a comprehensive description of the system administrator's responsibilities. In general, you administer SCO Merge by using UNIX procedures to accomplish UNIX tasks and DOS procedures to accomplish DOS tasks.

Who should use this guide

This guide is for the SCO Merge system administrator, the person responsible for maintaining the day-to-day operation of the system. This guide covers the administrative procedures that are necessary to manage the combined DOS and UNIX environment of the SCO system. It supplements the documentation on your computer hardware, DOS, and the UNIX System.

This guide is also intended for advanced SCO Merge users who would like to know more about customizing the DOS environment and maximizing its efficiency.

Conventions used in this guide

This guide uses the following notational conventions:

- bold** Commands, command options, and daemons appear in bold. UNIX commands are case sensitive and will always be presented exactly as they must be entered. DOS commands are not case sensitive. Unless the command in question appears within a file (for example within *config.sys*), DOS commands will always be presented in lower case bold.
- BOLD CAPS** Parameters and variables and DOS commands contained in files appear in uppercase bold. For example:
... SCO SCO Merge does *not* interpret FCBS commands in any of your system's *config.sys* ...
- SMALL CAPITALS** Acronyms are shown in SMALL CAPITALS. For example:
... the name UUCP is an acronym for UNIX to UNIX Copy ...
- italics* Files and directories appear in italics. Italics are also used for emphasis. DOS filenames will always appear in lower case even though DOS filenames are not case sensitive. Additionally, the DOS Drive name (for example, A:) will be treated as a filename when used as an indicator of a file location and it will always be in uppercase.
- bold italics*** Variables that you supply are in bold italics. For example, in the command ***path:pathname***, you replace the variable "pathname" with an actual pathname when you type the command.
- <>** Special keys that you press are in angle brackets, < >. For example, <Ctrl>x means to hold down the Control key and press the x key simultaneously, then release them.
- Courier** System responses, programming, and screen examples appear in Courier style font.

Organization of this guide

This guide has seven additional chapters and two appendices:

Chapter 2: “Administering SCO Merge,” tells you how to manage user accounts, set up and administer computer hardware used by DOS, and configure your computer’s resources to meet the combined needs of DOS and UNIX System users.

Chapter 3: “Installing DOS applications,” provides hints for installing DOS applications for personal or public use, installing copy-protected applications, setting up DOS applications for use from the UNIX shell, and removing DOS applications.

Chapter 4: “Tuning Applications with the **merge** Command,” tells you how to tune the operation of the DOS process for certain applications.

Chapter 5: “Tailoring the Operation of DOS,” and Chapter 6: “DOS Options,” tell you how to modify DOS characteristics and configure DOS to suit your needs.

Chapter 7: “Using SCO Merge in a Multilanguage Environment,” describes SCO Merge features that support multiple languages in an environment shared by DOS and the UNIX operating system..

Chapter 8: “Other Advanced Topics,” covers a wide variety of techniques for customizing DOS and improving SCO Merge efficiency.

Appendix A: “Using the System Console and Terminals,” explains SCO Merge’s operation with different console display adapters and terminals.

Appendix B: “terminfo Guide,” explains the function of the **terminfo** file and tells you how to use it to run SCO Merge on the terminals attached to your UNIX system.

Other SCO Merge documentation

Other guides that describe SCO Merge operation and administration include:

- *Using SCO Merge*
- *SCO Merge Release and Installation Notes*

Installing SCO Merge

This guide assumes that you have installed SCO Merge according to the instructions in the *Release and Installation Notes*.

Release Notes

Be sure to read the *SCO Merge Release and Installation Notes* for up-to-date information on supported hardware and software as well as information on product changes since this guide was printed.

Chapter 2

Administering SCO Merge

This chapter covers the topics that are essential for using your computer as a combined DOS and UNIX machine. The topics include:

- using the **dosadmin** program,
- adding and deleting user accounts,
- configuring applications that buffer keystrokes,
- administering the system console,
- administering terminals,
- administering com ports,
- administering DOS printers,
- backing up filesystems,
- administering disk and diskette drives,
- administering physical DOS partitions,
- administering virtual DOS partitions and floppy disks,
- installing plug-in cards,
- making new DOS images, and
- system files affected by system administration.

To administer SCO Merge effectively, you should be familiar with the contents of the DOS sections in *Using SCO Merge*, in addition to this chapter.

Most of the descriptions in this chapter assume that you are logged in as root or you are the super user.

Using the dosadmin program

The **dosadmin** menu system provides an easy way to change the values of three important DOS characteristics: memory, DOS startup files, and DOS device files. Throughout this guide you will use the **dosadmin** menu to perform various system administration tasks.

You can run **dosadmin** from either the DOS environment or the UNIX shell. When you start **dosadmin** while using the DOS environment, your prompt before starting **dosadmin** and after exiting **dosadmin** is a DOS prompt, such as C>. When you start **dosadmin** from the UNIX shell, your prompt before and after using **dosadmin** is \$ (or # if you are logged in as root). The actual **dosadmin** procedures and displays are identical, whether you start **dosadmin** from the UNIX shell or in the DOS environment.

Adding and deleting user accounts

SCO Merge requires no special procedures for adding or deleting user accounts. Any user with a valid UNIX account can use SCO Merge. You are not required to be logged in as root or have any special permissions, for example, to run DOS or use DOS and UNIX commands.

In general, follow the instructions in the Administrator's Guide for adding, deleting, and administering user accounts.

Although no special configuration files or setup procedures are required to use DOS, SCO Merge users may want to customize the way DOS runs. Users who follow the instructions in the DOS sections can alter the behavior of DOS in many ways without help from the system administrator. If users at your site are unfamiliar with the DOS or UNIX systems, they may require your assistance. *Using SCO Merge* contains hints for system administrators who want to modify SCO Merge defaults or help users configure their own individual environments.

Administering the system console

SCO Merge works with any system console that consists of a monochrome or color monitor with a PC-compatible keyboard, connected to a monochrome, Hercules®, CGA, or VGA display adapter. Appendix A describes how to use the console. This section tells you how to connect and configure the console. The topics covered here are:

- Setting Up the Console
- Changing Console Display Adapter Cards
- Using Extended Video Modes
- Using the UNIX MultiScreen™ Facility with DOS
- Redefining the SCO Merge Switch-Screen Key Sequence

Setting up the console

No special procedures are required to set up the system console for use with SCO Merge. The only requirement for using SCO Merge is that you must use one of the display adapters that SCO Merge recognizes. These include monochrome, Hercules, CGA, and VGA adapters.

SCO Merge and display modes

When you use SCO Merge, you can run DOS applications on serial terminals or on a console that uses a monochrome, Hercules, CGA, or VGA display adapter and a compatible display. In addition, by using different DOS images, SCO Merge can simulate the characteristics of display adapters less powerful than your actual physical display adapter. (Refer to “Making new DOS images” later in this chapter for more information on DOS images.)

This flexibility can lead to unexpected problems if you are not careful how you install, configure, and run graphics applications. For example, an application may fail to run if you have configured it to use a different type of display than you really use. If your system uses more than one type of display, you may find that some applications run correctly on some displays but not on others.

Some applications automatically detect the type of display adapter being used and adjust themselves to display graphics data correctly. Other applications do not work properly unless you identify your display adapter when you install them. When you install these applications, be sure you follow the application manufacturer’s instructions and configure the applications to work with the display adapter you intend to use.

If you run DOS processes from a serial terminal, identify your display as a monochrome display adapter (MDA).

If you use a graphics display adapter, you should normally configure your DOS applications to use the type of adapter in your computer, provided the application is compatible with that type of adapter. For example, if you use a VGA card, configure your DOS applications to use a VGA card.

In rare cases, you may run into difficulty because you are trying to use an incompatible combination of physical display adapter, DOS image, and application. By default, SCO Merge uses DOS images that correspond to your physical display — VGA images for VGA displays, CGA images for CGA displays, and so on. Most DOS applications that are configured for less powerful display adapters run correctly, automatically, when you run them on more powerful adapters. For example, an application configured for CGA displays may run correctly on a VGA display. However, some applications fail when run on a more powerful display than they are configured for. If you run into this restriction, you can work around it in either of two ways:

- Reconfigure the application so it expects to be run on the display adapter you are using. For example, if your application expects to be run on a CGA card and you use a VGA card, reconfigure the application to expect a VGA card.
- Use a DOS image that corresponds to the display adapter the application expects to use.

For example, if you have a VGA card but your application expects to use a CGA card, you can start a DOS environment with the command:

```
dos +acga
```

Then start your application. If you want to start the application directly from the UNIX shell, you can also use a command such as:

```
dos +acga appl
```

where **appl** is the command that starts your application. These commands cause SCO Merge to use a CGA image instead of the default image, and your application views your computer hardware as though you actually have a CGA card.

Note that this technique works only if your physical display is capable of using the DOS image you request. The following table shows usable combinations of physical display adapters and DOS images. Fields with a dash (-) indicate unusable combinations.

Option	Capability By Physical Display Type			
	Monochrome	Hercules	CGA	VGA
None	mono	herc	CGA	VGA
+amono	mono	mono	mono	mono
+aherc	-	herc	-	-
+acga	-	-	CGA	CGA
+avga	-	-	-	VGA

Refer to Chapter 6 for more information on the **+a** option. Refer to Appendix A for more information on the characteristics of graphics displays.

Changing console display adapter cards

If you change your console display adapter card after you install SCO Merge, you may need to make a new DOS image. See "Making New DOS Images," later in this chapter, for further information.

Using extended video modes

Whether or not you can use extended video modes on VGA cards depends on how your DOS images were created. If you have a VGA display and want to run high-resolution graphics applications, you need to create a VGA DOS image. You can run DOS applications that require monochrome or CGA displays using your VGA card even if you do not make a VGA image.

To make a VGA image, your computer must have a 100 percent IBM-compatible VGA card. If your card is not fully IBM-compatible, the procedure for making a VGA image may fail. When the procedure fails, symptoms range from an error message like the one shown below to a system lockup.

```
VGA Image was not made.
(Incompatible VGA card?)
(1)
```

For information on how to create a new DOS image, refer to "Making New DOS Images," later in this chapter.

Using the UNIX MultiScreen facility with DOS

SCO Merge is compatible with the standard UNIX MultiScreen facility. If your console can display multiple, independent screens, you can run DOS commands and applications on a DOS environment in any screen.

When you run multiple DOS processes in separate screens, the default key sequence for switching between screens is **<Alt><Ctrl><Fn>**, where **<Fn>** is a function key (such as **<F1>** or **<F2>**). **<Alt><Ctrl><F2>** switches to screen 2, **<Alt><Ctrl><F3>** switches to screen 3, and so on. The UNIX system also uses the **<Alt><Ctrl><Fn>**

convention for switching between screens, so you can switch between any active DOS and UNIX screens using `<Alt><Ctrl><Fn>`.

Redefining the SCO Merge switch-screen key sequence

You can redefine the switch-screen key sequence that applies to DOS screens if `<Alt><Ctrl><Fn>` does not suit your needs. For example, you may have a DOS application that does not work properly with the default switch-screen sequence because it assigns `<Alt><Ctrl><Fn>` a special meaning. You can redefine the switch-screen key sequence to use a function key together with any combination of the `<Ctrl>`, `<Alt>`, and `<Shift>` keys, or to use only a function key. To redefine the switch-screen key sequence, use the **switchkey** command with the syntax:

switchkey [-cas]

In this syntax, **c** stands for the `<Ctrl>` key, **a** stands for the `<Alt>` key, and **s** stands for the `<Shift>` key. You can use these options in any combination and specify them in any order. To switch screens any time after you use the **switchkey** command, use the keys you specified with **switchkey** in addition to a function key. For example, to specify that you want to require `<Ctrl>` and `<Shift>` along with a function key, type:

switchkey -cs

Thereafter, to switch screens, press `<Ctrl><Space><Fn>`

To specify that you want to use only function keys, without `<Ctrl>`, `<Alt>`, or `<Shift>`, use **switchkey** with only a hyphen as an argument:

switchkey -

Thereafter you can use `<F1>`, `<F2>`, `<F3>`, and so on, without additional keys, to switch from your current DOS screen to a different one. The **switchkey** command with no arguments displays the current switch-screen key sequence.

The switch-screen key sequence you define with **switchkey** applies only when you are viewing a DOS screen. When you are viewing a UNIX screen, the existing UNIX switch-screen sequence is effective.

SCO Merge and the X Window System both use the same default switch-screen key sequence and are affected the same way when you use **switchkey**.

Any switch-screen key sequence that you define with **switchkey** applies to any DOS or X Window sessions run on the console where you ran **switchkey**. DOS and X Window sessions that are currently running when you redefine the switch-screen key sequence are also affected. Your specified key sequence remains effective until you issue another **switchkey** command or reboot your computer. To make sure your preferred key sequence remains effective even after a reboot, include the **switchkey** command in `/etc/profile` or your home directory `.profile`.

Administering terminals

SCO Merge works with a variety of terminals. Appendix A of this guide discusses using terminals. This section tells how to set up serial and PC scan-code terminals. See the *Release Notes* for a list of supported terminals. Refer to Appendix B in this manual for more information on the UNIX *terminfo* file and how SCO Merge uses it to run DOS on terminals attached to the system.

Setting up serial terminals

No special procedures are required to set up serial terminals for use with SCO Merge. Simply follow the terminal and UNIX system manufacturers' instructions for setting up the terminal for standard UNIX use. The procedures require physically attaching the terminal to a serial port in your system unit and making sure the UNIX **TERM** environment variable is set correctly for anyone who uses the terminal.

You may also need to modify entries in your *terminfo* file. See Appendix B for a full description.

Setting up PC scancode terminals

SCO Merge supports PC scancode terminals that can be switched between character and scancode modes under software control. These terminals are referred to as "switchable scancode terminals." Wyse® Technology's WY-60 and Kimtron® Corporation's Model KT-70/PC terminals are examples of switchable scancode terminals. SCO Merge automatically switches scancode terminals into character mode when they communicate with the UNIX system and into scancode mode when they communicate with DOS.

To set up a scancode terminal, follow these procedures:

1. Configure your scancode terminal to use eight data bits and one stop bit, no parity, and 25-line display mode.
2. Configure the terminal to use normal character (ASCII) mode by default. (SCO Merge switches these terminals to scancode mode automatically when necessary.) Follow the steps that apply to your terminal:
 - Wyse WY-60 terminals: Set the terminal's "Personality" to WY-60.
 - Kimtron KT-70/PC terminals: Set the "Emulation" parameter to "IX" mode.
3. Attach your terminal to any available serial line on your computer. Make sure the serial line is configured for eight data bits, one stop bit, and no parity.

4. Define the **TERM** environment variable appropriately for the terminal you are using:
 - Use the name **wy60ix** for Wyse Technology's WY-60 terminal.
 - Use the name **kt70pcix** for Kimtron Corporation's Model KT-70/PC.

Refer to the manual for your terminal for more information on these procedures.

Administering com ports

On the SCO system, both the UNIX environment and DOS can use com (serial) ports, but only one process (either UNIX or DOS) can access a particular com port at one time. If you plan to use a com port for DOS work, follow these procedures:

1. Make sure no **getty** is running on the physical device (for example, */dev/tty1a* or */dev/tty2a*) that you want to make available to DOS as a com port. To do this, log in as root or become the super user and run the **disable** command. For example, to disable the **getty** process on */dev/tty1a*, type:

disable /dev/tty1a

2. Use the **chmod** command to set permissions of the device to be readable and writable. If you want to use */dev/tty1a*, you would type:

chmod 666 /dev/tty1a

The com port is now ready for use with DOS. You can attach it to any DOS session with the **+a** option:

dos +acom1

To return the com port for use by the UNIX system, type:

enable /dev/tty1a

For more information on using com ports with DOS, refer to Chapter 6 in this guide and to the DOS sections in *Using SCO Merge*.¹

1. "Smart" cards are not supported by SCO Merge unless connected through a special driver which is provided by the manufacturer. See "com Ports" in Chapter 6.

Administering DOS printers

The DOS section of *Using SCO Merge* describes, from the user's point of view, how to print using DOS commands or applications. This section explains how to set up and administer DOS printers. It covers the following topics:

- configuring the default DOS printer,
- changing the default DOS printer,
- redirecting DOS printer output, and
- administering printers directly attached to DOS.

Configuring the default DOS printer

By default, SCO Merge sends all DOS printing via the UNIX spooler to a printer named **doslp**. The SCO Merge installation routine automatically attempts to configure your default UNIX printer, if one exists, so it can also be used as the default DOS printer, **doslp**. To determine whether **doslp** has been configured, type the following **lpstat** command:

```
lpstat -p doslp
```

This command tells you whether or not **doslp** is properly configured. If it is not, you need to configure **doslp** yourself.

To configure **doslp**, use the UNIX **sysadmsh** command. Refer to the Administrator's Guide for complete instructions on configuring printers. You need to know the UNIX device name of the printer you want to use as **doslp** (for example **/dev/lp0**). Use standard UNIX procedures to perform the following operations:

1. Assign the printer name **doslp** to your preferred printing device.
2. Choose the appropriate printer interface program as described below. (The interface program is also known as a printer model.)

Selecting a printer interface program

SCO Merge supplies a printer interface program named **dosmodel**. It is the same as the **standard** model, except that it does not print banner pages at the beginning of each print job, and it does not output a form feed at the end of each print job.

If SCO Merge automatically configured **doslp** during the installation procedure, it used **dosmodel**. Any printer that recognizes the **standard** model can also use **dosmodel** for printing DOS output or any other output that does not require banner pages or form feeds.

You can choose the **standard** model for **doslp** if you prefer to have banner pages and automatic form feeds.

If your printer does not recognize the **standard** model, you should not use either the **standard** model or **dosmodel**. Instead, choose a model that is appropriate for your printer.

Changing the default DOS printer

The default DOS printer is always named **doslp**. To change the physical printer that **doslp** refers to, follow the standard UNIX procedures for configuring printers. Assign the name **doslp** to your preferred printing device and choose an appropriate interface program.

Redirecting DOS printer output

You can use printers other than **doslp** for DOS printing. You can select any convenient printer name and any appropriate interface program when you configure a printer.

To use any available UNIX printer for DOS printing, use the SCO Merge **printer** command to correlate a print stream with a particular UNIX printer and print command. The syntax for selecting a print stream, a printer, and a UNIX print command is:

printer [*print_stream*] *unixprint_command*

where *print_stream* is *lpt1*, *lpt2*, or *lpt3* and *print_command* is a UNIX command that processes the specified print stream. If you do not specify a print stream, **printer** assumes *lpt1* by default. Use the **printer** command in the DOS environment. Assume, for example, that you want to direct DOS printer output sent to *lpt2* to a UNIX printer named "laser." Follow these steps:

1. Start a DOS environment if you have not already started one.
2. Use the following **printer** command to direct print stream *lpt2* to the printer named "laser":

printer lpt2 unix "lp -dlaser"

In this command, the **-d** ("destination") option to the **lp** command identifies the printer named "laser."

3. To send DOS printer output to the printer, name the DOS print stream in your DOS print command. For example:

```
copy letter.txt lpt2
```

You can direct printer output from DOS applications to any UNIX printer using the same procedures.

If you want a **printer** command to be effective every time you use the DOS environment, you can include it in an *autoexec.bat* file.

Note that the *print_command* that you specify when you use the **printer** command is typically **lp -dprinter**, where *printer* is the name of a UNIX printer. However, *print_command* can be any UNIX command that you choose to use to process a print stream.

2

Administering printers directly attached to DOS

Under some circumstances, SCO system users may not want to use the UNIX spooling system when they use DOS printing. In these cases, users can attach a printer directly to a DOS process. The printer is then under the direct control of DOS, and not available for UNIX printing.

If a user wants to attach a printer directly that is currently configured for UNIX printing, you need to disable UNIX printing on that printer. Follow these procedures:

1. Log in as root or become the super user.
2. Disable UNIX printing by using the **disable** command. For example, if you want to disable UNIX printing on the printer named **doslp**, issue the command:

```
disable doslp
```

Note that while UNIX printing is disabled, users can continue to spool print jobs to that printer. However, the jobs are not printed until the printer is re-enabled for UNIX use.

3. To directly attach a printer to a DOS process, start DOS using the **+a** option to specify the physical printing device you want to use. SCO Merge uses the device names **lp0**, **lp1**, and **lp2** to identify the first, second, and third parallel printer ports. Use the name that corresponds to the port your printer is attached to. For example, if your printer is attached to */dev/lp0*, you can start the DOS environment with the command:

```
dos +alp0
```

Consult the manuals for your computer and the SCO Merge */etc/dosdev* file if you need assistance in identifying your printer port.

4. The printer can now be used with any of the standard DOS printing techniques, including the **print** command, copying a file to the printing device, and printing from an application.

The DOS name of the printing device is determined by DOS, which recognizes a maximum of three printing devices. These are always named *lpt1*, *lpt2*, and *lpt3*. The UNIX system has fewer restrictions — in theory, any number of printer ports can exist, and they can be named in any fashion. Nevertheless, SCO Merge recognizes only three parallel printer ports and names them *lp0*, *lp1*, and *lp2*.

Notice that the DOS name does not necessarily correspond to the name of the physical port attached in the previous step.

DOS checks each of the directly attached physical printer ports, in order, to identify the first port connected to a printer. No matter which physical port the printer is attached to, DOS names the first printer it finds "*lpt1*," the second printer it finds "*lpt2*," and so on. If you have only one printer and it is attached to the physical port named *lp2*, DOS views this printer as *lpt1*.

For example, if you have one directly attached printer, you can print a file from your DOS prompt with the command:

```
copy filename lpt1
```

When your DOS session ends, the printer can be enabled again for UNIX printing.

NOTE When you directly attach a printer to a DOS process as described in this section, you can use all DOS **print** command options when you use that printer.

5. When the user finishes the printing that required directly attaching the printer to DOS, you can use the **enable** command to re-enable UNIX printing. For example, to re-enable the printer **doslp**, issue the command:

```
enable doslp
```

NOTE A single physical printer may have more than one printer name associated with it. If the UNIX printer you are disabling has more than one UNIX printer name, then, in step 2, you should issue the **disable printer_name** command for each printer name correlated with that printer. Similarly, in step 3 of this procedure, you may need to issue multiple **enable** commands.

Changing the printer timeout

When you use a DOS application that prints, SCO Merge by default sends DOS printer output to the UNIX spooler when either of the following conditions is true:

- You exit the application and return to your DOS or UNIX prompt.
- More than 15 seconds have elapsed since the application has sent a character to be printed.

This default is convenient for most applications. However, under certain conditions you may want to change the printer timeout from 15 seconds to a different value. For example, if your application pauses for more than 15 seconds while printing, the printer output sent before and after each pause is spooled as a separate UNIX print job.

To change the printer timeout in the DOS environment, use the SCO Merge **printer** command with the syntax:

```
printer [lptn] unix /t[timeout]
```

In this command, *lptn* identifies the print stream for which you wish to change the timeout. *lpt1* is assumed if you do not specify a print stream. *timeout* is a value, in seconds, between 5 and 3600. For example, if you want to increase the timeout for print stream *lpt2* to one minute, type the command:

```
printer lpt2 unix /t60
```

Using the **/t** option without specifying a value returns the printer timeout for the specified print stream to the default of 15 seconds.

The timeout value "0" has a special meaning: it delays the spooling of DOS printer output until you exit the application.

You can use the **±s** option to the **dos** command instead of the **printer** command to control the printer timeout. Refer to Chapter 6 for information on the **±s** option.

Using the DOS mode command

If you use the **mode** command to configure a DOS printer, the command has no effect unless the printer is directly attached to the DOS process.

Backing up filesystems

To guard against permanently losing important data, you should regularly back up all data on your fixed disk. SCO Merge requires no special procedures for system backups. Simply use UNIX backup procedures for the filesystem shared by DOS and the UNIX operating system and DOS backup procedures for the physical DOS partition(s).

Consult your DOS and UNIX documentation for specific procedures recommended for your system.

NOTE If they are stored on a physical or virtual DOS partition, some copy-protected DOS applications cannot be backed up and restored using the **backup** and **restore** commands. For more information, consult your DOS user's manual and the instructions for your applications.

Administering disk and diskette drives

Your SCO Merge system may have one or more diskette drives that can be assigned to either the DOS or the UNIX environment. A diskette drive mounted as a UNIX device is assigned to the UNIX environment and is not accessible as a DOS diskette drive. If you insert a diskette containing a UNIX filesystem into a diskette drive and mount it as a UNIX device, any files on that diskette become part of the filesystem shared by DOS and the UNIX operating system.. You can then use DOS drive C: to access files on the mounted diskette in the same way you access files in the fixed disk filesystem shared by DOS and the UNIX operating system..

To be directly accessible to DOS as DOS drives A: or B:, diskette drives must not be mounted as UNIX devices or in use by UNIX programs (such as **cpio**). If you have more than one diskette drive on your system, you must not have any of them mounted as UNIX devices or otherwise in use by the UNIX system when you attempt to access a diskette drive as DOS drive A: or B:. SCO Merge prevents a DOS process from using all diskette drives as long as any other process is using any diskette drive.

Sharing diskette drives

When diskette drives are not mounted as UNIX devices or otherwise used by the UNIX system, DOS users can access them on a first-come, first-served basis. Whenever a diskette drive has not been used for five seconds or more, it is available to the first DOS process that accesses that drive.

Diskette drives on very fast machines

In a standard DOS machine, each ROM BIOS chip is specially designed to accommodate the hardware used by that manufacturer. To make DOS run under UNIX, SCO Merge emulates these ROM BIOS routines, but must run on whatever hardware is being used by the host system.

A few of the newest machines are so fast that SCO Merge may not read the diskettes accurately. On machines faster than 66 Mhz, users may see one of the following messages when trying to read that drive.

```
Not ready reading drive A:
Abort, Retry, Fail?
```

```
I/O error reading drive A
```

The system administrator may find evidence in system logs that some process is timing out repeatedly while waiting for the drive to respond. This behavior is usually consistent rather than intermittent.

SCO Merge offers the **patchflop** utility to correct this problem if it occurs. It "patches" the operation of the routines that read the floppy drive. The syntax for **patchflop** is as follows:

```
./patchflop [-p] [value]
```

The **-p** option prints the current setting used to adjust the ROM BIOS timing loops. The default value is 100.

The *value* argument is a single decimal number that sets the timing loops in the ROM BIOS. The number you need to use corresponds roughly to the MHz rating for your machine.

To use **patchflop**, assume root privileges and move to the directory where both the *bios.rom* file and the **patchflop** utility are stored.

```
cd /usr/lib/merge
```

The repair procedure is a trial and error process. The **patchflop** value does not take effect until a new DOS session is started.

First use the **-p** option to display the current setting:

```
./patchflop -p
```

For your first attempt, start with the MHz rating of your machine. For example:

```
./patchflop 50
```

Every time you use **patchflop** to modify the timing loops in the *BIOS ROM* file, start a new DOS session, and see if the floppy(s) can now be reliably accessed. If not, exit the DOS session, and run **patchflop** with a larger number. Increment the number by units of 5 or 10 each time. Do this until the diskette drive can be reliably accessed.

Administering physical DOS partitions

A physical DOS partition is a portion of the fixed disk formatted under DOS and reserved exclusively for DOS files. DOS partitions can be created in a variety of sizes; different versions of DOS support different maximum sizes for partitions. SCO Merge supports both the DOS 3.3 partitions, which are limited to 32 Mbytes, and the larger partitions that can be created under DOS 5.0.

DOS allows you to create two categories of partitions. Primary partitions are those from which you can boot the DOS operating system. Extended partitions are non-bootable and may be further subdivided into two or more logical drives. See your DOS documentation for further details. SCO Merge supports both primary and extended DOS partitions. Your fixed disk may have just one physical DOS partition (the primary DOS partition); it may have both a primary and an extended DOS partition; or it may have no physical DOS partitions.

If your computer has a primary DOS partition, the partition is automatically available as DOS drive E: whenever you run DOS. If your computer has an extended DOS partition and you want to use logical drives within it, you must attach each logical drive that you want to use to an available SCO Merge drive letter. Attaching logical drives is described in the DOS sections in *Using SCO Merge*. A physical DOS partition is useful on the SCO system for the following reasons:

- Some copy-protected DOS applications that cannot be installed in the file-system shared by DOS and the UNIX operating system can be successfully installed in a DOS partition.
- You can use files and applications contained in DOS partitions under “raw” DOS. That is, you can use them even if you shut down the UNIX system and boot your computer with DOS alone.

If you do not already have a physical DOS partition and you choose not to create one, you can still use all SCO Merge features except drive E:. You can also create and use a virtual DOS partition any time after you install SCO Merge without removing or reinstalling any files that are currently on your fixed disk. Virtual DOS partitions offer advantages similar to those of a physical DOS partition. Refer to “Administering Virtual DOS Partitions and Virtual Floppy Disks” later in this chapter for more information.

This section tells you how to create, format, and administer a physical DOS partition.

Creating and formatting physical DOS partitions

If you want to have physical DOS partitions on your computer, you should create them at the time you install the UNIX system, before you install SCO Merge. Boot standard "raw" DOS and use the DOS **fdisk** utility to create and format DOS partitions. Each DOS partition should have a minimum size of 2.5 Mbytes. Some DOS copy-protection schemes cannot install on a partition smaller than 2.5 Mbytes.

If you do not have a physical DOS partition on your computer and wish to create one, you will have to back up your fixed disk, create the DOS partition, and reinstall your UNIX system and all applications. Refer to the documentation packaged with your computer for further information.

You can also add a second hard disk to your computer and put one or more DOS partitions on it. SCO Merge automatically senses these new partitions the next time you reboot and makes them available. You still need to associate each of the new partitions with a drive letter. Refer to the section entitled "Using Physical DOS partitions" in your User's Guide for more information.

Default protection of DOS partitions

The UNIX filesystem protection mechanisms apply in only a limited way to DOS partitions. Because physical DOS partitions are not UNIX filesystems, access to individual DOS files in these partitions is not governed by UNIX system user or group ownership or by UNIX system read, write, or execute permissions,

However, a DOS partition itself is accessed as a UNIX file or device and can be protected like any other UNIX file or device.

Here are the considerations that affect access to and administration of DOS partitions:

- Although multiple users can read files on a given DOS partition at the same time, only one DOS process can write to the primary DOS partition or to a logical drive in the extended DOS partition at one time. In addition, when any DOS process writes to a DOS partition or logical drive, no other DOS processes can read from or write to that partition until the DOS session that was writing to the partition is terminated.
- DOS partitions contain DOS filesystems and are located on a portion of the fixed disk that is physically distinct from the filesystem shared by DOS and the UNIX operating system.

- DOS partitions are accessed as UNIX devices (sometimes called special files). Each of these devices appears to the UNIX system as a single UNIX file, but can contain any number of DOS files and directories. The device name for the primary DOS partition is */dev/dsk/ devicename*. The device file name for any DOS partition (if it exists) may be determined by looking in the file */etc/dosdev*. In that file, the first line under the heading "Real DOS device definition" is the name of the primary partition. The next line shows the name of the extended DOS partition. You can find an example of these lines in the DOS section of the User's Guide.
- When you install SCO Merge, the DOS partitions (that is, the UNIX system special device files associated with the partitions) are owned by root and are readable and writable by everyone. To check the ownership and permissions of the primary DOS partition, type:

```
ls -l /dev/dsk/devicename
```

To check the ownership and permissions of the extended DOS partition, type:

```
ls -l /dev/dsk/extended devicename
```

where *devicename* is the name of the special device file associated with your DOS partition(s), as determined by looking in */etc/dosdev*.

For both kinds of partitions, the system shows root as the owner and displays the following permissions:

```
brw-rw-rw-
```

The initial "b" identifies these files as block-type special files (as opposed to character-type). UNIX execute permission for the files is not necessary.

Although owned by root, DOS partitions by default are considered to be public resources that any user can access. All users have read permission so they can inspect the contents of the partitions or execute DOS programs stored there. All users have write permission so they can install DOS applications on the partitions. Write permission is granted to all users for an additional reason: even if users do not need to create files within DOS partitions, some DOS applications require write permission when they run and do not work if run from an unwritable filesystem.

Changing the protection of DOS partitions

If you need to restrict access to a DOS partition, you can use one or both of the following methods:

- Use the DOS **attrib** command to make specific files in the DOS partition read-only (that is, remove DOS write permission).
- Change the UNIX system permission mode of the UNIX system special file associated with the DOS partition to restrict access to the partition.

Using the DOS **attrib** command to make a DOS file read-only reduces the risk of that file being accidentally removed or corrupted. The protection offered by DOS **attrib** is limited, however, because anyone who can access a DOS file can also use **attrib** to make the file writable.

To make a DOS file read-only, use the **attrib** command to assign the read-only (R) attribute as follows:

attrib +r filename

When the read-only attribute is assigned, the file cannot be deleted or changed. To make the file writable again, type:

attrib -r filename

If you want to restrict access to a DOS partition in a more general way, you can use the UNIX command **chmod** to assign any desired permission mode. For example, to deny access to users outside the group the primary DOS partition belongs to, you could type:

chmod o-rw /dev/dsk/devicename

For the extended DOS partition, you would type:

chmod o-rw extended devicename

where *devicename* is the special device file associated with the your DOS partitions.

NOTE Changing the permissions for the extended partition changes permissions for all logical drives within that partition.

Administering virtual DOS partitions and virtual floppy disks

A physical DOS partition is a section of the fixed disk that is formatted under DOS and reserved for DOS files. SCO Merge also allows you to create virtual DOS partitions. A virtual DOS partition is actually a UNIX file that has been formatted as a DOS volume and is used to store DOS files. A virtual DOS partition has advantages similar to those of a physical DOS partition, as well as some special features.

Just as with a physical DOS partition, you can use a virtual partition to install copy-protected DOS applications that cannot be installed on the shared filesystem. Virtual partitions have the following additional advantages:

- You can create a virtual DOS partition easily at any time after you install SCO Merge. You can therefore avoid the time-consuming process of reconfiguring your fixed disk to create a physical DOS partition if your system does not already have one.

- Users can create and own virtual partitions. By using personal virtual partitions, they can avoid ownership and permission mode conflicts that may arise from sharing physical DOS partitions.

Because virtual DOS partitions are actually UNIX files and therefore part of the UNIX filesystem (unlike physical DOS partitions):

- Virtual partitions are backed up or restored using standard UNIX backup utilities, in contrast to physical DOS partitions which are backed up with DOS utilities.
- You cannot access a virtual partition when you have shut SCO Merge down and booted standard DOS.
- Virtual partitions take up space in the UNIX filesystem.
- Virtual partitions are limited to 7 Mbytes.

After you create a virtual partition, you can attach it as a DOS drive.

SCO Merge also allows you to create a UNIX file and format it under DOS for use as a virtual DOS floppy drive. You can use virtual floppies much as you would use physical DOS diskette drive A: or B:.

The following sections show how to create and administer virtual DOS partitions and how to create virtual floppies.

Using dosadmin to create a virtual DOS partition

To create a virtual DOS partition, use the following **dosadmin** procedure:

1. Start **dosadmin** by typing:

dosadmin

(You may start **dosadmin** from either the DOS or UNIX command line.)
The **dosadmin** "DOS Administration" main menu appears.

2. Press the Left or Right Arrow key (<< or >>) to move the highlighted field to "Create".

- Press the Up or Down Arrow key ($\uparrow$ or $\downarrow$) to move the highlighted field to the "Create Virtual Partition" field. Press $\langle$ Enter $\rangle$ to select this item and display the following screen:

- In the "DOS Partition Name" field, type the full UNIX path of the file you want to use as your virtual DOS partition. For example, to create a virtual DOS partition named */usr/joe/vpart*, type:

/usr/joe/vpart

Then press $\langle$ Tab $\rangle$ to move the highlight to the "Partition Size" field.

- Enter the size of the virtual partition, in kilobytes (1024 kilobytes equals 1 Mbyte.) The maximum size is 7 Mbytes; the recommended minimum size is 2.5 Mbytes. For example, to create a 2.5-Mbyte virtual DOS partition, type:

2560

Then press $\langle$ Tab $\rangle$ to move to the "Cancel" field.

- If you choose to cancel the operation and not create the virtual partition, press $\langle$ Enter $\rangle$ while "Cancel" is highlighted. Otherwise, check your entries for accuracy. If any information on the screen is incorrect, press $\langle$ Tab $\rangle$ to move to the field you want to change and retype the entry. (Press $\langle$ F5 $\rangle$ to clear a field completely.)

When the information is correct, press $\langle$ Tab $\rangle$ to highlight the "Create" field. Then press $\langle$ Enter $\rangle$. SCO Merge creates the virtual DOS partition you have selected.

Using virtual partitions

To use a virtual DOS partition, you must attach it as a drive between E: and I: when you start DOS. For example, to use the `/usr/joe/vpart` DOS partition as drive F:, you would start the DOS environment by typing:

```
dos +af:=/usr/joe/vpart
```

For the duration of the DOS environment, `/usr/joe/vpart` is accessible as DOS drive F:. Other SCO Merge users (or Joe himself) can simultaneously run other DOS sessions with `/usr/joe/vpart` attached as drive F: (or any other drive between E: and I:) provided no user writes to that drive. Simultaneous access to the same virtual partition is allowed only when all access is read-only. As soon as any DOS process writes to the virtual partition, that process has exclusive control of that virtual partition until it exits. When other processes attempt to access the same virtual partition, they receive the message, "Not ready drive F:"

You can use the `dosopt` command to automatically attach a DOS partition whenever you run DOS, but remember the restrictions concerning access whenever you use a virtual partition. Refer to the descriptions of the `dosopt` command in Chapter 5 and the `±a` option in Chapter 6 for further information.

Note that if you have a primary physical DOS partition, it is available by default as drive E: when you start a DOS process. Only one partition at a time is available as drive E:. If you want to use a virtual partition as drive E:, you can use the command `dos +ae:=pathname` even if you have a physical DOS partition, however, E: no longer references the primary physical DOS partition.

You should not use drives C:, D: or J: to attach virtual partitions. These drives have specific functions under SCO Merge. See the User's Guide for more information.

Administering virtual DOS partitions

Most of the information covered under "Administering Physical DOS Partitions" in this chapter also applies to virtual DOS partitions. A virtual DOS partition differs from a physical DOS partition in that a virtual DOS partition is a single file within the filesystem shared by DOS and the UNIX operating system. This single file contains a DOS filesystem within itself. A virtual DOS partition is not useful from the UNIX environment because the files within it cannot be accessed from the UNIX environment. However, you can move it around, back it up, or delete it using standard UNIX tools. Deleting it frees up space on the shared filesystem.

Using dosadmin to create a virtual DOS floppy

A virtual floppy disk is a UNIX file that has been formatted under DOS and contains a DOS volume in a size that corresponds to a standard DOS diskette. You may use a virtual floppy just as you would a physical diskette to store DOS files. You may also make it bootable so that you can start a DOS session from it.

NOTE When you boot from a virtual floppy, you normally cannot access the filesystem shared by DOS and the UNIX operating system.

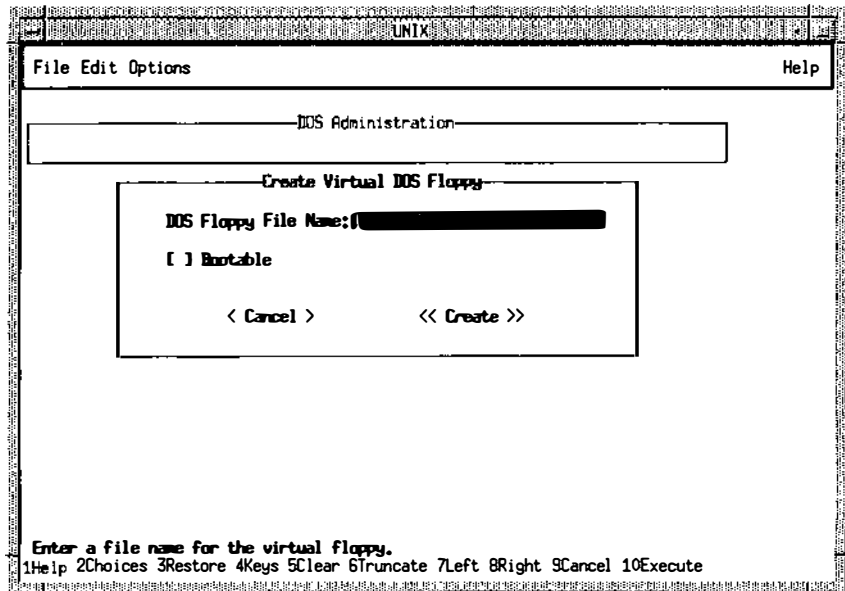
To create a virtual DOS floppy, use the following procedure:

1. Start **dosadmin** by typing:

dosadmin

(You may start **dosadmin** from either the DOS or UNIX command line.)
The **dosadmin** "DOS Administration" main menu appears.

2. Press the Left or Right Arrow key (<←> or <→>) to move the highlighted field to "Create".
3. Press the Up or Down Arrow key (<↑> or <↓>) to move the highlighted field to the "Create Virtual Floppy" field. Press <Enter> to select this item and display the following screen:



2. Administering SCO Merge

4. Enter the full UNIX pathname of the file you want to use for your virtual DOS floppy in the "DOS Floppy Name" field. For example, to create a virtual DOS floppy named */usr/joe/vflop*, type:

/usr/joe/vflop

5. If you want this floppy to be bootable, use the <Tab> key to move to the "Bootable" field and press <Enter> or <Space> to check the selection box next to it.
6. To create the virtual floppy, press <Tab> until the "Create" field is highlighted. Then press <Enter>. SCO Merge checks the size and density of your physical A: drive. The system creates and formats the virtual floppy to match the size and density of your physical drive. It is made bootable or nonbootable as you have specified.

If you decide to cancel the operation and not create a virtual floppy, press <Tab> until the "Cancel" field is highlighted and then press <Enter>.

To use the virtual floppy, you must attach it as floppy drive A: or B: when starting DOS. For example, to use the */usr/joe/vflop* virtual floppy as drive B:, you would start the DOS environment by typing:

dos +ab:=/usr/joe/vflop

When you attach a virtual floppy like this, you cannot access a physical DOS diskette drive with the same name (drive B: in this example). You can, however, access physical diskette drives with different names.

You can boot a virtual floppy attached as drive A: if you selected the "Bootable" option when you created the floppy. To boot a virtual floppy, use the **dosboot** command with the **+a** option to attach the virtual floppy to your DOS process. The syntax is:

dosboot +aa:= pathname

where *pathname* is the full path of the UNIX file containing the virtual floppy. For example:

dosboot +aa:=/usr/joe/vflop

dosboot is a synonym for the **dos -l** ("no load image") option. Refer to Chapter 6 for further information on this command.

Installing plug-in cards in your computer

SCO Merge allows DOS to use standard hardware devices in a convenient way. Standard DOS hardware devices include displays, disk drives, com ports, expanded memory (EMS), a mouse, printer ports, and the game port. A few pointers on administering several of these devices appear earlier in this chapter. Also refer to Chapter 6 (especially the description of the `+a` option) for information on using standard devices.

This section tells you how to install, configure, and use DOS hardware devices that are not included as part of SCO Merge. Examples of such devices are scanners and LAN cards. SCO Merge can use these devices only by communicating directly with them, without the intervention of the UNIX system. This form of device access is called direct attachment.

When you install new cards, the DOS Device Information Popup Window can be helpful. See the DOS sections of the User's Guide for further information on this utility. When you add, remove, or change computer hardware, you should make new DOS images. See "Making New DOS Images," later in this chapter, for more information.

Using directly attached DOS devices

Install your directly attached DOS device according to the manufacturer's instructions.

To use a directly attached DOS device, you must issue the `dos` command with the `+a` option. This attaches the device directly to the DOS process. The command structure is complex and you will probably choose not to issue the entire command from the command line every time you use the device. The next section, "Setting up `/etc/dosdev`," describes a method for tokenizing the command string. This allows you to issue simplified startup commands that directly attach your device.

The information in this section permits you to test direct attachment of your device from the command line until you have found the combination of options that works best. Typically, you will then tokenize the command string as described in the next section. All of the examples that follow assume that you are operating with root privileges.

The syntax for direct device attachment is:

+a *device_specification*

where *device_specification* consists of the following fields:

"io_port_range"	(typically in the range of 0-3ff, hex)
"interrupt_level"	(0, 2-7, 9-13, or 15)
"memory_mapped_io_range"	(typically c0000-effff, hex)
"dma_channel"	(0, 1, 2, 3, 4, 5, 6, 7)

The device specification fields must be listed in the order shown, but only those needed must be specified. Fields are separated by dots (.). The first field has a leading dot (between it and +a). When a field is not relevant, a single dot is used. An example of a command that starts the DOS environment and directly attaches a device is:

dos +a.2ff5..

Following are more complete descriptions of each of the device specification fields:

- "io_port_range" has this recursive definition:

X[-Y][,io_port_range]

where the form *X* specifies an I/O port address *X* (in hex) and the form *X-Y* specifies an I/O port address range, from low *X* (in hex) to high *Y* (in hex), that the device uses. As the syntax shows, single or multiple ports or ranges may be given.

- "interrupt_level" is a single decimal number that denotes which interrupt level is used.
- "memory_mapped_io_range" has this recursive definition:

X-Y[,memory_mapped_io_range]

where *X* and *Y* are two hex numbers, separated by a hyphen, denoting the range of the memory-mapped I/O. The first number is the lowest address and the second is the highest address. Each range must start on a 4K boundary and be a multiple of 4K. As the syntax shows, multiple ranges may be specified.

- "dma_channel" is a single decimal number that denotes which DMA channel is used. Note that devices that use certain DMA modes may not work.

Determine the values for "io_port_range", "interrupt_level", "memory_mapped_io_range", and "dma_channel" by consulting the hardware technical specifications for the devices being used. If you have difficulty determining the required values for your hardware, contact the hardware manufacturer.² The following example shows how to use the `+a` option to specify a directly attached device, in this case a hypothetical adapter card for a device called "widget."

- The widget card interrupt is 7, the ports range from 240 to 25f, and the device requires an 8K RAM buffer with a starting address of ce000. The device specification is therefore:

`.240-25f.7.ce000-cffff`

The last field, DMA channel, is blank because this field is not relevant for this adapter card. You can attach this device to DOS when you start the DOS environment with the command:

`dos +a.240-25f.7.ce000-cffff`

If you have an application called "runwidget" that uses this device, you can invoke the application and attach the device at the same time with the command:

`dos +a.240-25f.7.ce000-cffff runwidget`

NOTE Be sure that the interrupt, the port ranges, and the RAM address you specify do not conflict with any already existing DOS devices (such as the mouse driver or the EMS driver). If a conflict does exist, we suggest you change the settings on your hardware device.³ This example was specifically chosen to avoid conflicts with other DOS devices. If you try this example but you use settings that do not match those specified here, this example may not work.

Incorrect use of device specifications for directly attached devices can crash the SCO system. Therefore, only the system administrator (logged in as root or super user) is allowed to specify directly attached devices on the command line, as illustrated in the preceding examples. To enable other users to access directly attached devices, you must define tokens for the devices in the *dosdev* file. The next section describes this procedure.

2. You can determine the values currently in use on your system by running the UNIX `hwconfig` command.

3. If you need to change the default settings for the mouse driver, refer to the file `/etc/dosdev`. If you need to change the default settings for the EMS driver, see "Using Expanded Memory (EMS)" in Chapter 8 of this guide.

Setting up *letc/dosdev*

The SCO Merge device specification file */etc/dosdev* serves several purposes. *dosdev* includes specifications for several standard devices, including printers, com ports, and expanded memory. It also includes comments and example “template” device specifications that you might find useful as you set up *dosdev* specifications for other DOS devices.

Normally, you should not alter existing *dosdev* specifications. You can add your own device specifications to *dosdev*, though. The primary reasons for specifying a directly attached device in *dosdev* are:

- To allow SCO Merge users to directly attach devices. When you specify a device in *dosdev*, users other than root or super user can directly attach devices.
- To create a short, easily remembered alias (or token) for a device specification. You and other users can use this token with the **dos +a** option instead of fully specifying the device as described in the previous section.

Before specifying a directly attached device in *dosdev*, you should test the device specification using the **+a** option as described in the previous section. Because incorrect use of device specifications for directly attached devices can crash the SCO system, you should conduct your tests in system maintenance mode and at a time when users are not inconvenienced if you need to reboot the system.

The syntax for specifying a directly attached device in *dosdev* is:

token d device_specification comments

Tokens may not contain spaces, commas, or equal signs and may not start with a slash or dot. Tokens also may not resemble a drive letter specification (a single letter followed by a colon). Aside from these restrictions, you can use any alphanumeric characters in tokens. Tokens can be up to 32 characters in length. For example, the *dosdev* line for the directly attached widget card discussed in the previous section might be:

```
widget d .240-25f.7.ce000-cffff. Direct attach widget card.
```

This line can be anywhere in *dosdev*. As long as this line exists in *dosdev*, users can directly attach the widget card with the command:

dos +awidget

Users can also use the **dosopt** command to cause specific applications or the **dos** command to request access to a directly attached device automatically. Use the token defined in *dosdev* when issuing the **dosopt** command. For example:

dosopt +awidget runwidge

When you use **dosopt** like this, SCO Merge looks up the specification for the token ("widget" in this example) in *dosdev* when you run DOS, not when you assign the option with **dosopt**. Therefore, if you change the device specification associated with the token in *dosdev*, SCO Merge always uses the current specification. Refer to Chapter 5 for further information on **dosopt**.

Note that directly attached devices cannot be shared. As long as one process is using a directly attached device, other processes are prevented from using the device until the process controlling the device exits.

Mapping device names

The SCO system allows DOS to access a hardware device via different device specifications than the physical device uses. For instance, assume you have a communications program that uses com1 by default. If you decide you want the application to use the physical com2 port instead, you can use the mapping feature to make the application treat com2 as though it were com1, without reconfiguring the application or physically moving hardware. This feature is useful whenever you want DOS to view a hardware device as though it were a different physical device.

To make DOS view a hardware device as though it had different specifications, you map the virtual device specification (the specification that DOS sees) to the physical device specification (the actual physical characteristics of the device). The syntax for mapping device names is:

+avirtual_device_specification=physical_device_specification

virtual_device_specification and *physical_device_specification* can be:

- Complete device specifications (including fields for I/O port range, interrupt level, memory-mapped I/O range, and DMA channel).
- Tokens that are defined in *dosdev*.

For example, the tokens **dcom1** and **dcom2** are defined by default in *dosdev*, so you can start the DOS environment and map com1 to com2 with the command:

dos +adcom1=dcom2

This command causes DOS to treat the physical com2 port as though it were com1.

There is no difference in performance when you map virtual and physical DOS devices like this, except for I/O ports, for which it is more efficient to make the virtual and physical port ranges the same.

Limitations of direct attachment

While the foregoing sections describe the steps for attaching devices directly to SCO Merge, we cannot guarantee that all devices attached this way will work.

There are many factors that may prevent a device from working under SCO Merge. For example, devices that use certain DMA modes may not work. Tape drives are a prime example.

Devices that are highly timing-dependent may fail under the SCO system's multitasking environment. Some devices need to be initialized or reset when the system is booted, and these devices may not work on a DOS invocation. Shared interrupts for devices are not supported on Micro Channel Architectures (MCA) and this may hinder other devices.

Directly attached devices are also sensitive to various configuration changes. Changing the environment — adding more users, changing the load, tuning UNIX parameters, adding or subtracting memory, or any number of other environment changes — can have unpredictable effects on the operation of direct-attached devices.

Making new DOS images

The DOS image is a file that reflects the configuration of your virtual PC environment. The DOS image is a frozen picture, or snapshot, of DOS after it has been booted and loaded into memory and begun running. This image includes information DOS needs about the system configuration, derived from the hardware installed on the system and the contents of the configuration file `/usr/merge/image/config.mki` at the time the image is made. It allows quick startup of a DOS program from the UNIX environment. The default DOS images (for monochrome, CGA, and VGA display adapters) are contained in the following files.

- `/usr/merge/image/std/mono/*.img`,⁴
- `/usr/merge/image/std/cga/*.img`,
- `/usr/merge/image/std/vga/*.img`.

These DOS image files are used by all SCO Merge users whenever they run any DOS process unless they specifically request a custom image when they start DOS.

4. Hercules display adapters also use the file `/usr/merge/image/std/mono/mono.img`.

There are several reasons to make new DOS images. Typically, when new images are required, the system administrator creates a new default DOS image for each type of display adapter used on the SCO system. The new default images are then available automatically to all DOS users. The following subsections describe when you need to create new DOS images and how to create them. The topics are:

- Hardware Changes that Require New DOS Images
- Changing Special `config.sys` Commands
- Using `mkimg` to Make New DOS Images
- Using `dosadmin` to Make New DOS Images

Hardware changes that require new DOS images

The most common reason to make new DOS images is that you have changed your computer hardware. New DOS images are required whenever you make hardware changes that change the BIOS data area or interrupt vectors. You may not know whether a particular hardware change has had these effects. Making new DOS images is simple, however, and it never hurts to make new images even when they are not required. We therefore recommend that you make new default DOS images for all types of displays used on your system after you make any hardware change.

Standard ROMs and on-card ROMs

When you create a DOS image for a VGA display adapter card, SCO Merge uses the data in the display adapter ROM. SCO Merge can use either the actual physical ROM on the card while making the image, or a standard ROM supplied (as a file) with the SCO Merge software. The images created during the SCO Merge installation procedure use standard ROM files, rather than the on-card ROMs, whenever standard ROMs exist. When you make a new image following the initial SCO Merge installation, you may be required to choose between a standard ROM and an on-card ROM. In most cases, you should use the standard ROMs. The image-making process is more reliable when you use standard ROMs. The only time to choose on-card ROMs is when you are using a display adapter with non-standard functionality and you want to take advantage of that functionality.

NOTE SCO Merge may not operate correctly when you use nonstandard video modes. Because extended video modes can vary in unpredictable ways (in their treatment of registers in particular), it is impossible to predict how SCO Merge will work when you use these modes.

To create a new DOS image, follow the procedures in "Using `mkimg` to make new DOS images" or in "Using `dosadmin` to make new DOS images," later in this chapter. "Using `mkimg` to make new DOS images" describes how to use

the command-line utility **mkimg** to make DOS images. “Using **dosadmin** to make new DOS images” describes how to use the **dosadmin** menu utility to make DOS images.

Changing special config.sys commands

SCO Merge allows you to use nearly all **config.sys** commands as you would on a conventional DOS computer. Commands in the *root* directory *config.sys* file affect all users when they start a DOS process. Commands in a user’s home directory *config.sys* file affect only that user’s DOS processes.

On the SCO system, however, DOS does not interpret certain *config.sys* commands unless they are incorporated into the DOS image you are using. These are **BUFFERS**, **DOS**, **FCBS**, **LASTDRIVE**, **STACKS**, and **SWITCHES**. DOS ignores these commands in the *root* directory and home directory *config.sys* files and any other *config.sys* files you specify with the **+e** option at DOS runtime.

Incorporating *config.sys* commands into your DOS images also enhances startup. If there are certain *config.sys* commands you always execute with every DOS session you start on your system, you should include those in the system’s images. This reduces startup time for all system users.

The factory default DOS images supplied with SCO Merge already contain nearly everything you will need. The only alterations you may wish to make are the addition of device drivers and a different value for the **LASTDRIVE** command. After making these alterations you create one or more new DOS images, one for each type of display to which you want the changes to apply. The new images can be made with the **mkimg** utility or with **dosadmin**.

Adding device drivers to DOS images

Never edit the image configuration file */usr/merge/image/config.mki* directly. This file is used only in the DOS image-creation process. It includes configuration information (such as device driver definitions) that are needed whenever users run DOS processes. The factory default images are made on the basis of the entries in this file. Altering any of its lines is likely to result in unexpected and undesirable SCO Merge behavior.

Below */usr/merge/image/std* are three special directories created specifically for the making new images. There is one for each of the images you might need to make. They are:

- */usr/merge/image/std/mono*
- */usr/merge/image/std/cga*
- */usr/merge/image/std/vga*

In each directory is a file called *config.sys* which contains only a single **LAST-DRIVE=** line. Add your device driver lines to these *config.sys* files and alter the **LASTDRIVE** value if you wish. Copy the DOS device driver files to the appropriate directory for each of the display types to which you want that device added.

The images you make with **mkimg** or **dosadmin** are constructed on the basis of the lines in the image configuration file plus the entries in these *config.sys* files.

Using mkimg to make new DOS images

You can use the **mkimg** utility instead of **dosadmin** to make new DOS images when you want to make new DOS images for all display cards and you want all images to be made using the standard ROM. To make new DOS images using **mkimg**, follow these steps:

1. Log in as root or become the super user. (Note that you can only use the **mkimg** program from the UNIX command line.)
2. Type:

mkimg

The **mkimg** utility creates new DOS images for all display cards.

Using dosadmin to make new DOS images

To make new DOS images using **dosadmin**, follow these steps:

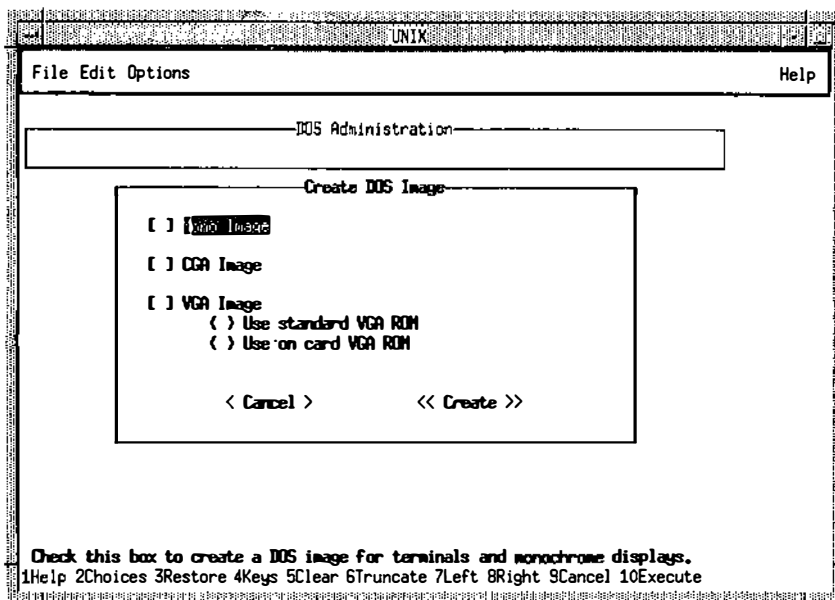
1. Log in as root or become the super user.
2. Start **dosadmin** by typing:

dosadmin

(You may start **dosadmin** from either the DOS or UNIX command line.)
The **dosadmin** "DOS Administration" main menu appears.

3. Press the Left or Right Arrow key (←) or (→) to move the highlighted field to "Create." A submenu opens under "Create."

4. If the "Create DOS Image" field is not highlighted, press the Up or Down Arrow key ($\uparrow$) or ($\downarrow$) to move the highlighted field to "Create DOS Image." Then press $\langle$ Enter $\rangle$ to select this item and display the following screen:



There are fields for each type of image you can create. These fields vary, depending on your computer hardware. For example, fields for VGA images appear only if you have a VGA card installed.

5. Use the $\langle$ Tab $\rangle$ key to move the highlight from one field to another, and press $\langle$ Enter $\rangle$ or $\langle$ Space $\rangle$ to check the selection box for each image you want to create. You can select any or all of the displayed image types.
6. If your system allows you to create images that use either a standard ROM or an on-card ROM, the **dosadmin** screen offers you a choice between the two types of ROM. The default is to use the standard ROM, so by default this selection has an asterisk next to it. If you want to use the on-card ROM, use the Down Arrow key ($\downarrow$) to move the highlight to the "Use on-card ROM" field. Then press $\langle$ Enter $\rangle$ or the $\langle$ Space $\rangle$ to select this field. An asterisk appears between the parentheses.
7. If you decide not to complete this procedure, press $\langle$ Tab $\rangle$ to move the highlight into the "Cancel" field and press $\langle$ Enter $\rangle$. Otherwise, to complete the procedure, press $\langle$ Tab $\rangle$ until "Create" is highlighted, then press $\langle$ Enter $\rangle$.

System files affected by system administration

For your reference, the following is a list of SCO Merge files affected by common administrative procedures. These system files are used by the general SCO Merge user community. Most of these files are described earlier in this chapter. Refer to Chapter 5 in this guide for further information on */etc/dosenv.def*.

- */autoexec.bat*: system-wide *autoexec.bat* file (affects all users)
- */config.sys*: system-wide *config.sys* file (affects all users)
- */etc/dosenv.def*: system-wide DOS environment file (affects all users)
- */usr/merge/image/config.mki*: *config.sys* file used for all DOS images
- */etc/dosdev*: DOS device definitions file
- */usr/merge/image/std/mono/mono.img*: default image for starting DOS with a monochrome or Hercules display adapter
- */usr/merge/image/std/cga/cga.img*: default image for starting DOS with a CGA display adapter
- */usr/merge/image/std/vga/vga.img*: default image for starting DOS with a VGA display adapter

Be sure that you inform all affected users (via email or message of the day) whenever you modify any of these files.

2. *Administering SCO Merge*

Chapter 3

Installing DOS applications

This chapter tells you how to install and configure DOS applications on the fixed disk, as well as how to remove them. In general, you can install most applications simply by entering the DOS environment as described in the SCO Merge chapters in *Using SCO Merge* and following the application manufacturer's instructions for installing on a fixed disk. You can then configure your application so that it can be invoked directly from the UNIX shell as well as from DOS. This configuration may be done either manually or with the **dosadmin** program. Both procedures are described in this chapter.

A few copy-protected applications cannot be installed on the filesystem shared by DOS and the UNIX operating system. You can install these applications on the DOS partition. For further information on this subject, see "Installing copy-protected programs," later in this chapter.

Some applications cannot be installed on the fixed disk at all because they boot their own operating systems from the distributed diskette to run the program. To run these applications, use the SCO Merge **dosboot** command as described in the SCO Merge chapters in *Using SCO Merge*.

Other applications require processor facilities not available in the Virtual 8086 environment created by SCO Merge. These can be installed on a real DOS partition, but cannot be configured to run under SCO Merge. The sole exception is Microsoft Windows 3.1, for which SCO Merge simulates the Protected Mode of 80x86 processors.

This chapter includes the following sections:

- Tips on installing DOS applications to run under SCO Merge
- Installing applications with advanced processor requirements
- Configuring DOS applications for use from the UNIX shell
- Installing copy-protected DOS applications
- Removing DOS applications

Tips on installing DOS applications to run under SCO Merge

The tips offered here assume that you have started a DOS environment under SCO Merge and are preparing to install your application according to the manufacturer's instructions.

Note that this discussion assumes that you are installing an application that requires its own subdirectory on the fixed disk. Simple DOS applications or commands (those that have a single *.exe* or *.com* file and do not require their own subdirectories) can be directly copied into the user's home directory or into the directory *C:\usr\ldbin*, rather than requiring an installation procedure.

Before installing an application, decide if the application is for personal use (for use by yourself in your own subdirectory) or for public use (accessible by multiple users and installed in a public subdirectory). It is generally recommended that you install applications for personal use, as the procedure for this type of installation is easier. However, if you have a multiuser system and you are short on disk space, you may prefer to install the application for public use. The next section gives tips on installing an application in a personal directory. The section following that gives tips on installing an application for public use.

Installing personal DOS applications

To install a personal DOS application, log into your system as yourself, start a DOS session, and follow the normal installation procedures for the application, taking into consideration the suggestions offered below.

Most DOS applications are installed with a program called "install" or "setup" that must be run from an installation diskette. These installation programs generally ask users to specify the following:

- The drive and directory on which the application is to be installed
- The display adapter type
- The printer type
- The mouse type

The installation program may also ask for permission to modify *autoexec.bat* and *config.sys* files. A few installation programs may offer to configure the application so it can run in 80286 or 80386 protected mode.

Following are some tips on how to respond to the prompts for these items of information.

- **Specifying the drive and directory:** The installation program often displays C:\ as the default drive and directory. Under SCO Merge, C:\ is the system *root* directory, and since you are not logged in as root, you do not have write permission there. We recommend that you specify a subdirectory under your home directory as the installation location. For example, if your home directory is C:\usr\pat and you are installing Lotus® 1-2-3®, you might specify C:\usr\pat\lotus as the program subdirectory.

If the application does not allow you to specify a subdirectory (that is, the application insists on being installed in the *root* directory of a drive) then specify the D: drive.¹ Under SCO Merge, D:\ is automatically substituted to your home directory with the DOS *subst* command. (Refer to the MS-DOS manual for a description of the *subst* command.)

- **Specifying the display:** Most applications automatically detect the type of display you have. If the one you are installing does not, specify your display adapter type here (for example, VGA).

VGA display modes are only supported in X Windows zoom mode (or on the console when you are not using the X environment). If you plan to run programs remotely (on other servers, where zooming is not permitted), if you do not want to zoom, or even when running X locally, you may want to specify CGA or HERCULES as your display adapter.

If you want to run an application in several different graphics modes, you can change the graphics mode by starting DOS with different video images (that is, *dos +acga*, *dos +aherc*, *dos +avga*).

1. Note that if you install an application on the D: drive, you may be required to run it from the D: drive.

- **Specifying the printer:** Specify lpt1, lpt2, or lpt3 as your printer port. Specify the type of printer attached to your UNIX system as the printer type.

By default, under SCO Merge, all DOS print requests to lpt1, lpt2 and lpt3 go to a UNIX print spooler named **doslp**. This print spooler sends its requests to the printer attached to your computer.

- **Specifying the mouse:** The mouse driver provided with SCO Merge is especially designed to provide optimum performance in a multitasking environment, and is automatically available to every DOS session you start. Respond "no" if the application asks whether it should install its own mouse driver.
- **autoexec.bat and config.sys Editing:** Some installation programs ask you if they can modify the standard *autoexec.bat* and *config.sys* files. Always answer "no" to this question and make any necessary modifications to your personal *autoexec.bat* and *config.sys* files in your own home directory. Consult the program documentation for a description of the modifications that need to be made.
- **Protected mode options:** If the installation program offers to configure the application for a specific processor, always choose 8086. SCO Merge does not support applications that require the features of 80286 or 80386 processors. In particular, SCO Merge can not run DOS programs that require protected mode. The one exception is Windows 3.1, which is supported in standard mode. See "Installing applications with advanced processor requirements" later in this chapter.

Installing public DOS applications

The tips listed above for installing DOS applications for personal use apply also to applications installed for public use, with the following differences:

- Instead of logging in as yourself, log in as root.
- For the installation directory, specify a subdirectory under *C:\usr\ldbin*, the system application directory. For example, if you are installing Lotus 1-2-3, you might specify *C:\usr\ldbin\lotus*. If the application prompts only for the drive and not for a directory, specify the J: drive. Under SCO Merge, J:\ is automatically substituted to *C:\usr\ldbin* with the DOS **subst** command. (Refer to the MS-DOS manual for a description of the **subst** command.)
- If the program's installation routine requests permission to modify the system-wide *autoexec.bat* and *config.sys* files, use your own discretion. Remember that you, as system administrator, are responsible for cleaning up any damage that a poorly written installation routine might do.

Before you let the **install** routine do any modifications, be sure to make backup copies of both files. When it has done its work, scrutinize the changes carefully by comparing the changed files to the backups.

Your alternative is to determine what changes the installation routine makes and generate those same changes with an editor.

- When you have completed the installation, you may need to use the UNIX **chmod** command to give appropriate read, write, and execute permissions to the application subdirectory and its files so that they are accessible to other users.

Defining the search path

After you have installed a personal or public DOS application, you may want to update the search path. If you have installed a personal DOS application, you can change the UNIX search path that is defined whenever you log in by modifying your home directory *.profile* file. Note that you do not include DOS drive designations (such as C:) in UNIX path definitions. By default, the UNIX search path applies in the DOS environment also. Alternatively, you can set the **PATH** variable in your *autoexec.bat* file to define your search path in the DOS environment.

If you have installed a public DOS application, make sure the directory where it is installed is in users' search paths. If necessary, you can update the search path for all users by changing the **PATH** definition in */etc/default/login*. By default, the UNIX search path applies both in the UNIX shell and in the DOS environment. Alternatively, you can set the **PATH** variable in the *root* directory *autoexec.bat* file.

Refer to Chapter 8 for additional techniques for managing your search path.

Installing applications with advanced processor requirements

SCO Merge creates a two kinds of DOS environment for your DOS software. For Windows 3.1 and its applications, SCO Merge creates a Virtual 286 environment which offers access to up to 15 Mbytes of memory. Only Windows 3.1 and its applications can access this feature.

For all other DOS programs, SCO Merge creates a 1 Mbyte DOS environment. This is done by invoking the Virtual 8086 mode of the Intel 80386 or 80486 processor.

Any application which you run in this standard V86 environment must be compatible with the Intel 8086 processor. Any DOS application which absolutely requires an 80286 or later processor does not run under SCO Merge. The sole exception is Windows 3.1 and its applications, which are supported in the Standard Mode of the 80286 processor.

For further discussion of memory and how your DOS programs can use it, see "Allocating memory for DOS" in Chapter 5. For more on processor-dependent functions, "Protected mode restrictions" in Chapter 8.

Most DOS applications are meant for the 8086. Of those designed for more powerful processors, many are smart enough to sense the nature of the processor on which they are running and adjust their operation accordingly. For these "intelligent applications" you need to do nothing special. They respond to the V86 environment as an 8086 processor and automatically run in the 8086-compatible "real mode."

For non-Windows applications that offer 286/386 features and fail to adjust to the V86 automatically, there are several ways to compensate:

1. **Install the application for the 8086.** Many DOS applications offer a backward-compatible installation option. The installation routine permits you to specify which processor the application is to run on. By choosing the 8086, your application may run slower or without some of its more advanced features, but it runs under SCO Merge.
2. **Choose an earlier version of the program.** If the most current version of the application requires an 80286 or 80386 processor, investigate earlier releases of the software. You may find that an earlier version offers the features you want and is 8086-compatible. For instance, Lotus 1-2-3 Version 2.2 runs under SCO Merge; Version 3.0 does not.
3. **Install and run under "raw" DOS.** If you have a DOS partition and you are willing to shut down your UNIX system every time you want to run the application, you can install your protected mode application under "raw" DOS. You must shut down the UNIX system, boot "raw" DOS from the DOS partition and install your application. The general procedure is described in an upcoming section.

The same UNIX shutdown is required every time you run the application. Only a single-user system should consider this solution.

Configuring DOS applications for use from the UNIX shell

Once you have installed a DOS application under SCO Merge, it is available to you whenever you are using the DOS environment. This section tells you how to configure DOS applications so that you can also invoke them from the UNIX shell. When an application is configured through one of the methods described here, you can invoke it simply by typing its name at the UNIX prompt.

This section describes two methods of configuring DOS applications for use from the UNIX shell: by using the **dosadmin** utility, and manually (using UNIX commands).

Using *dosadmin* to configure DOS applications

Following are step-by-step instructions for using **dosadmin** to configure DOS applications to be used from the UNIX shell. You follow the same procedures whether you are configuring a personal-use application or a public-use application.

To configure an already-installed application using **dosadmin**, follow these procedures:

1. If you are configuring an application for personal use, log in to your system as yourself. If you are configuring an application for public use, log in as root.
2. Type:
dosadmin
(You may start **dosadmin** from either the DOS or UNIX command line.)
The **dosadmin** main menu is displayed.
3. The "Applications" submenu should be highlighted. If it is not, press the <Left Arrow> or <Right Arrow> key (← or →) until it is.

4. Press <Enter> to select "Install". The "Install DOS application for UNIX" menu appears, and your display looks like this:

rothko shell

File Edit Options Help

DOS Administration

Install DOS Application for UNIX

Application Directory:

DOS Executable File:

< Cancel > << Install >>

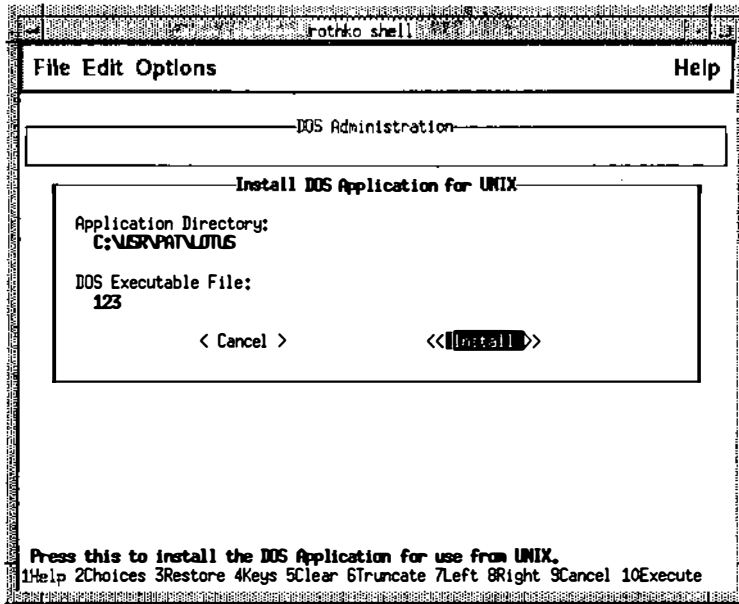
Enter the directory name where the application is installed.
1Help 2Choices 3Restore 4Keys 5Clear 6Truncate 7Left 8Right 9Cancel 10Execute

5. The "Application Directory" field is highlighted. Type in the full path of the directory where you installed the application. For example, type `C:\usr\pat\lotus`.

Press the <Tab> key to move to the next field.

6. The "DOS Executable File" field is highlighted. Type in the name of the executable DOS program: that is, the name of the command you use when you run the program. It is not necessary to include the filename extension (for example, `.exe` or `.com`); `dosadmin` automatically adds the extension if you do not provide it. For example, type `123` if this is the command you use to run the program.
7. Check the fields of the display for accuracy. To make changes, press the <Tab> key repeatedly until the field you want to change is highlighted. Then type in the correct information.

8. When the display shows correct information, press the <Tab> key until "Install" is highlighted. The display should resemble this:



9. Press <Enter> to complete the configuration procedure. An "In Progress" message appears while **dosadmin** configures your application. Then the top-level **dosadmin** menu returns.
10. If you want to configure more executable files, simply repeat steps 3 through 9, providing the appropriate DOS directory name and executable filename at the prompts.

To return to your system prompt, press the <Left Arrow> or <Right Arrow> key (← or →) until "Exit" is highlighted, and press <Enter>.

Manually configuring DOS applications

The previous section described how to make a DOS application executable from the UNIX prompt by using **dosadmin**. Users can manually accomplish the same task by executing a series of UNIX commands. Following are step-by-step instructions for manually performing this configuration. The procedure is illustrated using the DOS program Lotus 1-2-3 as a typical example, and assumes that you want to start the program by typing 123. (Note that you must perform this procedure from the UNIX shell.)

1. Log in to your system.

2. Change to the directory where the DOS executable program resides.
3. Use the **dosopt** command with the **+y** option to add the default DOS options record to the program:

dosopt +y 123.exe

4. Set permission modes so that the UNIX system recognizes the program as an executable:

chmod +x 123.exe

5. Link the DOS program name to a UNIX system name with no *.bat*, *.com*, or *.exe* extension:

ln 123.exe 123

You can now execute the program from the UNIX prompt by typing its name:

123

You can perform further configuration with the **dosopt** command. For example, to make 1 Mbyte of EMS memory available every time you run Lotus 1-2-3 from the UNIX prompt, you would execute the command:

dosopt +aems 123

Refer to Chapters 5 and 6 for further information on DOS options and the **dosopt** command.

Installing copy-protected DOS applications

A few DOS applications are designed by the manufacturer to prevent installation on more than one machine or execution by more than one user at a time. You can use these applications with SCO systems, but the installation procedures differ according to the type of copy protection used.

Many different copy-protection methods are in use, and it is impossible to recommend installation procedures that apply universally. There are, however, two common classes of copy-protected applications for which we can supply general recommendations that apply in most cases. These classes are:

- applications that use "key disks."
- applications that use special installation utilities.

Procedures for installing applications in each of these classes are described in the following sections.

Using a key disk

An application that uses a key disk allows you to install part of the application on the system fixed disk but requires you to insert a protected diskette in the diskette drive whenever you use the application.

Such applications generally require no special installation procedures to work on SCO systems. You can install them by running a DOS environment and following the manufacturer's instructions. When you run an application that requires a key disk, you use the key disk just as you would on a conventional personal computer running DOS.

Using special installation procedures

Another class of copy-protected applications uses special installation (and sometimes also removal) procedures to prevent illegal copying of the application. These applications typically include a special installation utility (often invoked with a command such as "install"). They are intended to be installed in their entirety on the system fixed disk so that no key disk is required. You can install such applications on SCO systems but, because different copy-protection methods are in use, you may have to try more than one installation procedure. The procedures, listed in the order you should try them, are:

- Install the application according to the standard instructions under "Installing DOS applications" earlier in this chapter.
- Install the application on a DOS partition (for example drive E:). This procedure is described in the next section, "Installing DOS applications on a DOS partition."
- Shut down your system, boot DOS, and install the application on a DOS partition. This procedure is also described in the next section.

Installing DOS applications on a DOS partition

This section describes two methods of installing DOS applications on a DOS partition. These procedures are particularly useful when you install copy-protected applications that cannot be installed on the filesystem shared by DOS and the UNIX operating system.

Installing DOS applications under SCO systems

This procedure is required for a few copy-protected DOS applications that must be installed on an actual DOS filesystem rather than on the filesystem shared by DOS and the UNIX operating system. This installation method works only if you have a DOS partition on your system. A DOS partition can be a physical partition (that can be booted and run independently of SCO

systems), or it can be a virtual partition. If your system does not have a DOS partition, you can create one as described in Chapter 2 of this guide.

NOTE These instructions assume you are installing an application on drive E:, which is automatically attached when you start a DOS environment. If you are installing on the extended DOS partition or on a virtual DOS partition, you must first attach the partition to your DOS process. This procedure is described in "Physical DOS partitions" (page 87) and "Using virtual partitions" (page 26) in this book and also in *Using SCO Merge*.

If you are using a drive other than drive E:, substitute that drive letter when performing the steps below.

1. Log in to your system.
2. Start the DOS environment by typing (at the UNIX prompt):

dos

Your prompt changes to the DOS prompt C:.

3. Change your working drive to drive E: by typing:

e:

4. Make sure you are working in the directory in which you want to install the application. If necessary, use the DOS **cd** (change directory) command to move to the desired directory.

If you plan to install an application in a physical DOS partition, you must have read and write permission for that partition. (By default, all users have read and write permission on physical DOS partitions.)

5. Install the DOS application according to the manufacturer's instructions. For example, if the application has an installation program named **install**, invoke it by entering (at your DOS prompt):

install

(If the manufacturer provides no other installation instructions, use the DOS **copy** command to transfer all files from drive A: to drive E:.)

The application is now usable from DOS drive E:.

Installing applications under "raw" DOS

Use this procedure for the few copy-protected DOS applications with special timing requirements that make them impossible to install while SCO Merge is running.

NOTE This method works only if your system has a physical DOS partition that can run DOS as a stand-alone operating system.

Since you boot your system under DOS during this procedure, you must have one of the following:

- A bootable DOS system diskette. If you do not already have such a diskette, you can create one by inserting a diskette into drive A: and typing:

dos format a: -s

- A bootable DOS partition. (It is bootable if you formatted it using the **format /s** option.)

Follow these procedures:

1. Shut down the UNIX system according to the instructions in Administrator's Guide.
2. Reboot the system from a bootable DOS diskette or from a DOS partition.
3. Follow the manufacturer's instructions to install the application on a DOS partition.
4. When you are finished installing the application, reboot the UNIX system.

The application should now be usable like any other application on a DOS partition.

Some DOS applications must be installed and run on the same drive. They make a record of the drive letter on which they are installed and fail if you try to run them from any other drive. When you run raw DOS, DOS usually identifies the DOS partition as Drive C:. When you run under SCO Merge, that same partition is referenced as Drive E:. The application therefore fails when run from SCO Merge.

You can avert this situation by making the application believe it is being installed on Drive E: Thereafter, when you run it from E:, it behaves properly.

After booting raw DOS, use the DOS **subst** command as follows:

subst e: c:

Then follow the manufacturer's instructions for installing the application on drive E:

When the installation is complete, return the drive to its normal naming conventions with the following command:

subst c: e:

Each time you run this application, it must believe it is running on Drive E: When you run it from SCO Merge the drive is automatically referenced as E:. Any time you run it from raw DOS, however, you must once again evoke the **subst** command shown above. You can do that in a batch file, which then runs the application; or you can do it in the *autoexec.bat* on the DOS partition.

Which method you choose depends on the requirements of the other applications installed on that partition.

Removing DOS applications

Following is an outline of the procedures you follow to remove installed DOS applications from the fixed disk. Each of these steps is described in the following sections.

- Remove the application according to the manufacturer's instructions.
- Remove UNIX links to DOS executable files.
- Redefine the search path, if necessary.

Removing the application from the fixed disk

Remove a DOS application from the fixed disk by following the manufacturer's instructions. If no instructions are provided, it is usually safe to remove the application's files using the UNIX `rm` or DOS `del` command.

Note that some copy-protected applications require special removal procedures. If you do not follow the required procedure, you may not be able to reinstall the application at a future time.

Deleting UNIX links

If the DOS application you have removed was configured for execution from the UNIX shell, there are UNIX files that were linked to DOS executable files. (For example, `123` would be linked to `123.com`.) When you remove DOS executable files (files with extensions ending in `.com`, `.exe`, or `.bat`), the corresponding UNIX links are not automatically removed. When you remove a DOS application, you should also delete any UNIX links to the DOS files you are removing. To delete these links, you simply use the UNIX `rm` or DOS `del` command. For example:

```
rm 123
```

Redefining the search path

After you have removed a DOS application, you may want to redefine your search path. Depending on whether the application you removed was a public or a personal application and on how you define your search path, you might want to modify the `PATH` definitions in `/etc/default/login`, your home directory `.profile` file, or an `autoexec.bat` file.

Chapter 4

Tuning applications with the *merge* command

The **merge** command allows you to tune the operation of the DOS process for certain applications. You use the **merge** command to set the following characteristics of the DOS environment from within the environment:

- You can set the clock interrupt rate that SCO Merge uses.
- You can keep programs that poll the keyboard from being put to sleep.
- You can configure the DOS environment so DOS interprets virtual drives or DOS file handles as being either local or remote.

The syntax of the **merge** command is:

```
merge set fastclk on | off  
merge set pollsleep on | off  
merge set drive local | remote  
merge set handle local | remote  
merge set netbios on | off  
merge set 5f02 on | off  
merge set lpt_binary on | off  
merge display [option]  
merge return option
```

These forms of the **merge** command are described below.

fastclk option

The **merge fastclk** option determines whether SCO Merge uses the default clock interrupt rate or a lower interrupt rate. The syntax for this option is:

merge set fastclk on | off

The SCO Merge default is **fastclk on**.

By default, SCO Merge uses a clock interrupt rate of approximately 18 per second for DOS, which is the same as a conventional DOS computer. You can improve the performance of most DOS commands and applications by using the **fastclk off** option to reduce the interrupt rate to one per second. To reduce the interrupt rate for your current DOS session, type:

merge set fastclk off

Some DOS applications sensitive to the clock interrupt rate do not run correctly at the lower rate. If you run one of these applications with **fastclk** set to **off**, you may notice symptoms such as:

- The application runs much more slowly than expected.
- The application hangs.

If you experience one of these problems and it is caused by the interrupt rate, you can correct it by using the **merge** command to set **fastclk** back to **on**.

To set **fastclk on**, type:

merge set fastclk on

pollsleep option

The **merge pollsleep** option determines whether programs that poll the keyboard are put to sleep when they are idle. The syntax for this option is:

merge set pollsleep on | off

The SCO Merge default is **pollsleep on**.

DOS applications that poll the keyboard can consume system resources even when they are idle by entering a polling loop. Applications that enter a polling loop include (but are not limited to) Lotus 1-2-3, dBASE III®, and WordPerfect®.

If not compensated for, these applications reduce system performance because when they poll the keyboard, less CPU time is available to other concurrently running processes. If you run more than one such application at once, overall system response can degrade to an unacceptable level. By default, SCO Merge corrects this problem by putting most applications that enter a polling loop to sleep until there is keyboard input.

Some applications may appear to be in a polling loop when they actually are not. By default, the **pollsleep** feature might put these applications to sleep inappropriately, which would not adversely affect other processes, but would cause these applications to perform poorly. This situation is not common, but if you encounter it, you might want to disable the **pollsleep** feature.

To disable this feature, type at the DOS prompt:

merge set pollsleep off

merge network control options

The **merge** command **drive**, **handle**, **netbios**, and **5f02** options all control how SCO Merge behaves when DOS applications issue specific system calls usually associated with network transactions. These system calls test whether DOS files or drives are local or remote, whether NetBIOS is available, and whether other network conditions are true or false. SCO Merge itself does not include any special support for any particular PC networks. From the point of view of some DOS applications, however, the SCO Merge filesystem shared by DOS and the UNIX operating system is a “network” device. These applications use various DOS system calls to determine how (or whether) they should operate. An application may enable or disable file locking, for example, based on its tests of network status.

The four **merge** command options described in the following sections all have defaults that allow DOS applications to run correctly when they access the shared file system. No known applications are adversely affected by the default values for these options. Although unlikely, you may discover an application that does not behave as expected unless you change the value of one or more of these options. We cannot predict specific symptoms that might require changing the value of one of these options because different DOS applications issue network system calls for different reasons.

drive option

Some DOS applications detect whether a DOS drive is local or remote. These applications may not operate correctly if a DOS drive is local when the application expects it to be remote or vice versa.

To configure your DOS environment so DOS interprets SCO Merge drives that refer to the shared filesystem (drives C:, D:, and J:, for example) to be local or remote, use the **merge** command with the following syntax:

merge set drive local | remote

The SCO Merge default is **drive remote**. To make DOS interpret the filesystem shared by DOS the UNIX system as local, type:

merge set drive local

The **merge set drive local** command causes DOS applications that use DOS interrupt 21H system call function 44H (IOCTL) code 09H (Is Logical Device Local or Remote) to interpret SCO Merge drives that refer to the shared filesystem (such as C:, D:, or J:) as local drives. The **merge set drive remote** command causes DOS applications using the same system call to interpret the same SCO Merge drives as remote drives. Use these **merge** commands when necessary to configure your DOS environment so applications that are sensitive to local and remote drives work properly.

The following section describes the related **set handle** option.

handle option

Some DOS applications detect whether a DOS file is local or remote. These applications may not operate correctly if a DOS file is local when the application expects it to be remote or vice versa.

To configure your DOS environment so DOS interprets file handles for files in the shared filesystem to be local or remote, use the **merge** command with the following syntax:

merge set handle local | remote

The SCO Merge default is **handle remote**. To make DOS interpret files in the shared filesystem as local, type:

merge set handle local

The **merge set handle local** command causes DOS applications that use DOS interrupt 21H system call function 44H (IOCTL) code 0AH (Is Redirected Handle) to interpret files in the shared filesystem as local files. The **merge set handle remote** command causes DOS applications using the same system call to interpret files in the filesystem shared by DOS and the UNIX system as remote files. Use these **merge** commands when necessary to configure your DOS environment so applications that are sensitive to local and remote files work properly.

netbios option

The **netbios** option determines how SCO Merge responds when a DOS program asks whether NetBIOS is available. The syntax for this option is:

merge set netbios on | off

netbios on, the default, causes SCO Merge to answer yes when a DOS program asks whether NetBIOS support is available. Note that SCO Merge does not actually support NetBIOS; it only responds this way to the initial query from the application.

There are no programs known to be adversely affected by this configuration. However, if you find that a DOS program behaves in an unexpected way that you think may be attributable to this configuration, you may want to turn off netbios as follows:

merge set netbios off

5f02 option

The **5f02** option determines how SCO Merge treats DOS interrupt 21H system call function 5F02 (Get Redirector List Entry). The syntax is:

merge set 5f02 on | off

5f02 on, the default, causes SCO Merge to enable processing of function 5F02, which is used by the Microsoft redirector.

There are no programs known to be adversely affected by this configuration. However, if you find that a DOS program behaves in an unexpected way that you think may be attributable to this configuration, you may want to disable processing of function 5F02 as follows:

merge set 5f02 off

lpt_binary option

The **lpt_binary** option determines whether SCO Merge opens character devices in binary mode or in character mode. **lpt_binary on**, the default, causes SCO Merge to open DOS character devices in binary mode. This configuration allows you to embed the DOS end-of-file character <Ctrl>z in a file and treat it as an ordinary character. For example, if you print a file containing one or more <Ctrl>z characters, SCO Merge by default treats those characters just like any other characters. A conventional DOS computer would treat <Ctrl>z as the end of the file and halt printing at the first <Ctrl>z.

If you want SCO Merge to open character devices in character mode instead of binary mode, use the following **merge** command:

merge set lpt_binary off

This setting causes SCO Merge to treat character devices such as printers the same way conventional DOS does.

display option

The **merge display** command displays the current settings for all the **set** options. The syntax for this command is:

merge display [option]

For example, to see the current setting for **fastclk**, type at the DOS prompt:

merge display fastclk

[1] displays the current setting for **fastclk**. To see the current settings for all **merge** command options, type:

merge display

return option

The **merge return** option sets the DOS exit code to a numeric value (0 or 1) that corresponds to the state of the specified option. The syntax is:

merge return option

With this form of the **merge** command, the system assigns the value 1 to be "on" or "remote" and 0 to be "off" or "local." For example, if **fastclk** is "on" and you type:

merge return fastclk

the DOS exit code is set to the value "1". You can create the DOS batch file *errorlevel.bat* command to check the **merge return** exit code and then take appropriate action. You might include the following lines in the batch file to determine whether **fastclk** is on:

```
merge return fastclk
if errorlevel == 1 goto ISON
echo FASTCLK is OFF
goto END
:ISON
echo FASTCLK is ON
:END
```

Using batch files for merge commands

Instead of manually typing a **merge** command to set the value of **fastclk**, **pollsleep**, **drive**, or **handle** when you enter the DOS environment, you can include the appropriate command or commands in a batch file that is executed when you start DOS. In general, you should *not* include **merge** commands in your home directory *autoexec.bat* file unless you want to run **merge** commands every time you run DOS. Instead, follow these procedures:

1. Create a batch file, in any convenient directory, that includes the **merge** commands you want to run. For example, you could create a file in your home directory called *special.bat* that includes the commands:

```
merge set drive local
merge set pollsleep off
```

2. When you want to run a DOS application that requires these **merge** commands, start the DOS environment with the command:

```
dos +pfile
```

where *file* is the name of the batch file. For example:

```
dos +p/usr/wayne/special.bat
```

The **merge** commands in *special.bat* are then effective for the duration of your DOS environment (or until you change them by issuing another **merge** command).

You can also tailor applications to run specific batch files automatically whenever you start them from the UNIX shell. Refer to Chapter 5 and to the description of the **±p** option in Chapter 6 for further information.

4. *Tuning applications with the merge command*

Chapter 5

Tailoring the operation of DOS

When SCO Merge is installed, it has the characteristics of DOS on a conventional personal computer with 1 Mbyte of memory. With this configuration, you can use DOS alone, use the SCO Merge system as a standard UNIX computer, or combine DOS and UNIX functions. Depending on the way you use SCO Merge and on the specific DOS commands and applications you use, you might prefer to configure DOS differently. Different DOS applications, for example, require different amounts of memory.

If you use applications requiring large amounts of memory, you can clear nearly all of DOS conventional (low) memory for their use.

This chapter and Chapter 6 (DOS options) tell you how to modify DOS characteristics to suit your needs. The SCO Merge system administrator can use the procedures described in these chapters to define systemwide default DOS characteristics. Individual SCO Merge users can use the same procedures to define their own preferred DOS characteristics.

This chapter uses simple examples to explain the concepts and procedures you need to understand in order to modify DOS characteristics. Refer to Chapter 6 for complete descriptions of all the DOS options that you can use to customize the operation of DOS. This chapter is organized as follows:

- **DOS defaults:** How DOS defaults affect DOS environments and DOS applications.
- **Using DOS options on the dos command line:** How to override DOS defaults temporarily by specifying options together with the `dos` command.
- **Installing DOS options to change defaults:** How the system administrator can change DOS defaults.

- **Installing DOS options to override system defaults:** How you can configure commands, applications, and the DOS environment to use your preferred DOS options automatically.
- **Using `dosadmin` to install and remove DOS options:** How to use the `dosadmin` menu system to install and remove DOS options.
- **Using `dosopt` to install and remove DOS options:** How to use the `dosopt` command to install and remove DOS options.
- **Allocating memory for DOS:** How to configure the DOS environment for the memory needs of individual programs.

DOS defaults

When running under SCO Merge, DOS has many characteristics that you can customize to suit your needs. Each characteristic has a default value that you can use or change. Each DOS option described in Chapter 6 controls a specific DOS characteristic. Some of the characteristics you can control are:

- Amount of memory allocated for DOS.
- Which *autoexec.bat* and *config.sys* files are interpreted when you start DOS.
- Which hardware devices DOS is allowed to access.
- The initial working drive when you start DOS.

Refer to Chapter 6 for more information on specific DOS characteristics, including default values. Unless you override these defaults, they apply to:

- Any DOS environment that any user starts with the `dos` command.
- All DOS commands and applications run from the UNIX shell.

Using DOS options on the `dos` command line

Whenever you want to change a default DOS characteristic for a single execution of the DOS environment or a DOS application, use command-line options. The syntax for specifying options on the `dos` command line is:

`dos options [command [arguments]]`

For example, to create a DOS environment for Windows 3.1 with 3 Mbytes of memory, type:

`dos +m3`

See "Allocating memory for DOS" at the end of this chapter, for a discussion of how this memory is used. Similarly, to run WordStar with 2 Mbytes of expanded memory to create a file called **report**, type:

dos +aems2 ws report

Options consist of a single alphabetic character (**m** in the previous examples) preceded by either a plus or minus sign (+ or -). The plus or minus sign generally turns the characteristic controlled by that option "on" or "off." The **+t** option, for example, means "turn on automatic translation of DOS switch characters and path separators," while **-t** means "turn it off."

Some options require or are allowed to take parameters. The number "2" in the command **dos +m2**, for example, is a parameter to the **+m** option that specifies the number of Mbytes of memory to allocate.

Options that do not take parameters can be grouped together following a single plus or minus:

dos +bc -tx compute

Options that can take parameters must be specified separately:

dos +e/usr/dave/config.sys +l/usr/dave

The order of options is immaterial. The following two examples are both valid and both have the same effect:

dos -e -s

dos -s -e

DOS options that you specify on the **dos** command line override DOS defaults and also override any DOS options that are installed in an application or options file.

Installing DOS options to change defaults

The system administrator can redefine DOS defaults by installing DOS options in the SCO Merge options file */etc/dosenv.def*. This file controls the default characteristics of DOS environments that users start with the **dos** command.

There are two ways the system administrator can change DOS defaults:

- Use the **dosadmin** menu system, which provides convenient ways to change default values for the DOS characteristics that most often need changing.
- Use the **dosopt** command, which is less convenient but more flexible than **dosadmin**.

Both **dosadmin** and **dosopt** are described more thoroughly later in this chapter.

Installing DOS options to override system defaults

You can override DOS defaults by installing DOS options in personal options files, DOS commands, and DOS applications. You should install DOS options to override system defaults whenever you want to override a default value more permanently than you can by using DOS options on the **dos** command line. Installed options remain effective until you change or override them. Command-line options are effective only for one execution of a DOS environment or application.

Users can install DOS options in their personal options file *dosenv.def*, which must be in their home directories. This personal options file does not exist until users create it. **dosadmin** creates personal options files automatically when required. Also see "Creating a home directory options file," later in this chapter.

Like the analogous file in the */etc* directory, the home directory *dosenv.def* controls the characteristics of DOS environments.

The values defined in this home directory file are effective only for the user who owns the file. Values defined in home directory files override values defined in the options files in */etc*. Home directory options files are fully specified. That is, values for every configurable DOS characteristic are defined.

You can also configure specific DOS applications (Lotus 1-2-3, for example) so they use your preferred values when you run them from the UNIX shell. When you install specific DOS options directly in executable DOS command and application files, the options override DOS defaults and options installed in options files. They can be superseded only by using DOS options on the **dos** command line. You do not need to fully specify all DOS characteristics when you install options in a command or application. Unspecified characteristics take their values from the home directory *dosenv.def*, if it exists, or from */etc/dosenv.def* if it does not.

Procedures for installing options in personal options files or in applications are generally the same as the procedures for changing defaults in */etc/dosenv.def*. You can use the **dosadmin** menu system or the **dosopt** command as described in the following sections.

Using *dosadmin* to install and remove DOS options

The **dosadmin** menu system provides an easy way to change the values of three important DOS characteristics: memory, DOS startup files, and DOS device files. The following paragraphs tell you more about these characteristics and how to modify them using **dosadmin**.

Characteristics you can tailor using *dosadmin*

You can use **dosadmin** to install DOS options that affect memory, DOS startup files, and DOS device files. Depending on how you use **dosadmin**, you can tailor these characteristics for:

- The DOS environment and all DOS commands and applications run from the UNIX shell.
- Specific DOS commands and applications.

Startup memory

The SCO Merge factory default for DOS memory is 1 Mbyte.

This total memory must be shared, however, by DOS, the application, all the device drivers in use, and any terminate-and-stay-resident (TSR) programs you have loaded. SCO Merge uses about 50K bytes for its own driver. For further information, see “Allocating memory for DOS,” later in this chapter.

SCO Merge runs more efficiently if you allocate only as much memory as you need. For example, if you normally need only 1 Mbyte of memory while in the DOS environment, you can install the 1 Mbyte value as your preferred memory value.

DOS startup file

On a conventional DOS computer, if you have a file called *autoexec.bat* on your system disk, the commands in that file are executed automatically whenever you boot the computer. When you use SCO Merge, you do not boot DOS the same way you would on a stand-alone personal computer. SCO Merge, however, can execute an *autoexec.bat* file when you use the DOS environment or run a DOS program, just as though you had booted DOS on a conventional personal computer.

The SCO Merge factory default is to execute two *autoexec.bat* files in succession when you start DOS by typing **dos** or when you run DOS programs from the UNIX shell. These two files are the *root* directory *autoexec.bat* and your home directory *autoexec.bat*.

Using **dosadmin**, you can specify a single DOS batch file that is to be executed instead of the default files when you start the DOS environment or a DOS program, or specify that *no* batch file is to be executed. The file you specify can have any legal DOS or UNIX filename, and you can specify it either by full UNIX path name, for example:

```
/usr/joe/dbin/appl.bat
```

or by DOS drive and full DOS path name, for example:

```
c:\usr\joe\dbin\appl.bat
```

DOS device file

On a conventional DOS computer, the file `\config.sys` is interpreted automatically when you boot DOS. SCO Merge by default treats `config.sys` files similarly to `autoexec.bat` files: it interprets the *root* directory `config.sys` file and your home directory `config.sys` file in succession whenever you run the DOS environment or a DOS application.

Using **dosadmin**, you can specify *one or more* configuration files that are to be interpreted instead of the defaults, or you can specify that *no* configuration file is to be interpreted. The files you specify can have any legal UNIX filename. You can specify files either by full UNIX path, for example:

```
/usr/joe/dbin/config.sys
```

or by DOS drive and full DOS path name, for example:

```
c:\usr\joe\dbin\config.sys
```

To specify multiple files, list them in the order they are to be interpreted, separated by a space, a comma, or both. For example:

```
c:\config.sys, d:\special.sys
```

Using dosadmin to tailor DOS

NOTE When you use **dosadmin** to tailor DOS, SCO Merge maintains a record of your configuration in the file `dosenv.def`. This file is stored in your home directory if you are a normal system user or in `/etc` if you are the system administrator logged in as root. You do not need to be concerned with the contents of this file, but be sure you do not accidentally delete it.

To use **dosadmin** to tailor DOS, follow these procedures:

- Select the appropriate Configuration menu.
- Identify the DOS application you are tailoring, if you are tailoring a specific application.
- Fill in the fields for memory, startup file, and device file.

The following paragraphs describe these steps in more detail.

Selecting the configuration menu

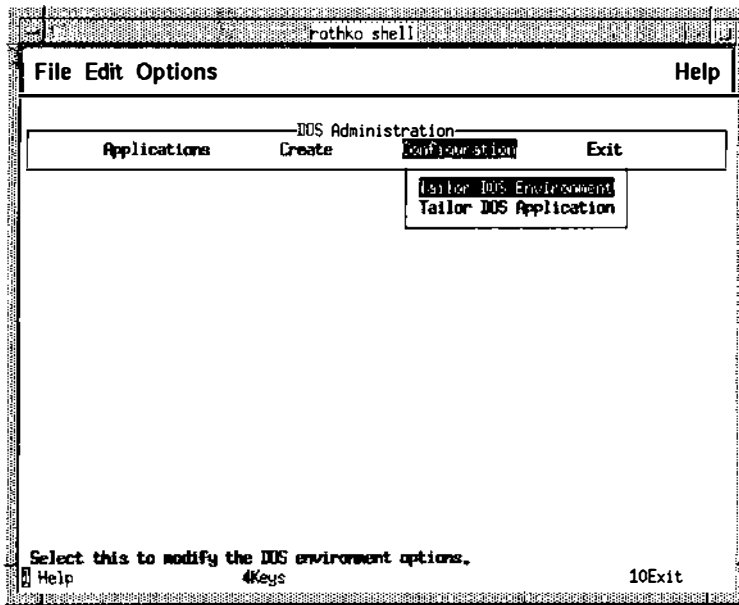
Follow these steps to select the appropriate Configuration menu:

1. Type:

dosadmin

(You can type the **dosadmin** command either at the UNIX shell prompt or in the DOS environment.) The main DOS Administration menu is displayed.

2. Press the left or right arrow key (← or →) to move the highlighted field at the top of the screen to "Configuration." Your display looks like this:



The Configuration menu offers two selections:

- Tailor DOS Environment
- Tailor DOS Application

Use the up or down arrow key (↑ or ↓) to highlight the appropriate field and press (Enter). Follow these guidelines to make your selection:

- **Tailor DOS Environment:** Use this selection when you want to modify the characteristics of the DOS environment (that is, the DOS characteristics that apply when you type **dos** and the characteristics of DOS commands and applications when they are run from the UNIX shell). If you are logged in as root, the characteristics you specify *replace* existing DOS

defaults for all users. If you are not logged in as root, the characteristics you specify *override* existing DOS defaults.

You can override characteristics specified with this menu by using DOS command-line options when you use the `dos` command as described earlier in this chapter.

- **Tailor DOS Application:** Use this selection when you want to modify the characteristics of a specific DOS command or application when it is run from the UNIX shell. If you are logged in as root, you can tailor any command or application, although normally you use this menu as root in order to tailor publicly accessible applications in directories such as `/usr/ldbin`. You can tailor only commands and applications for which you have write permission.

If you have selected the Tailor DOS Application menu, you must identify the command or application you want to tailor as described in the next section. If you have selected the Tailor DOS Environment menu, skip ahead to “Filling in the configuration menu.”

Selecting the DOS command or application

If you have selected the Tailor DOS Application menu, your display resembles this:

rothko shell

File Edit Options Help

DOS Administration

Tailor DOS Applications

DOS Application File Name:
[REDACTED]

Startup Memory:
DOS Startup File:
DOS Device File:

< Cancel > << Change >>

Enter the DOS executable file name.
1Help 2Choices 3Restore 4Keys 5Clear 6Truncate 7Left 8Right 9Cancel 10Execute

You must now fill in the “DOS application file name” field to identify the command or application you want to tailor.

Type in the name of a DOS executable file. You can name the file by full UNIX path or by DOS drive and full DOS path, for example:

```
/usr/ldbin/lotus.com
```

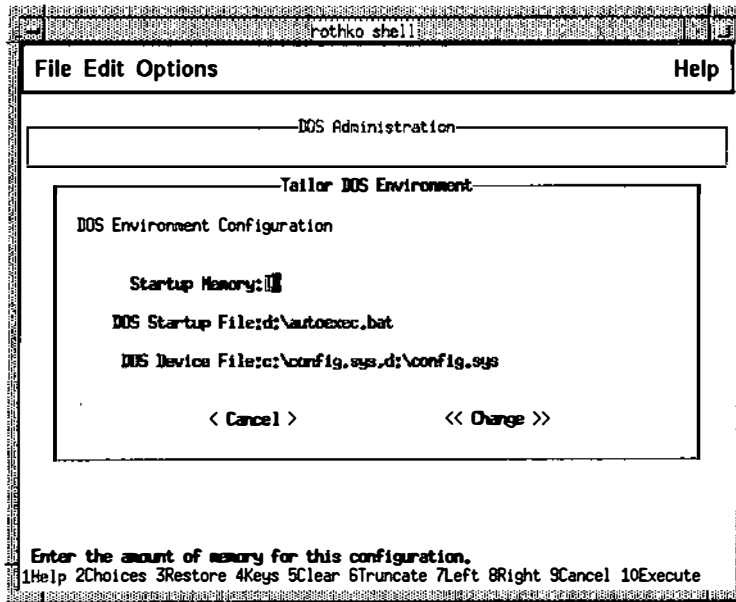
or

```
c:\usr\ldbin\lotus.com
```

Continue with the steps described in the next section.

Filling in the configuration menu

The Tailor DOS Environment and Tailor DOS Application menus are similar. They both have fields named "Startup Memory," "DOS Startup File," and "DOS Device File." The display resembles this:



The values shown for "Startup Memory," "DOS Startup File," and "DOS Device File" are the current values that apply to the DOS environment, application, or applications you are configuring. The actual values shown on your display may differ from those in this illustration.

To change any of the displayed values, press the (Tab) key until the field you want to change is highlighted. Then type in the value you want. If you make a mistake, press the (Bksp) key (←) and retype your entry or press the (F5) key to clear the field. Then retype your entry.

When the “DOS Startup File” or “DOS Device File” field is highlighted, you can use the <F5> key to clear the field, which results in no startup file or device file being interpreted when you start the DOS environment.

Follow these guidelines as you fill in each field:

- **Startup Memory:** The value displayed here is the amount of memory (in Mbytes) reserved for this application when you run it from the UNIX system shell. Note that because SCO Merge takes up some space, there is actually somewhat less memory available than the displayed value. “Allocating memory for DOS,” at the end of this chapter, discusses this issue in further detail.

Not all applications need the full amount of memory. Check the application manufacturer’s instructions for the recommended amount of memory.

WARNING Specifying too little memory may cause your DOS environment or application to fail.

- **DOS Startup File:** If this field is filled in *only* with *d:\autoexec.bat*, it means *both* the *root* directory *autoexec.bat* and personal, home-directory *autoexec.bat* files are executed.¹ Note that the home directory *autoexec.bat* is executed second, so settings in that file override settings in the *root* directory *autoexec.bat*. If you want only one *autoexec.bat* file to be interpreted, identify it using DOS drive C: or a UNIX-style path name. For example, filling in this field with *c:\usr\greg\autoexec.bat* causes only Greg’s home-directory *autoexec.bat* to be executed. Filling in *\autoexec.bat* causes only the *root* directory *autoexec.bat* to be executed.

You are not restricted to files named *autoexec.bat* for this field. You can identify any single DOS file, in any directory in the filesystem shared by DOS and the UNIX system, provided that it contains legal DOS batch commands.

For example, if you want your application to run the file *C:\usr\pat\lotus\appl.bat* every time you invoke it, type the following in this field:

C:\usr\pat\lotus\appl.bat

If you do not want any file to run automatically when you use this application, erase the field by pressing the <F5> key.

- **DOS Device File:** This field identifies the device configuration files that you want DOS to interpret when you run your application from the UNIX

1. If you are logged in as root when you use DOS, however, only one *autoexec.bat* file is executed — the one in the *root* directory.

shell. It normally shows both *c:\config.sys* and *d:\config.sys*. When these two filenames are displayed, DOS interprets both the *root* directory *config.sys* and a home-directory *config.sys*. If you are logged in as root when you use DOS, however, only one *config.sys* file is executed—the one in the *root* directory. (Note that home directory *config.sys* files do not exist until users create them.)

You can change or delete the files displayed in this field. For most purposes, however, it is preferable to let DOS interpret the default *config.sys* files.

You are not restricted to files named *config.sys* for this field. You can identify one or more DOS files, in any directories in the shared filesystem, that contain legal DOS *config.sys* commands.

When you have filled in the configuration menu, check your entries for accuracy. If you choose to cancel the tailoring operation, press the <Tab> key until the "<Cancel>" field is highlighted and press <Enter>.

When you have entered the values you want, press the <Tab> key until the "<<Change>>" field is highlighted and press <Enter> to save the new values you have specified.

The top-level DOS Administration menu reappears. Press the left or right arrow key until "Exit" is highlighted, and press <Enter>. Your UNIX or DOS system prompt returns.

Using dosopt to install and remove DOS options

You use the **dosopt** command (like you use the **dosadmin** menu system) to assign DOS characteristics such as memory, startup file, and device configuration file to applications or to your environment. The **dosopt** command is more flexible than **dosadmin**. You can use it to assign many characteristics besides memory, startup file, and configuration file. You can also use **dosopt** where you cannot use **dosadmin**—inside UNIX shell scripts, for example.

As with **dosadmin**, you can use the **dosopt** command to install DOS options that determine:

- Characteristics of DOS environments and applications run from the UNIX shell for all users. Characteristics of specific DOS applications run from the UNIX shell.

The syntax for **dosopt** is:

dosopt options options_file

or

dosopt options application

As described earlier, under "Using DOS options on the dos command line," *options* are key letters preceded by a plus or minus (+ or -) sign that identify the values of specific DOS characteristics. Chapter 6 describes each DOS option thoroughly.

options files are files that hold records of DOS options you have installed. In the command form **dosopt options application**, *application* is the name of a specific DOS application—"123," for example—in which you want to install DOS options.

The names of the options files and the DOS processes they affect are:

- */etc/dosenv.def*: This file defines system default characteristics for DOS environments and DOS applications started from the UNIX prompt by all users. DOS options defined in this file can be overridden by using DOS options on the **dos** command line or by installing DOS options in home directory *dosenv.def* files.
- *\$HOME/dosenv.def*: A user's home directory *dosenv.def* file overrides *all* values defined in */etc/dosenv.def* and defines characteristics of DOS environments and DOS applications started from the UNIX prompt by that user. DOS options defined in *\$HOME/dosenv.def* can be overridden by using **dos** command-line options.
- **A specific DOS application**: Although not an options file in the same sense as the files listed above, any DOS executable file can hold a record of DOS options that apply to that file when you run it from the UNIX shell. Options installed in DOS commands and applications override */etc/dosenv.def* and *\$HOME/dosenv.def*. You can override options installed in commands and applications by using **dos** command-line options.

Only the system administrator, logged in as root, can change options in */etc/dosenv.def*, and only with **dosopt**. The file cannot be edited directly with **edlin**, **vi** or any ordinary text editor.

All users can create *dosenv.def* files in their home directories and install options in them. Anyone who has write permission for a command or application can install options in the executable file.

Creating a home directory options file

The system default options file */etc/dosenv.def* is created automatically when you install SCO Merge. Home directory *dosenv.def* files, however, do not exist until you create them. There are two ways to create this home directory file:

- Use the **dosadmin** menu system as described earlier in this chapter to modify the characteristics of the DOS environment or DOS applications. If you are logged in as a user who has a home directory, **dosadmin** automatically creates *\$HOME/dosenv.def* the first time you use the "Tailor DOS environment" menu to tailor the DOS environment.
- Copy */etc/dosenv.def* to your home directory and make sure it is writable. If you are working in your home directory, you can use the following commands:

```
cp /etc/dosenv.def dosenv.def
chmod u+w dosenv.def
```

Installing DOS options

Following are a few examples of **dosopt** commands to illustrate how you install options in options files and in applications. For more information on the meaning of specific options, refer to Chapter 6. All of the examples which follow assume that you already have a *\$HOME/dosenv.def* file, created by one of the methods above.

If you are working in your home directory, the command:

```
dosopt +m2 dosenv.def
```

overrides the default values for memory (**+m**) so that you get 2 Mbytes of memory. If the system administrator enters the following similar command:

```
dosopt +m2 /etc/dosenv.def
```

these values become the defaults for all users who run DOS environments.

Preceding a DOS option with a minus sign (**-**) generally turns off that function. For example, to prevent *autoexec.bat* from running, type:

```
dosopt -p dosenv.def
```

You can change values at different times without affecting values you specified previously:

```
dosopt -p dosenv.def  
dosopt +t dosenv.def
```

In this example, *autoexec.bat* is disabled and automatic path separator translation is turned on in two separate commands. The *autoexec.bat* file remains disabled unless it is later explicitly enabled with the **+p** option. If you are working in your home directory, the command:

```
dosopt -s dosenv.def
```

changes the default to disable the DOS printer timeout.

The command:

```
dosopt +b /usr/vicky/dbin/cl
```

identifies the *cl* program as stream-oriented.

Note that DOS applications that you can run from the UNIX shell normally have two names: a DOS filename with an extension (*.exe*, *.com*, or *.bat*) and a UNIX name without an extension, which is linked to the DOS name. For example, the file *cl.exe* would be linked to *cl*. When you use **dosopt** to install an option in an application, you can use either name. That is, the following two commands have the same effect since *cl.exe* and *cl* are two names for the same file:

```
dosopt +b cl  
dosopt +b cl.exe
```

Displaying installed options

You can display the **dos** options currently in effect by typing:

```
dosopt +v
```

This command displays the contents of your home directory *dosenv.def* file, if you have one. Otherwise it displays the contents of */etc/dosenv.def*. The display is similar to the following:

```
File: /usr/vicky/dosenv.def  
+ae:=/dev/dsk/Os1 +amouse -b +dc +e +l +m640 +p +s15 +t.
```

To display the options installed in a specific options file, use the command:

```
dosopt +v file
```

For example, to display the current system default DOS options, issue the command:

```
dosopt +v /etc/dosenv.def
```

Removing installed options

To remove an option installed in a command or application, use the **dosopt +z** option. For example, if you have assigned 2 Mbytes of memory to Windows 3.1 using the command:

```
dosopt +m2 win.com
```

you can use the **+z** option to remove the installed memory value and allow the memory value from **\$HOME/dosenv.def** (if it exists) or **/etc/dosenv.def** to be effective. The command is:

```
dosopt +zm wincom
```

When you use **+z** to remove previously installed **+a** ("attach device") options, you can remove individual **+adevice** options or you can remove *all* **+a** options with a single **dosopt** command. For example, assume you have used the following **dosopt** commands to specify attachment of both com1 and com2 for the application named APPL:

```
dosopt +acom1 appl
```

```
dosopt +acom2 appl
```

To remove the **+acom2** instruction but keep the **+acom1** instruction, type:

```
dosopt +zacom2 appl
```

To remove all **+a** options from the file APPL at once, type:

```
dosopt +za appl
```

Refer to Chapter 6 for further information on the **±a** option.

To remove *all* installed options in a DOS command or application, use an uppercase **"Z"**:

```
dosopt +Z appl
```

This command removes all installed options from **appl** and allows the option values in **\$HOME/dosenv.def** (if it exists) or **/etc/dosenv.def** to be effective.

The **+z** and **+Z** options cannot be used on the options files **\$HOME/dosenv.def** or **/etc/dosenv.def**.

Restoring DOS options files

If for any reason you want to restore the original contents in your system default DOS options file (**/etc/dosenv.def**), you can copy the corresponding file from the directory **/usr/merge**. **/usr/merge/dosenv.def** is a permanent copy of **/etc/dosenv.def** in the form that exists when you install SCO Merge.

Allocating memory for DOS

When you start DOS or any DOS application, SCO Merge allocates 1 Mbyte as the default amount of RAM for DOS use. This is known as “base memory” or “conventional DOS memory.” Base memory is the only area most DOS programs can use for both the program code and the program’s data. Unless special care is taken to conserve this scarce resource, much of this base memory can be absorbed in “overhead.”

DOS 5.0 offers considerably better tools for managing memory than previous versions of DOS, and SCO Merge uses those tools to optimize the default amount of memory left for your DOS programs. The factory default DOS images enable the High Memory Area and the Upper Memory Blocks for storage of DOS, device drivers and TSRs. DOS itself is loaded into the HMA.

The SCO Merge Bridge driver is the standard redirector that permits DOS programs to access the filesystem shared by DOS and the UNIX system. This driver absorbs about 50K bytes and it too is loaded into the Upper Memory Blocks.

The amount of Conventional Memory available for your DOS applications varies depending on how much additional memory is consumed by the resident portions of DOS and *command.com* and by the other drivers and TSRs you load. An average arrangement, consisting of the factory default images with a mouse driver added, leaves approximately 610K of working RAM for your programs. Your own total will vary according to what you load and the portion of memory to which you allocate it. We advise you to read carefully those sections of your DOS documentation that deal with memory management and to use the tools DOS provides, especially the **mem** command. Load device drivers and TSRs high whenever possible.

Running RAM-hungry applications in base memory

Applications that cannot access Extended or Expanded memory and that need a lot of low DOS memory need special handling. We advise that, in addition to loading all possible drivers and TSRs into high memory, you configure your DOS environment to start such programs with only the absolute essentials. A minimal DOS environment includes only the drivers, environment variables, TSRs, and other memory-consuming resources that are essential to run a specific application. All drivers and TSRs that are not critical to the operation of the program are omitted.

To run an application in a minimal DOS environment, you create alternative *autoexec.bat* and *config.sys* startup files that are customized to meet the minimal requirements of that application. Then you can use any of the following procedures:

- Configure the application so it always uses the alternative startup files whenever users run it from the UNIX shell.
- Start the application with the **dos** command from the UNIX command line, using the **dos +e** and **+p** options to specify your customized startup files.
- Start a DOS environment using the **+p** and **+e** options to specify your customized startup files. Then start the application from the DOS prompt.

The following sections describe these procedures.

Preparing minimal startup files

To prepare minimal startup files for a specific application, simply create an *autoexec.bat* file and a *config.sys* file that include only the startup commands required by that application. Do not modify your personal home directory *autoexec.bat* or *config.sys* files or the system default *autoexec.bat* and *config.sys* files in the *root* directory. Instead, create separate startup files for your RAM-hungry application. These files do not have to be named *autoexec.bat* and *config.sys*. They can have any legal UNIX filename, and they can be stored in any accessible UNIX directory. For simplicity, we recommend that you name them *autoexec.bat* and *config.sys* and create them in the same directory containing your application.

For example, if your application is named **ramhog** and is in the directory */usr/pat/dbin/ramhog*, create both the *autoexec.bat* and *config.sys* files in */usr/pat/dbin/ramhog*. If **ramhog** is a public application that resides, for example, in the directory */usr/ldbin/ramhog*, the system administrator can create the required startup files in */usr/ldbin/ramhog*.

You can create the startup files in a variety of ways. You can copy your personal startup files or the system default startup files and edit out unnecessary commands; or you can start with new, empty files and manually add only the required commands. If your application does not require either *autoexec.bat* or *config.sys*, you need not create the unnecessary file.

Using minimal startup files

You use the **dos +e** and **+p** options to specify the customized startup files you want to use with your application. Specify the full path of the *config.sys* file with the **+e** option, and specify the full path of the *autoexec.bat* file with the **+p** option.

For example, assume again that your application is named **ramhog** and your customized startup files are in the directory */usr/pat/dbin/ramhog*. One way to run **ramhog** is with the command:

```
dos +e/usr/pat/dbin/config.sys +p/usr/pat/dbin/autoexec.bat ramhog
```

If either of the startup files is not required, use the **-e** or **-p** option as appropriate. For example, if *autoexec.bat* is not used, type:

```
dos +e/usr/pat/dbin/config.sys -p ramhog
```

Instead of using the **+e** and **+p** options on the command line, you can install them permanently in the application using the procedures explained in "Installing DOS options," earlier in this chapter. The following command causes **ramhog** to interpret */usr/pat/dbin/autoexec.bat* whenever **ramhog** is run from the UNIX shell:

```
dosopt +p/usr/pat/dbin/autoexec.bat ramhog.exe
```

Refer to Chapter 6 for more information on the **+e** and **+p** options.

Allocating memory for Windows

SCO Merge supports two versions of Windows, and they use memory in different ways. To make additional memory available to Windows 3.0, you must add Expanded Memory. To make additional memory available to Windows 3.1, you should add Extended Memory.

Memory for Windows 3.0

Windows 3.0 runs in Real mode, as it would on an 8086 processor. Windows 3.0 can use Base memory plus Expanded Memory. Because EMS memory was originally developed as an add-in adapter, it is treated as if it were a separate device. To allocate memory beyond the 1 Mbyte limit to Windows 3.0, you "attach" that memory to your DOS process.

From the command line, you use the **+a** option of the **dos** command. The following command requests 4 Mbytes of EMS memory and starts a Windows session.

```
dos +aems4 win
```

To achieve the same purpose from X, use **xdosopt** to bring up the DOS options dialog box. For more information on how to use **xdosopt**, see *Using SCO Merge*.

In the Memory box, click on EMS. A drop-down list shows the increments of EMS memory you may add. Click one of the selections. Then click the Start button.

Your Windows session begins with the requested amount of memory.

Memory for Windows 3.1

Windows 3.1 runs in Standard or "Protected" Mode on an emulated 80286 processor. Windows 3.1 can use Base Memory, plus Extended Memory. (It *can* use EMS memory as well, but Extended Memory is much more efficient.) Additional memory beyond the 1 Mbyte limit should always be allocated in the form of XMS memory.

When you allocate memory to Windows 3.1, Windows dispenses that memory to its applications according to their demands and their ability to use memory. You do not need to be concerned with what kind of memory is given to which application. For this reason, SCO Merge refers to the memory given to Windows 3.1 as "Standard Memory." That means an aggregate of 1 Mbyte of "conventional" DOS RAM, plus as many Mbytes of Extended Memory as you care to add. The upper limit is 15 Mbytes, total.

Standard Memory is not attached as a hardware device. It is simply allocated as "memory," using the **+m** (add memory) option of the **dos** command.

From the command line, the following command requests 4 Mbytes of Standard memory and starts a Windows session.

\$ dos +m4 win

To achieve the same purpose from X, use **xdosopt** to bring up the DOS options dialog box.

In the Memory box, click on Standard. A drop-down list shows the 1 Mbyte increments of Standard memory you may add. Click one of the selections. Then click the Start button.

Your Windows session begins with the requested amount of memory.

Note that Extended memory can be used *only* by Windows 3.1 and the applications that run under it. Special modifications have been made to SCO Merge to accommodate Windows 3.1. Other DOS programs cannot access memory in this way, even if they can use Extended Memory under "raw" DOS.

Chapter 6

DOS options

This chapter explains the options you can use to modify the operation of DOS. To use these options effectively, you should be familiar with the procedures for using the **dosopt** and **dos** commands described in Chapter 5. You can use the options described in this chapter with the **dosopt** command to install options in options files (*/etc/dosenv.def* and *\$\$HOME/dosenv.def*) and in DOS applications. You can use the same options on the **dos** command line when you do not want to install permanent values—that is, when you want the options to be in effect only for the remainder of the DOS session.

This chapter contains a table listing the options and summarizing their use.

Following the table are complete descriptions of each option, arranged alphabetically by option name. Each description includes a “Usage” note, showing the possible ways of using that option with **dos** or **dosopt**. Type all boldface letters and words exactly as shown. The symbol $\pm$ means you can use the option with either a + or a - (usually to turn the indicated value on or off). Bolded and italicized words (like *file*) are generic terms for which you should substitute actual names (such as the actual name of a file on your computer). Brackets surrounding data (such as [*file*]) mean that supplying the bracketed data is optional. A vertical bar (|) means “either or.” Choose one of the separated items and type it as part of the command. For example, **dosopt +v[options_file | program]** means you may name either an options file or a DOS program when you use the +v option.

Unless otherwise indicated, you can use any option in the following ways:

- With the **dosopt** command to install your preferred values in either of the following options files:
 - */etc/dosenv.def*
 - *\$HOME/dosenv.def*
- With the **dosopt** command to install your preferred values in a DOS application.
- On the **dos** command line.

For example, the usage line:

+d[drive]

means any of the following commands are allowed:

```
dosopt +dd /etc/dosenv.def  
dosopt +dd $HOME/dosenv.def  
dosopt +dd cl  
dos +dd  
dos +dd cl temp.c
```

You can substitute any legal DOS drive letter for the second “d” in **+dd** in each of these examples. You can also omit the drive letter to indicate the default (drive C:).

Each option description also includes a “Factory default” note that shows the original system default value (defined in */etc/dosenv.def*) when SCO Merge is installed.

Option	Meaning
+ados_device	Attach specified DOS device.
-ados_device	Cancel DOS device attachment.
+b	DOS program is stream-oriented.
-b	DOS program is display-oriented.
+c	Pass command unchanged to <i>command.com</i> ¹
+d[drive]	Set initial current drive.
+e[file[,file ...]]	Interpret one or more <i>files</i> instead of default configuration files. If <i>files</i> are not specified, default <i>/config.sys</i> and <i>\$HOME/config.sys</i> are used.
-e	Do not interpret any <i>config.sys</i> when DOS is run.
+h	Display help text.
+l[file directory]	Load DOS image <i>file</i> or an image file from <i>directory</i> instead of default image.
-l	Do not use a DOS image. Boot from drive A: instead.
+mn	Allocate <i>n</i> Mbytes of Standard Memory (Base plus Extended).
+p[file]	Run <i>/autoexec.bat</i> and <i>\$HOME/autoexec.bat</i> . <i>file</i> , if specified, is the alternative batch file to be run instead of the defaults.
-p	Do not run any <i>autoexec.bat</i> .
+s[n]	Print accumulated DOS printer output when no characters have been sent to the printer for <i>n</i> seconds. <i>n</i> is a timeout value between 5 and 3600 seconds (default 15 seconds).
-s	Do not print DOS printer output until DOS process exits.
+t	Translate DOS switch characters and path separators in standard way.
-t	Do not translate DOS switch characters and path separators.
+v	"Verbose" mode: display dosopt acknowledgement message. ²
+y	Null option for DOS .com files that have no other options. ²
+x	Forces DOS to start up as an X client, even if XMERGE is not set.
+z	Remove an option from an application. ²
+Z	Remove all options from an application. ²

1. The **+c** command can be used only on the **dos** command line. It cannot be used with **dosopt**.

2. The **+v**, **+y**, **+z**, and **+Z** options can be used only with **dosopt**. They cannot be used on the **dos** command line.

Attach devices to DOS

Usage: `±adevice`
`-a[device]`

Factory default: Attach default devices automatically

There are several ways you can configure the hardware devices that DOS processes can access. SCO Merge automatically manages access to the most important hardware devices, such as the keyboard, the display, and diskette drives. You can use the `±a` option to alter the default SCO Merge treatment of some of these devices. There are also several devices that are not available automatically when you use DOS because they are shared with the UNIX system and other DOS processes. You must explicitly request access to these devices using the `+a` option.

Default devices

SCO Merge automatically makes the following devices available to DOS processes:

- Console and terminal keyboards
- The fixed disk
- Diskette drives A: and B:
- A mouse
- The primary physical DOS partition
- Console and terminal displays

The keyboard

Communication between DOS processes and console and terminal keyboards is transparent and automatic.

The fixed disk

The fixed disk filesystem shared by DOS and the UNIX system, including DOS drives C:, D:, and J:, is also available automatically to all DOS processes.

Diskette drives

Diskette drives A: and B: are available to DOS as long as they are not mounted as UNIX devices or otherwise used by the UNIX system (by a `cpio` process, for example). The diskette drives are assigned on a first-come-first-served basis to any DOS process that attempts to access them whenever they have been unused for five seconds or longer. Drives A: and B: are assigned and released simultaneously.

The mouse

Any mouse that is properly installed and accessible to the UNIX system is also accessible to DOS (provided it is not actively being used by another process at the moment DOS tries to use it). DOS processes always view the mouse as a Microsoft bus mouse. There is a device driver entry in the default SCO Merge */config.sys* file to support DOS communication with a mouse. When you use the MultiScreen™ feature on the console, the mouse automatically communicates with the screen currently displayed.

Physical DOS partitions

The primary physical DOS partition, if your computer has one, is automatically available as drive E: to all DOS processes. If your computer has an extended DOS partition, it must be explicitly attached. For more information, see “Attaching DOS drives,” later in this chapter, and “Administering physical DOS partitions” in Chapter 2.

Display adapters

SCO Merge supports standard serial terminals, PC scancode serial terminals, and consoles using standard monochrome (MDA), CGA, VGA, or Hercules display adapters. In normal use, SCO Merge automatically sets up your DOS environment to accommodate the type of console or terminal you are using when you run DOS.

You can explicitly specify monochrome, CGA, VGA, or Hercules capabilities using the **+a** option, although there is typically no need to do so. Your display must have at least the capabilities reflected by the option you select—for example, you cannot specify a CGA adapter type if you are using a serial terminal or a console that has an MDA adapter. The following table summarizes the display adapters you can specify on each type of terminal and the resulting capabilities:

Capability by physical display type				
Option	Monochrome or Serial Terminal	Hercules	CGA	VGA
None	mono	herc	CGA	VGA
+amono	mono	mono	mono	mono
+aherc	-	herc	-	-
+acga	-	-	CGA	CGA
+avga	-	-	-	VGA

If you use the MultiScreen feature with either a VGA display and switch away from a screen displaying a high-resolution graphics application, the application is suspended. Applications running in monochrome, CGA, or Hercules mode continue to run even when their screens are not visible.

Other standard DOS devices

The devices described in this section must be explicitly requested using the **+a** option. The specific **+a** syntax that applies to each device is shown together with the device description.

Except for expanded memory (EMS), these devices are available to only one process at a time. If DOS attempts to access a device that is currently in use by another DOS or UNIX process, SCO Merge displays an error message and aborts the DOS process.

com ports

Usage: **+acom1**[=*unix_device*]
+acom2[=*unix_device*]
+adcom1[=*dcom2*]
+adcom2[=*dcom1*]

You can attach com ports to DOS processes either directly (**+adcom1** or **+adcom2**) or indirectly (**+acom1** or **+acom2**). When you use direct attachment, DOS controls the com hardware directly. When you use indirect attachment, SCO Merge mediates the communication between DOS and the com hardware. Refer to *Using SCO Merge* for further information on the advantages and disadvantages of both forms of attachment.

unix_device is the name of any terminal line in the form */dev/ttyxx*, where *ttyxx* is a specific name such as *tty1a* or *tty20*. When you attach a com port to a DOS process, the specified terminal line is accessible to DOS as com1 or com2. By default, **+acom1** means **+acom1=/dev/tty1a** and **+acom2** means **+acom2=/dev/tty2a**. If your computer has a compatible³ multiport serial card, you can attach other terminal lines as com1 or com2 by using options such as **+acom2=/dev/tty20**.

Use the option form **+adcomn=dcomn** when you want a DOS application to view one com port as though it is a different port. For example, the option **+adcom1=dcom2** causes DOS to treat the physical com2 port as though it is com1.

3. Not all cards are supported. "Smart" serial cards are those which have their own CPU. They usually offer buffering and caching. "Smart" cards are supported only if their drivers have been specially constructed to support SCO Merge. Consult your "Smart" card documentation for more information.

Expanded Memory (EMS)

Usage: `+aems[n]`

EMS memory was originally available only on an adapter board which was installed in the computer. Even though SCO Merge emulates expanded memory in a completely different fashion, it is still “attached” to your DOS session as if it were a separate device. For a discussion of how DOS programs can access EMS memory, see “Allocating memory for DOS” in Chapter 5. SCO Merge simulates DOS expanded memory as specified by Lotus, Intel, and Microsoft.⁴ `+aems` requests the default of one Mbyte of expanded memory. To request a different amount, use `+aemsn`, where *n* is 512, for 512K bytes of expanded memory, or 1, 2, 3, 4, 5, 6, or 8, specifying Mbytes.

There is a device driver entry for EMS in the system default `/config.sys` file, so EMS is available only if DOS interprets this file or another `config.sys` file that has a similar entry.

Each DOS process can request its own private expanded memory, so multiple DOS processes and multiple users can use EMS simultaneously.

Refer to Chapter 8 for technical information on the device driver SCO Merge uses for expanded memory.

Printer ports

Usage: `+alpn`

By default, SCO Merge sends all DOS printer output to the UNIX print spooler, so DOS does not require access to printer ports. If you want DOS to control a printer port directly, attach the port using the `+alpn` option, where *n* is 0, 1, or 2. Refer to “Administering DOS Printers” in Chapter 2 for further information.

The game port

Usage: `+agame`

To use the game port with DOS, use the `+agame` option.

4. *The Lotus/Intel/Microsoft Expanded Memory Specification*, Version 4.0, Lotus Development Corporation, Intel Corporation, and Microsoft Corporation.

Attaching DOS drives

Usage: `+adrive_letter:=unix_file_name[e,i,r]`

As mentioned earlier, the filesystem shared by DOS and the UNIX system, including DOS drives C:, D:, and J:, is always available to DOS, and SCO Merge automatically manages access to diskette drives.

The primary physical DOS partition (if your computer has one) is accessible as drive E:. The `+a` option `+ae:=/dev/dsk/devicename`, which is included in the default `/etc/dosenv.def` file, is required to make the DOS partition accessible as drive E:.

The `+ae:=/dev/dsk/devicename` option is one example of a general option form that you can use to access physical DOS partitions, virtual DOS partitions, and virtual DOS floppies. To access any of these devices with DOS, you use the `+a` option and specify a DOS drive and a UNIX filename. The UNIX filename identifies the file that contains the DOS partition or virtual floppy. Follow these guidelines when you select a DOS drive letter:

- Use drive letter **a** or **b** to access a virtual floppy.
- If you attach a virtual floppy using either **a** or **b** as the drive letter, you cannot simultaneously access a physical diskette drive with the same name. You can, however, access physical and virtual floppies with different names at the same time.
- Use drive letters **e**, **f**, **g**, **h** and **i** to access the primary physical DOS partition, logical drives within the extended DOS partition, and virtual DOS partitions. For a list of the devices currently attached, look at `/etc/dosdev`.
- The primary physical DOS partition (`/dev/dsk/devicename`) is accessible as drive E: by default. You can use drive letter **e** to access a virtual partition or a logical drive within the extended partition, but if you do, the primary physical partition is not simultaneously accessible as drive E:. For details, see "Using physical DOS partitions" in *Using SCO Merge*. You can also use `+a` with a drive letter other than **e** (that is, **f**, **g**, **h**, or **i**) to access `/dev/dsk/devicename` if you choose.

Virtual partitions and floppies have the same limitations as physical DOS partitions. Multiple DOS processes can read them at the same time, but only one process can write to each of these devices at a time. When one process starts writing, other processes are prevented from both reading and writing to that device until the DOS session that owns the original process is terminated.

Specifying immediate, exclusive, or read-only access

You request access to a DOS drive at the start of your DOS session using the `+adrive_letter=unix_file_name` option. SCO Merge does *not* immediately attach the specified device to your DOS process. SCO Merge attaches the drive only when DOS actually attempts to read or write to the device. The requested device may become unavailable between the start of your DOS session and the time when DOS actually attempts to access it. In this case the DOS process may display an error message or behave incorrectly. To avoid these problems, you can request that the device be attached to your DOS process immediately or exclusively.

To specify immediate attachment, include the `i` key letter following the UNIX filename. The DOS process then aborts immediately if the specified device is unavailable (for example, if another process is writing to it or it does not exist).

The `e` key letter requests exclusive access to the specified device so that no other process can access it. This option is useful whenever you want to prevent other processes from reading or writing to a device while you use it. The `e` key letter automatically implies the `i` key letter. That is, exclusive attachment also means immediate attachment. It is therefore unnecessary to specify `i` explicitly when you use `e`.

The `r` key letter is equivalent to using a write-protect tab on a physical diskette—it prevents you from making any changes to the designated devices. Note that the `r` key letter does not prevent *others* from modifying your files unless you combine it with the `e` key letter.

For example, to set up your DOS environment with the primary DOS partition attached as read-only, you could use a command like this:

```
dos +ae:=/dev/dsk/devicename,r
```

The order of the `i`, `e`, and `r` key letters is immaterial, but each must be preceded by a comma. These key letters can be used only with devices that SCO Merge interprets as files—the physical DOS partition, virtual partitions, and virtual floppies.

Specifying custom devices

Usage: `+acustom_device`

You, your system administrator, or your SCO Merge distributor can create device definitions in the file `/etc/dosdev` for hardware devices other than those described in the previous sections. See Chapter 2 for information on setting up and using custom hardware devices, including details on `/etc/dosdev`.

Canceling device specifications

Usage: **-a**[*device_name*]

The **-a** option prevents access to DOS devices when you use it on the **dos** command line or install it in an application. You can also use **-a** to remove device specifications from options files. The **-a** option can affect access only to those devices you can attach with the **+a** option. You cannot use **-a** to affect access to devices such as the diskette drive that are managed automatically by the SCO Merge system.

Using **-a** to prevent DOS from accessing devices

On the **dos** command line, **-a device_name** prevents DOS from attaching the specified device even if there is an installed **+a** option in an options file or application that requests that the device be attached. For example, the command:

dos -ae:

starts the DOS environment without allowing access to the DOS partition. It is not necessary to type **dos -ae:=/dev/dsk/devicename**. You do not have to supply the UNIX pathname of the device when you use **-a** with the name of a DOS partition (either physical or virtual) or a virtual floppy.

When installed in a DOS application, the **-a** option automatically prevents access to specified devices or to all devices. **-a device_name** prevents access to the specified device unless you override the **-a** option with an explicit **+a** option. For example, the command:

doso~~pt~~ -acom1 xtalk

prevents **crosstalk** from requesting access to the com1 port unless you specify **+acom1** on the **dos** command line.

Use the **-a** option without a specific device name to deny access to *all* devices. For example, the command:

doso~~pt~~ -a xtalk

prevents **crosstalk** from accessing *all* DOS devices. Users can override the installed **-a** by using a **+a** option on the **dos** command line.

Installing the **-a** option in an application to deny access to a device is not the same as *removing* the **+a** option. For example, assume you have installed the **+acom1** option in the application **crosstalk**. The command:

doso~~pt~~ -acom1 xtalk

replaces the **+acom1** request with **-acom1**. Then, when you run **crosstalk** from the UNIX shell, it does not request access to com1 even if */etc/dosenv.def* or your home directory *dosenv.def* contain explicit **+acom1** instructions.

Refer to the description of the **+z** option later in this chapter for information on removing options from applications.

Using -a to remove defaults from options files

You can use a command with the following syntax to remove device specifications from options files:

```
dosopt -a[device_name] options_file
```

For example, the command:

```
dosopt -acom2 /usr/dave/dosenv.def
```

removes the **+acom2** specification in */usr/dave/dosenv.def* but does not affect other **+a** options in that file.

When device specifications are removed from */etc/dosenv.def* files, users can still request access to any device by installing **+a** options in their home directory *dosenv.def* files, by installing **+a** options in their own applications, or by using **+a** on the **dos** command line.

Display-oriented and stream-oriented DOS programs

Usage: **+b**

Factory default: **-b** (display-oriented)

When a DOS program does all its I/O via standard I/O, it is called *stream-oriented*. Most standard DOS commands, such as **dir**, **sort**, and **tree**, are stream-oriented programs. DOS applications that do not write directly to the system screen (many compilers, for example) are also stream-oriented programs.

When a DOS program avoids some or all of the standard I/O mechanisms, it is called *display-oriented*. DOS applications that write directly to the system video memory (including many text-processing, database, and spreadsheet programs, as well as most games) are display-oriented programs.

To properly handle I/O for a DOS program, SCO Merge must know whether it is stream-oriented or display-oriented. The standard DOS commands in */usr/dbin* are assigned the appropriate value (in most cases, stream-oriented) when SCO Merge is initially installed. All other DOS programs are assumed to be display-oriented unless you specify otherwise. You can run any DOS program with the default assignment of display-oriented. To take advantage of the additional flexibility available with stream-oriented programs, though, you should identify these programs as such using the **+b** option. When you

identify a stream-oriented DOS program with the **+b** option, you can use UNIX pipe and I/O redirection mechanisms to manipulate the program's input and output.

To determine whether a program is display-oriented or stream-oriented, follow these guidelines:

- Interactive programs like DOS word processors are generally display-oriented and should have the default (**-b**) assignment.
- Any program that requires a graphics display adapter is display-oriented.
- Most noninteractive programs (such as compilers) are stream-oriented.

If you are uncertain whether a noninteractive program is really stream-oriented, you can perform the following test. Start the DOS environment and run the program in question, redirecting the input or output (as appropriate) to a file. For example, to test output redirection for the program **APPL**, issue this command:

```
C: appl > temp
```

When the program has finished executing, display the contents of the file you created. You can do this with the DOS **type** command:

```
C: type temp
```

Or, you can use a text editor to display the contents of the *temp* file. If the output of your DOS program is correctly captured in this file, the program has stream-oriented output.

To find out whether **APPL** accepts stream-oriented input, you can test it with a command such as:

```
C: appl < con
```

to test input redirection from the console. If **APPL** behaves correctly, it accepts stream-oriented input.

NOTE If you run a program that is really display-oriented with the **+b** option, screen output is lost. To abort the program, send a UNIX interrupt (by default, the **** key).

Pass command directly to DOS command interpreter

Usage: `dos +c command`

Factory default: None

SCO Merge by default performs the following actions when you run a DOS command from the UNIX shell:

- Searches your UNIX path for the executable program.
- If the executable program is found, interprets any DOS options that are installed in the file.

The `+c` option causes the command to be passed directly to the DOS command interpreter, **command.com**. SCO Merge does not search the UNIX path or interpret any DOS options installed in the application when you use `+c`.

The `+c` option is useful with commands that use names (such as DOS device names) that are not meaningful to the UNIX system. For example, the commands:

```
dos +c a:wp
dos +bc "dir | sort"
```

run Word Perfect from the diskette drive and sort the output of the `dir` command. SCO Merge runs both commands properly even without the `+c` option, but it finds the programs more quickly when the `+c` option is included in the command. Without the `+c` option, SCO Merge searches your UNIX search path for files called, literally, "a:wp" or "dir | sort" before passing the command to **command.com**. With the `+c` option, the command is passed immediately, without searching your UNIX path.

Although `+c` prevents SCO Merge from interpreting DOS options installed in the application, options files are still interpreted. Any options specified on the `dos` command line are also interpreted.

Set initial current drive

Usage: `+d[drive]`

Factory default: `+dc`

The `+d` option sets the initial current DOS drive when you enter the DOS environment or run a DOS program. The *drive* parameter can be any drive letter from "a" to "z." For example:

```
dos +dd
```

puts you in the DOS environment with drive D: as your current drive. The `+d` option with no *drive* specification causes the default drive (C:) to be used.

The drive you specify with `+d` can usually be any valid drive—a physical floppy, a physical DOS partition, an extended DOS partition, a portion of the UNIX filesystem being used as a virtual floppy or virtual DOS partition—any drive available to DOS. When the `+d` option is used alone, the only prerequisite is that the drive must exist at the time you name it with the `+d` option. If it does not, DOS aborts.

When `+d` is used in combination with the `+p` option to run an *autoexec.bat*, the drive must be one of the drives shared by the DOS and UNIX systems. It cannot be a DOS-only drive like E:.

Interpret configuration file

Usage: `+e[file[,file ...]]`
`-e`

Factory default: `+e` (interpret */config.sys* and *\$HOME/config.sys*)

Use the `±e` option to tell SCO Merge whether or not DOS should interpret a configuration file when it runs. By default (`+e`), DOS interprets two files in succession:

1. The system default *config.sys* file in the *root* directory (*/config.sys*).
2. Your personal home-directory *config.sys* file.

\$HOME/config.sys files do not exist until users create them. If one of these files does not exist, SCO Merge interprets the file that does exist. When both files exist, SCO Merge behaves as though the contents of */config.sys* and the home-directory *config.sys* are all in a single *config.sys* file.

To specify that you want a different file to be interpreted when you run DOS, name the file (by full path name) along with the **+e** option. For example:

dos +e/usr/dave/dbin/config.sys

runs the DOS environment with the configuration specified in */usr/dave/dbin/config.sys* instead of the configuration specified in */config.sys* and *\$HOME/config.sys*. The specified file can be any UNIX file—it need not be called “*config.sys*.” It must, however, contain valid *config.sys* commands.

When you specify two or more files, DOS interprets the contents of the files, in order, when it starts. For example:

dos +e/usr/ldbin/config.sys,/usr/dave/dbin/config.sys

No default *config.sys* files are interpreted automatically when a filename appears following **+e**. Therefore, if you want one or more files interpreted *in addition* to the defaults, you must list all files explicitly:

dos +e/config.sys,/usr/dave/dbin/config.sys

The **-e** option tells DOS not to interpret any configuration file.

Help text

Usage: **+h**

Factory default: None

Use the **dos +h** or **dosopt +h** command to display a help screen that briefly describes all the available options. You can use the **+h** option only on the **dos** or **dosopt** command line. It cannot be installed in an options file or in an application.

Alternative DOS load image

Usage: **+l**[*file* | *directory*]
-l

Factory default: **+l** (use system default image)

The default DOS images (one each for monochrome, CGA, and VGA display adapters) are contained in the files:

- */usr/merge/image/std/mono/mono.img*
- */usr/merge/image/std/cga/cga.img*
- */usr/merge/image/std/vga/vga.img*⁵

The option **+l** without a file or directory name means the default image. The option **+lfile** uses the specified DOS image file rather than the default DOS image to execute the invoked program. *file* is the full path name of the alternative image, which can have any legal UNIX name. Note that there is no space between the **+l** and the name of the file.

In the **+ldirectory** form, *directory* is the directory that contains the DOS image subdirectory you want to use. When you use this form, SCO Merge automatically searches the specified directory for an image file that corresponds to your display. The image files in the specified directory must be named *mono/mono.img*, *cga/cga.img*, or *vga/vga.img*.

When you use VGA images, the directory containing the image must also contain the file or *vgarom.img* in addition to *vga.img*. These files are created automatically when you create the corresponding image file. The procedures for creating DOS images are described in Chapter 2.

It is rarely necessary to store different alternative DOS images and load them with the **+l** option because SCO Merge accommodates nearly all configuration changes automatically when DOS is run.

The **-l** option starts a “DOS environment” with no image, booting the virtual machine’s BIOS from the reset address *f000:fff0*. You can use the **-l** option to boot either a physical or virtual diskette. The diskette need not be a DOS diskette as long as it is bootable and is compatible with the virtual personal computer environment you have specified. Many games designed for personal computers, for example, do not run under DOS and need to be booted

5. Hercules display adapters also use the */usr/merge/image/std/mono/mono.img* file.

from the diskette drive using the **-l** option. To use such a game, insert the diskette in the diskette drive and type:

```
dos -l
```

If the game is stored in a virtual diskette, you can run it by typing:

```
dos +aa:=pathname -l
```

where *pathname* is the name of the virtual diskette. When you run a "DOS" session this way on the console, you can use the UNIX MultiScreen feature to view and work with other concurrent UNIX or DOS sessions. The process booted with the **-l** option, however, typically runs independently of any other UNIX or DOS activity. That is, usually there is no connection between the process invoked with **dos -l** and the shared filesystem, and no interprocess communication between DOS or the UNIX system and the **dos -l** process. To stop a session started with **dos -l**, press **<Ctrl><Alt>** or use the **KILL** DOS terminal control code described in Appendix A. The **dosboot** command is a synonym for **dos -l**.

DOS memory size

Usage: **+mn** **Factory default:** **+m1** Whenever you use DOS on the SCO Merge system, the DOS process requests a specific amount of memory for the environment used by the DOS process. Using the **+m** option, you can request a specific amount of Standard Memory (base memory plus extended memory) for a DOS process.

See "Allocating memory for DOS" in Chapter 5, for a discussion of how this memory is used.

To specify explicitly the amount of memory you want, use the form **+mn**, where *n* is an integer between 2 and 16, representing Mbytes. For example:

```
dos +m2
```

requests 2 Mbytes for the DOS environment.

Only windows 3.1 can use Standard Memory beyond 1 Mbyte. The command:

```
dos +m4 win
```

requests four Mbytes and runs Windows.

Remember that SCO Merge overhead consumes a small amount of the memory available to DOS, so the actual amount of memory available to DOS applications is slightly less than the amount of memory that you request. Remember also that specifying too small an amount of memory may cause DOS applications to fail. See "Allocating memory to DOS" in Chapter 5 for details.

Run *autoexec.bat*

Usage: **+p**[*file*]
-p

Factory default: **+p**

Use the **±p** option to tell SCO Merge whether or not DOS should interpret an *autoexec.bat* file when it runs.

Use **+p** alone in options files, in applications, or on the **dos** command line to cause DOS to interpret first */autoexec.bat* and then *\$HOME/autoexec.bat*. Use **-p** to prevent interpretation of any *autoexec.bat* files.

When you use the **+p** option, be sure you do not put a space between the **+p** and the *file*. The **+p** option by itself, with no filename, executes both the systemwide and home directory startup files. If **+p** is followed immediately by *file*, that file is executed as a startup file. But, if **+p** is followed by a space and then *file*, the space sets off *file* as an argument to the **dos** command. SCO Merge interprets *file* as the name of a DOS command to be run. The result is that the systemwide and home directory startup files are executed, then *file* is executed as a DOS command, and the DOS environment is terminated.

The */autoexec.bat* and *\$HOME/autoexec.bat* files do not exist until the system administrator or individual users create them. If one of these files does not exist, SCO Merge interprets the file that does exist. When both files exist, SCO Merge behaves as though the contents of both files are all in a single *autoexec.bat* file. The contents of *\$HOME/autoexec.bat* override the settings made by */autoexec.bat*, because the user's home directory startup file is the second one to be read.

In the form **+pfile**, *file* is the name of a single batch file that DOS is to run. The file can have any legal UNIX name and can be anywhere in the UNIX file-system. You must specify it by its full UNIX path name.

When you run DOS with the **+p** option in effect, a temporary file */tmp/aennn.bat* is created, where *nnn* is the UNIX process ID of the DOS server. This file is normally removed when DOS exits, but if the DOS server is killed, the file is not deleted.

Change DOS printer timeout

Usage: `+s[n]`
`-s`

Factory default: `+s15` (spool to UNIX system with a 15-second timeout)

Both in the DOS environment and from the UNIX prompt, DOS printing by default is handled through the UNIX print spooler. The `±s` option controls when accumulated DOS printer output is sent to the UNIX spooler. By default, DOS printer output is sent to the UNIX spooler when any one of the following events occurs:

1. The program or command exits. “Exit” means the program finishes and your DOS or UNIX prompt (C: or \$) returns. Many commands (such as the DOS **print** and **copy** commands) exit automatically when they are done. Other programs require a specific user action to exit.
2. The amount of time specified by the printer timeout value has elapsed since DOS last printed. The default printer timeout is 15 seconds.
3. DOS receives a break signal (for example, you press <Ctrl>C at the DOS prompt).

Use the `+sn` option to change the default 15-second timeout. The variable *n* is a timeout between 5 and 3600 seconds. If *n* is omitted, the default value of 15 seconds is used. The `-s` option makes the timeout *infinite*, which prevents DOS printer output from being sent to the UNIX system until the DOS process exits or you press <Ctrl>C.

The `±s` option has the same effect as the SCO Merge **printer** command `/t` option. The `±s` option can be used on the UNIX command line or installed in options files and applications. The **printer** command must be used in the DOS environment. Both `±s` and the **printer** `/t` option are effective only when DOS printer output is spooled to the UNIX system. When you attach a printer directly to DOS using a command such as **dos +alp1**, the `±s` option and the **printer** `/t` option have no effect.

Translate DOS switch characters and path separators

Usage: `±t`

Factory default: `+t` (translate switch characters and path separators according to standard SCO Merge conventions)

By default (`+t`), SCO Merge requires UNIX-style switch characters and path separators when you issue DOS commands from the UNIX shell. For example:

`dir /tmp -w`

Before passing the command to DOS, SCO Merge automatically translates this command to:

`dir \tmp /w`

The `-t` option prevents SCO Merge from translating switch characters and path separators on a DOS command line. For example, consider a hypothetical DOS command **compute**, which does simple arithmetic calculations like division and subtraction. With the default translation in effect, a command such as:

`compute 8 - 4`

would be translated to:

`compute 8 / 4`

Translation of `" - "` and `" / "` in contexts like this, where they are not intended to be interpreted as path separators or switch characters, can result in undesired behavior.

To prevent this translation, use the command:

`dos -t compute 8 - 4`

You can also use the **dosopt** command to install the `-t` option in **compute** so it is effective each time you run the application from the UNIX shell:

`dosopt -t compute`

The `-t` option does not prevent the UNIX shell from interpreting metacharacters. If you use metacharacters in a DOS command, follow the guidelines listed in Chapter 8.

Dosopt “verbose” mode

Usage: dosopt +v [*options_file* | *program*]

Factory default: None (do not display acknowledgements)

Use the +v option with dosopt to display the DOS options installed in an application or *dosenv.def* file. For example, a typical display in response to the command:

dosopt +v database

is:

```
File: database
*-a *-b *+dc *+e *+l *+m1 *-p +s20 *+t
* defaults from /usr/dave/dosenv.def.
```

This display shows the values of all DOS options used when **database** is run. These values include both:

- Values that have been installed explicitly in **database** with the dosopt command:

dosopt +s20 database

- Values that are taken from an options file because they are not explicitly installed in **database**.

The values taken from the options file are marked with an asterisk (*), and the options file is listed in the last line of the display. The options file is either the system default options file */etc/dosenv.def* or the user’s home-directory *dosenv.def* file.

You can use the +v option to display the values installed in an options file:

dosopt +v /usr/dave/dosenv.def

You can include the +v option on the same command line as other dosopt options. For example:

dosopt +v +s20 database

This command installs the option +s20. Then it displays all options effective for **database**, as described above.

The command dosopt +v without a program or options file name prints the values that apply to the DOS environment—your home-directory *dosenv.def* file, if you have one, or */etc/dosenv.def*, if you do not.

Set X Client startup

Usage: `+x`

Factory default: X Client Operation Off

The `+x` option forces DOS to start up as an X client, even if the **XMERGE** variable has not been set.

dosopt null option

Usage: `dosopt +y program`

Factory default: None

Use the `dosopt +y` option to create a null DOS options record in a DOS application. The `+y` option is only necessary for DOS `.com` files that are to be executed from the UNIX shell and that have not already had DOS options installed using the `dosopt` command. The options record that `+y` creates has no effect except that it allows SCO Merge to identify a `.com` file as a DOS program when you run it from the UNIX shell. For example, assume you want to run the DOS `calc.com` program from the UNIX shell. Install the `+y` option by typing:

`dosopt +y calc.com`

The standard MS-DOS `.com` files distributed with SCO Merge already have appropriate `dosopt` records, and you do not need to use `dosopt` to change the records unless you want to change the default behavior of these commands.

Remove assigned options

Usage: `dosopt +zoption program`
`dosopt +Z program`

Factory default: None

Use the `+z` option to remove an option previously assigned to a DOS application with `dosopt`. For example, you might use `dosopt` to identify the DOS program `work` as a stream-oriented program requiring 512K bytes of expanded memory and using the DOS configuration file `/usr/dave/config.new`:

`dosopt +b +aems512 +e/usr/dave/config.new work.exe`

Values other than those assigned explicitly to **work** come from an options file—either */etc/dosenv.def* or *\$HOME/dosenv.def*. The command:

```
$ dosopt +zb +ze work.exe
```

removes the **+b** and **+e** options from **work** and allows the option values installed in the relevant options file to take effect. Each **+z** requires one and only one option letter. If you want to remove more than one option, use **+z** once for each option you are removing, as illustrated above. The command **dosopt +za *application*** removes all **±a** options from *application* if there are multiple **±a** options installed at the time you use the **+z** option. The command **dosopt +zadevice *application*** removes only the **+adevice** option without affecting any other **±a** options that might exist in *application*.

The **+Z** option removes *all* installed options from DOS applications. For example, the command:

```
$ dosopt +Z work.exe
```

removes the complete record of assigned options from **work** and allows all options in the relevant options file to take effect.

You cannot use the **+z** or **+Z** option on options files (*/etc/dosenv.def* or *\$HOME/dosenv.def*).

Chapter 7

Using SCO Merge NLS features

This chapter deals with configuring your SCO Merge system for use with languages other than U.S. English. It describes the SCO Merge features that support multiple languages in an environment shared by DOS and the UNIX operating system.

This chapter includes the following topics:

- the features offered by SCO Merge NLS,
- prerequisites for using NLS features,
- the theory behind National Language Support,
- an explanation of DOS Code Pages and UNIX Code Sets,
- how DOS handles NLS procedures,
- how the UNIX system handles NLS procedures,
- how SCO Merge handles NLS procedures,
- sample NLS procedures for VGA displays,
- sample NLS procedures for ISO terminals,
- how to avoid name conflicts in an NLS environment,
- how to print in an NLS environment,
- how to use the utilities provided for text conversion.

Features

As delivered, your system recognizes and displays all the characters necessary for English as written in the U.S. If you use only data in *U.S.* format, you can skip this chapter entirely. If you use data in other languages, however,

you need to reconfigure the system to recognize characters that are not part of the default character set.

Support of multiple languages is often called *National Language Support* or *NLS*. SCO Merge NLS includes the following features:

- Support of multiple DOS code pages. You can select the DOS code page best suited to the language you use.
- Automatic translation of keyboard input and displayed output. For example, you can view a DOS text file that contains non-ASCII characters even if you use an ASCII terminal.
- The text conversion programs **dos2unix**, **unix2dos**, and **charconv**. You can use these programs to convert text from one code page or code set to another and convert from DOS to UNIX format or vice versa.

Prerequisites

To take full advantage of SCO Merge NLS, you need to run DOS commands and applications on a VGA console, in a VGA DOS environment inside an X window, or on an ISO terminal. ISO terminals conform to standards set forth by the International Standards Organization and support the ISO Latin-1 character set. Digital Equipment Corporation's VT320 is one well-known example.

Always read the NLS sections of your DOS documentation before you try to do NLS procedures. This chapter assumes you have some familiarity with DOS code pages. The main concepts and procedures are described here, but consult your DOS manuals for details. Also see your DOS manuals for more complete information on related DOS commands such as **chcp**, **country**, **keyb**, **mode**, and **nlsfunc**.

Additional useful information about UNIX operating system NLS characteristics appears in the *SCO UNIX System International Operating System Guide* and the *SCO UNIX System Operating System International Supplement: Release and Installation Notes*.

You can use the material in this chapter in several different ways, depending on your level of familiarity with NLS.

- **If you who are already familiar with NLS concepts and functions**, you can skip the Theory section entirely or simply skim it for new information.
- **If you are unfamiliar with NLS procedures**, refer to your DOS documentation first. Get a good understanding of the commands and procedures DOS uses. Then read the Theory section of this chapter, paying special attention to the few differences in the way SCO Merge handles character sets.

- **If you are in a hurry** to get a system set up for non-English computing, you can skip the Theory section of this chapter temporarily. Nevertheless, you should read that section when time permits; it presents information that you need for full comprehension.
- **If it does not work as you expect**, keep in mind that the procedures described in the second section are intentionally brief. They make certain assumptions about how your system is set up. If you follow the instructions and your system does not respond as expected, it probably means that your actual setup differs from those assumptions. If that happens, read your DOS documentation and the Theory section for a full understanding of the underlying concepts. Then customize the procedures for your own particular needs.

Theory of National Language Support

As delivered, your SCO system is configured to recognize the characters of English as written in the United States of America. You can enter all the characters of U.S. English from the keyboard and display all those characters on the screen. If all your computing is done with this kind of data, you do not need to do anything further.

If, however, you want to perform computing functions on data that uses the characters or conventions of other languages, you need to reconfigure the system. The keyboard must be remapped so you can enter non-U.S. characters. The screen needs to be provided with information on how to display non-U.S. characters. Additional data must be provided to account for differences in the conventions used for punctuation, dates, and currency. Both DOS and the UNIX operating system provide standardized functions for this language-related reconfiguration.

U.S. English is written with twenty-six characters. Other European languages use most of these same characters, but often add other glyphs to express sounds not present in English. The majority of these are written as Roman characters with diacritical or accent marks added, but others are derived from Greek.

DOS supports different character sets that fill the needs of most European languages. See the table later in this chapter under "DOS National Language Support codes" for a list of the character sets that SCO Merge supports. SCO Merge currently does not support any non-European languages such as Japanese.

DOS code pages and UNIX operating system code sets

All symbols used by computers are stored and manipulated as numbers. An eight-bit byte of computer data offers 256 distinct bit combinations. All European languages are encoded in relatively few characters—few enough to be encoded in a single byte of computer data.

Both DOS and the UNIX system use tables that determine how the numeric codes stored in the computer are converted into displayed letters, numbers, symbols, and other characters. The DOS tables are called *code pages* and the UNIX tables are called *code sets*.

Differences between DOS code pages and UNIX code sets

Code pages and *code sets* are essentially just two names for the same type of object: a table that relates 256 written symbols to the 256 bit combinations available in a single byte. Different languages are written with different character sets. Therefore, different code pages and code sets are required to represent different languages.

If you use only DOS or only the UNIX system, you do not need to be aware of any differences between DOS code pages and UNIX code sets. When you use SCO Merge to combine DOS and the UNIX system, however, you may notice some differences between the DOS and UNIX environments. Some characters in data and filenames created with one operating system may be displayed as different characters when you view them with the other operating system. For example, suppose you have a DOS file called *memo* containing the following text:

This memo describes the features of our æNET product.

If you issue the command `cat memo` from the UNIX shell, the words might be displayed like this:

This memo describes the features of our *NET product.

Similar character conversions can occur when you view DOS text on different types of terminals.

These character transformations occur because each character is represented by a code number. The same code number may exist in both your current DOS code page and UNIX code set, but it may be matched with a different symbol in each.

Later sections in this chapter explain how to set up your DOS and UNIX environments so you get the most consistent behavior when you combine DOS and the UNIX system. They also describe the SCO Merge `dos2unix`,

`unix2dos`, and `charconv` programs, which you can use to convert text files in a variety of different ways.

How MS-DOS handles NLS

These sections cover the concepts and procedures MS-DOS uses to provide National Language Support.

Varieties of DOS code pages

MS-DOS recognizes two kinds of code pages. A *hardware* code page is built into a hardware device. A *software* code page is provided in software form and stored in code page information (*.cpi*) files.

Following are some of the code pages commonly supported by MS-DOS and various hardware devices:

- **Code Page 437** supplies characters for U.S. English.
- **Code Page 850** is called the Basic Multilingual Code Page. It supplies characters that suffice for most of the languages of Europe, North America, and South America.
- **Code Page 852** supplies characters for Slavic languages.
- **Code Page 860** supplies characters for Portuguese.
- **Code Page 863** supplies characters for Canadian French.
- **Code Page 865** supplies characters for Nordic languages.

In the U.S., hardware devices commonly use code page 437 by default. Hardware devices designed for use in other countries use other code pages by default.

Not all devices can recognize all software code pages. Some printers, for instance, can print only the symbols contained in their hardware code pages. Thus, a particular character may be stored in the computer's memory as part of a software code page, but the attached printer may be unable to print that symbol correctly when it is sent. Some DOS monitors are limited in the characters they can display for the same reason. See "Display considerations," later in this chapter, for further details.

Not all Code Pages and Code Sets are supported for every operation. A particular character set may be viewable on the console, for instance, but not viewable at an attached terminal or printable by the printers attached. See the *SCO Merge Release and Installation Notes* for a list of the DOS Code Pages and UNIX Code Sets that SCO Merge supports.

Code page switching

DOS allows you to work alternately in several languages on the same machine. This is accomplished by *code page switching*. You can work first in the characters of one language, then switch code pages to work in the characters of another language. Each device affected must be prepared ahead of time with the DOS **mode** command. The actual switching is done with the **chcp** (CHange Code Page) command.

Tailoring DOS for different languages

The following list summarizes the procedures required to set up a standard DOS computer for a different character set. Refer to your DOS documentation for more complete information about following these procedures on a conventional DOS computer. Specific examples of these procedures as they apply to the SCO Merge environment appear later in this chapter.

- **Configuring for country** is done with the **country** command. This command is placed in the *config.sys* file. It sets country-specific variables, such as the formats used for dates, times, and currencies. It also establishes the character collation sequence.
- **Configuring system devices** is done by placing **device=** statements in the *config.sys* file. These statements tell DOS which hardware and software code pages to associate with each device.
- **Loading memory-resident NLS code** is done by placing the **nlsfunc** command in the *autoexec.bat* file. This must be done before you try to specify code pages or keyboard codes.
- **Preparing (loading) code pages** is done by placing the **mode device codepage prepare** command in *autoexec.bat*. It prepares code pages for those devices that support code page switching.
- **Mapping the keyboard** is done with the **keyb** command.
- **Switching between code pages** is accomplished with the **chcp** command.

How the UNIX operating system handles NLS

A UNIX operating system must perform the same basic activities described above for DOS, but the system interface is simpler. Nearly everything is done by defining a single **LANG** variable, which takes care of setting all other language- and place-related variables. **LANG** is of the form:

LANG=language_territory.codeset

For instance, **LANG=french_swiss.8859** would set the language to French as spoken in the French-speaking sections of Switzerland, with ISO Latin 1 (ISO

8859-1) used as the character set. All dates, times, and currencies would be expressed as is customary in that region.

There are two different ways that LANG can be defined:

1. In the file */etc/default/lang*. This file contains the LANG definition in the syntax shown above. Although this syntax is characteristic of environment variable definitions, the LANG variable as defined in */etc/default/lang* is not actually an environment variable.¹ The definition in this file applies to all users unless they override it using the second method.
2. As an environment variable. If you define LANG as a UNIX operating system environment variable, this definition overrides the definition in */etc/default/lang*. You can define LANG just as you would define any other UNIX operating system environment variable. For example, if you use the Bourne shell, you can define the LANG environment variable at your shell prompt using the following syntax:

```
LANG=language_territory.codeset
export LANG
```

In addition, the UNIX operating system collects user messages into *message files*. This simplifies translating the operating system and its applications into a new language. All the message strings are translated into the new tongue and collected into a new message file.

Switching languages under the UNIX system, then, is a simple matter of resetting one variable.

How SCO Merge handles NLS

This section describes how SCO Merge integrates the characteristics of DOS and UNIX operating system NLS support.

Display considerations

A VGA console can display any character set for which the font files are loaded. The nature of your display, however, limits the availability of automatic character conversion and the other NLS features offered by SCO Merge.

1. For example, the command `echo $LANG` does not display the LANG definition when LANG is defined only in */etc/default/lang*.

The complete set of SCO Merge NLS features is available when you run DOS:

- on an ISO terminal,
- on a VGA console, or
- in a VGA DOS environment inside an X window.

Throughout this chapter, the phrase *VGA display* refers to a VGA console or to a VGA DOS environment inside an X window. The phrase *VGA session* refers to a DOS session run on any of these displays.

Only a subset of SCO Merge NLS features is available when you use any of the following displays:

- MDA, CGA, or Hercules consoles,
- standard ASCII terminals
- PC scancode terminals.

On a conventional DOS computer, monochrome, CGA, and Hercules displays are always restricted to DOS code page 437. The same restriction applies to these displays when you use SCO Merge. However, you can use the SCO Merge `charconv` command to convert text files created with other code pages or code sets to code page 437 before using them with these displays. `charconv` is described near the end of this chapter.

The same restrictions apply to standard ASCII terminals and PC scancode terminals. In addition, there are some DOS characters that cannot be displayed on ASCII terminals. When you use DOS to display text containing these characters, SCO Merge replaces them with asterisks.

Setting up the SCO Merge NLS environment

There are two ways to set up your SCO Merge environment for NLS. One uses the DOS *startup files* `autoexec.bat` and `config.sys`; the other uses *DOS images*. Both require that you also set certain UNIX environment variables.

The “startup files” method is similar to the procedures you would follow on a standard DOS computer, which require adding appropriate commands to your `autoexec.bat` and `config.sys` files. With this method, as on a standard personal computer, every time you start DOS, these commands are executed and your DOS environment is set up for NLS.

The second method allows you to put DOS NLS commands into appropriate DOS images. A DOS image is a frozen picture, or snapshot, of DOS after it has been loaded into memory and is running. This image includes information DOS needs about the virtual PC configuration. There is a DOS image for every supported display type (VGA, Hercules, CGA, and monochrome). By install-

ing NLS commands and drivers in DOS images instead of in *autoexec.bat* and *config.sys*, DOS sessions started with these images are automatically configured for an NLS environment.

The advantage of putting NLS commands in *autoexec.bat* and *config.sys* is its simplicity. By putting a few commands in these startup files, any user can quickly set up an NLS DOS environment. Different users can have different NLS environments by putting NLS commands appropriate for their needs in their personal *autoexec.bat* and *config.sys* files.

The “startup file” method has several drawbacks, however. Since these NLS commands are executed every time you start DOS, they can slow down the DOS startup time. In addition, while a VGA DOS session can support all NLS commands, a CGA or a monochrome DOS session supports only a limited number of these commands (see the next section). Therefore, every time you start a CGA or monochrome DOS session, you see several NLS-related error messages as NLS commands are executed by your startup files. Although not harmful, these error messages can be distracting.

The second method for setting up NLS (using DOS images) overcomes most of these shortcomings. Since there is a separate image for each type of display, you can put only the appropriate NLS commands in each image. In addition, since an image contains an environment that is already set up, the system spends no time executing these NLS commands every time you start DOS. (The commands have already been executed.) There are two drawbacks to this method. First, it loads the NLS commands permanently into your DOS environment and they can take up precious RAM. Second, they affect all users. In addition, you must be logged in as root to create DOS images.

If you use NLS only on occasion or if only a few users on your system need it, then we recommend the first method. However, if you want to set up system defaults for a particular NLS environment, we recommend the more permanent DOS image method.

Procedures for setting up ISO terminals for multilanguage use are different from the procedures for setting up other displays. A later section of this chapter describes the steps you need to follow if you use an ISO terminal.

Setting UNIX operating system environment variables

SCO Merge uses the UNIX operating system environment variables **CODEPAGE**, **COUNTRY**, and **KEYB**, in addition to the **LANG** variable described earlier. The **LANG** definition is used by SCO Merge as well as other SCO applications. The **CODEPAGE**, **COUNTRY**, and **KEYB** variables exist specifically to support SCO Merge NLS functions. System-wide values for these variables are defined in the files */etc/profile* and */etc/cshrc*. They have the following default values:²

```
CODEPAGE=pc437
COUNTRY=1
KEYB=us
```

To provide the greatest consistency between the UNIX and DOS environments, defines these three environment variables to correspond to the values you use with the DOS **mode**, **chcp**, **country**, and **keyb** commands. Note that the **CODEPAGE** environment variable has "pc" prepended to the code page number. Define **COUNTRY** and **KEYB** using the same values you use in DOS. See the table in the next section for a list of valid country codes, code pages, and keyboard codes.

Users can customize their own NLS environments and the system administrator can change the defaults for all users by redefining these environment variables.

DOS National Language Support codes

The table that follows is a chart of languages (by country) supported by SCO Merge. Column 2 gives the code to use with the **country** command. Column 3 shows the default code page used for that country and the alternate code page that is usually switched in with the **chcp** command. Column 4 gives the code issued with **keyb** to remap the keyboard.

Consult the *SCO Merge Release and Installation Notes* for information about any newly supported languages that are not listed here.

2. Throughout this chapter, examples showing definitions of UNIX environment variables use Bourne shell syntax and refer to the Bourne shell startup files */etc/profile* and *.profile*. If you use the C shell, you should modify the examples to use proper C shell syntax and to refer to the equivalent C shell startup files */etc/cshrc* and *.cshrc*.

Supported languages

Country or language	Country code	Code pages	Keyboard code
United States	001	437,850	us
Latin America	003	437,850	la
Netherlands	031	437,850	nl
Belgium	032	437,850	be
France	033	437,850	fr
Spain	034	437,850	sp
Italy	039	437,850	it
Switzerland	041	437,850	sf,sg
United Kingdom	044	437,850	uk
Sweden	046	437,850	sv
Germany	049	437,850	gr
Australia	061	437,850	us
Finland	358	437,850	su

7

NLS commands supported by SCO Merge

Some DOS NLS commands are supported only in VGA DOS sessions. You should use these commands only if you use one of these environments. The following table lists the DOS commands that are supported on each display type:

	VGA displays ³	ISO terminals	Other displays
CONFIG.SYS COMMANDS	country ⁴ device= ⁵	country ⁴	country ⁷
OTHER DOS COMMANDS	chcp keyb mode ⁶	chcp	None

3. In this table, "VGA Displays" are VGA DOS environments run inside an X window.
4. SCO Merge does not support the optional third **country** parameter, which specifies the *country.sys* file to use. SCO Merge always uses *\country.sys* on the current drive.
5. **DEVICE=** entries for NLS features such as *display.sys*.
6. **mode** options for NLS features such as **codepage prepare**.
7. On displays other than VGA or ISO, SCO Merge supports only the first **country** parameter, which specifies the country number.

SCO Merge exceptions

When you use SCO Merge, you do *not* need to run the DOS `nlsfunc` command as you would on a conventional DOS computer. SCO Merge automatically runs `nlsfunc` whenever you start DOS.

Sample NLS setup procedures for VGA displays

Following are example NLS setup procedures for VGA displays. Both the startup file method and the DOS image method are described here. The instructions are based on a specific but typical example that assumes you want to type and display data in French instead of English. You can modify these instructions to match your own requirements by using different country codes, code pages, keyboard codes, and environment variable definitions.

Assume you want to use the DOS multilingual code page (850) and the French country code (33) by default.

Setting up the startup files

Follow these procedures to set up your DOS NLS environment using the startup file method:

1. Add the following line to your home directory `config.sys` file:

```
device=c:\usr\dbin\display.sys con=(ega,437,1)
```

This line sets up your VGA display so it can accept one code page in addition to the built-in hardware code page. The built-in hardware code page is 437.

NOTE Use "ega" in the "con=(ega,437,1)" portion of the "device=" line shown above for VGA displays.

2. Add the following line to your home directory `autoexec.bat` file:

```
mode con codepage prepare=((850)c:\usr\dbin\ega.cpi)
```

This line prepares the VGA display for the additional code page (850), which is to be retrieved from the file named.

Note that you do not have to run the `nlsfunc` command because SCO Merge automatically loads `nlsfunc` whenever you run DOS. Including an `nlsfunc` command in `autoexec.bat` does no harm, but it has no effect on SCO Merge NLS functionality.

3. Define the UNIX CODEPAGE, COUNTRY, and KEYB environment variables by typing:

```
CODEPAGE=pc850
COUNTRY=33
KEYB=fr
export CODEPAGE COUNTRY KEYB
```

Add these lines to your home directory *.profile* file if you want them to be effective permanently.

Now, whenever you start DOS, code page 850 and country code 33 are set up automatically. You can also switch back to code page 437 by using the DOS `chcp` command.

If you want code page 850 to apply when you run DOS applications directly from the UNIX prompt, be sure SCO Merge interprets your home directory *autoexec.bat* and *config.sys* files. All DOS applications automatically interpret these files unless you have used the `±p` or `±e` options to change the default behavior. (See Chapters 5 and 6 for more information on DOS options.)

Using DOS images

An alternative way to set up your DOS NLS environment is to create new DOS images loaded with the NLS commands that prepare the code pages you want to use. When you use the DOS image method, you need to make new images only when you want to use a code page that is not prepared in the current images. You must be logged in as root to perform the steps described in this section.

As with the previous example, we assume that you want to use the DOS multilingual code page (850) and the French country code (33) by default. Only the system administrator, logged in as root, can make DOS images, so you must be logged in as root to follow these procedures.

1. Modify the definitions of CODEPAGE, COUNTRY, and KEYB in */etc/profile* as follows:

```
CODEPAGE=pc850
COUNTRY=33
KEYB=fr
```

2. Change to the directory where the DOS VGA image is located by typing the following command:

```
cd /usr/merge/image/std/vga
```

3. Use a DOS editor to create two files with the same names as the DOS startup files: *autoexec.bat* and *config.sys*. If these files already exist, then use an editor to add the lines specified in the next step.

4. In the *config.sys* file, add:

```
device=display.sys con=(ega,437,1)
```

This line sets up your VGA display so it can accept one code page in addition to the hardware code page. The built-in hardware code page is 437.

NOTE Use "ega" in the "con=(ega,437,1)" portion of the "device=" line shown above for VGA displays. Also note that, unlike the previous example, the "display.sys" file on the "device=" line should *not* have a full path.

5. In the *autoexec.bat* file, add:

```
mode con codepage prepare=((850)ega.cpi)
```

This line prepares the display for the additional code page (850). Also note that, unlike the startup file method, the file *ega.cpi* should not have a full path.

6. Now make new images by typing the following command:

```
mkimg
```

Your DOS environment is now set up for NLS.

Resetting the DOS image

If you want to remove the NLS commands from the image, follow these procedures:

1. Remove the lines you added to the *config.sys* and *autoexec.bat* files in the image directory. You may delete the files if they have no other commands in them.
2. Return the **CODEPAGE**, **COUNTRY**, and **KEYB** variables in */etc/profile* to their original values
3. Build new images by typing **mkimg**.

Sample NLS setup procedures for ISO terminals

As before, this example assumes you want to use DOS code page 850 and country code 33. Note that some unique steps are required to set up NLS for use with an ISO terminal. These steps must be precisely followed.

1. Define the **ISO_TERM**, **CODEPAGE**, **COUNTRY**, and **KEYB** environment variables by typing:

```
ISO_TERM=1
CODEPAGE=pc850
COUNTRY=33
KEYB=us
export ISO_TERM CODEPAGE COUNTRY KEYB
```

Add these lines to your home directory *.profile* file if you want them to be effective permanently. The **ISO_TERM=1** coding tells the system that this is an ISO terminal. The **KEYB** environment variable must *always* be defined as "us".

2. Determine the current value of **LANG**. The current **LANG** value may be defined in either of two ways. First type the following command to see if **LANG** is defined as an environment variable:

```
echo $LANG
```

If this command displays a value (for example, "french_france.ascii"), then this is the current **LANG** value.

If the **echo \$LANG** command does *not* display a value, inspect the contents of the file */etc/default/lang* by typing:

```
cat /etc/default/lang
```

This file contains a **LANG** definition in the form

```
LANG=language_territory.codeset
```

If the **echo \$LANG** command does not display a value, your current **LANG** value is the value shown in */etc/default/lang*.

For this example scenario, assume that the current **LANG** value is "french_france.ascii".

3. If the code set portion of the **LANG** environment variable is *not* already "8859," redefine the **LANG** environment variable. To redefine **LANG**, type:

```
LANG=french_france.8859
export LANG
```

Add these lines to your home directory *.profile* file if you want them to be effective permanently. Note that ISO terminals *always* require the code set portion of the **LANG** definition to be 8859.

Now, whenever you run DOS, code page 850 and country code 33 are set up automatically. You can also switch to code page 437 by using the DOS `chcp` command.

Naming DOS and UNIX operating system files

Using SCO Merge describes how SCO Merge treats DOS and UNIX file names, including the SCO Merge filename mapping feature. (SCO Merge maps filenames when you use DOS to access a file created by the UNIX system with a name that is illegal in DOS. For example, SCO Merge might map the UNIX filename *messagetoall* to *MESS'BAQ* when you use DOS.) The description of file naming issues in *Using SCO Merge* applies in a multilanguage environment just as it does in a single-language environment.

In addition, the following points apply when you want to create or access files on the SCO Merge system in a multilanguage environment:

- You can simplify file creation and file access by only using characters from "a" to "z" and "0" to "9" and the characters ".", "-", and "_" in your filenames. All DOS code pages and UNIX code sets contain these characters.
- If you use the UNIX system to create filenames using characters that do not exist in your current DOS code page, SCO Merge maps these file names to legal DOS names when you access them with DOS. For example, the UNIX filename *ms-dos®* might be mapped to *MS-D'BDQ* when you use DOS. SCO Merge maps this filename because the character "®" exists in some UNIX code sets but does not exist in supported DOS code pages. Refer to *Using SCO Merge* for further information on using mapped file names.
- You *cannot* use DOS to create filenames in the shared filesystem using characters that do not exist in the UNIX code set. You *can* create these files in real DOS filesystems, such as the DOS partition or a diskette drive. For example, if the character "μ" does not exist in the UNIX code set, you cannot use DOS to create a file named μNET on drive C: no matter what DOS code page you use. If your current DOS code page does include the "μ", you can use DOS to create μNET on drive E:, drive A:, or any other real DOS filesystem.

Printing in an NLS environment

By default, SCO Merge sends DOS printer output to the UNIX print spooler.⁸ SCO Merge by default does *not* translate DOS text when you print. If the UNIX printer used for DOS printing does not support all the characters you send for printing, the results are unpredictable.

You *cannot* use the DOS **mode** or **chcp** commands to change the printer code page when you use a UNIX printer. However, when you use the UNIX print spooler, you can use the SCO Merge **charconv** command to translate DOS text files before you print them. For example, assume your DOS text file *memo* was created using DOS code page 850 and you want to print it using the UNIX ISO standard code set (8859) and convert characters that do not exist in code set 8859 to the best multibyte approximation. You could convert *memo* using the command:

```
charconv /i pc850 /o 8859 /m /d memo memo.unx
```

You can then print the converted file using standard DOS or UNIX commands (for example, **copy memo.unx lpt1**).

You can also use the SCO Merge **printer** command to change the default UNIX print command that SCO Merge uses when you send DOS printer output to the UNIX spooler. For example, if you want to use the **charconv** command shown above whenever you print from DOS, you could issue the following command at your DOS prompt or in your *autoexec.bat* file:

```
printer unix "charconv -i pc850 -o 8859 -m -d | lp"
```

After you issue this **printer** command, all text files you print from DOS are converted automatically.

Refer to the next section, "Converting text files," for more information on the **charconv** command. See Chapter 2 for more information on the **printer** command.

Instead of using the UNIX print spooler, you can attach a printer directly to your DOS process. When a printer is directly attached to DOS, you can use the DOS **mode** or **chcp** commands to change printer code pages and use it in all other ways exactly as you would on a conventional DOS computer. See Chapter 2 for further information on setting up and using a directly attached printer.

8. SCO Merge uses the script */usr/lib/merge/lp* to spool DOS printer output.

NOTE The examples using **charconv** and **printer unix** in this section are relevant only for text files. It is not necessary to convert DOS graphics or other nontext files before printing.

Converting text files

SCO Merge has three commands that you can use to convert text files so they are usable with different code pages and code sets. The commands are **dos2unix**, **unix2dos**, and **charconv**. All of these commands are usable both in the DOS environment and at the UNIX shell prompt.

Using dos2unix and unix2dos

Using SCO Merge describes how to use **dos2unix** and **unix2dos** to convert text in DOS format to UNIX format and vice versa. In addition, **dos2unix** and **unix2dos** by default translate each character in your text from the DOS code page to the corresponding character in the UNIX code set or vice versa. When you use these commands in the DOS environment, SCO Merge translates between your current DOS code page and the UNIX code set defined by the **LANG** environment variable. When you use them at the UNIX shell, SCO Merge translates between the DOS code page defined by the **CODEPAGE** environment variable and the UNIX code set.

When all characters in your text file exist in both the code page and the code set, the converted output looks exactly the same as your original, unconverted text file (except that it is now usable with a different operating system). When your original text file contains characters that do not exist in both the code page and code set, **dos2unix** and **unix2dos** by default convert untranslatable characters into asterisks (*).

You can modify the behavior of **dos2unix** and **unix2dos** by using the same options that apply to the **charconv** command, which is described in the next section.

Using charconv

The **charconv** command is a flexible and powerful text conversion tool. **charconv** is most useful in mixed-language (NLS) environments. For simple file conversion cases, use **dos2unix** and **unix2dos**. Unlike **dos2unix** or **unix2dos**, **charconv** has no defaults for code pages and code sets. The **/I** and **/O** options are required unless you use the **/X** option.

The syntax of the **charconv** command is:

charconv [*options*] *sourcefile* [*targetfile*]

The source and target files must not be the same. When neither file parameter is specified, **charconv** prints a usage message to your screen. If you specify only one file parameter, that file is considered the source file, and the output is written to standard output.

charconv options

When issued from the UNIX command line, **charconv** options must be specified with a hyphen (-) rather than a slash (/). In the DOS environment, you can use either the slash or the hyphen. The examples here show the hyphen, which will work in either environment. You can use either uppercase or lowercase options both from the UNIX shell and in the DOS environment.

All of the options below can be set on the command line or with the **CONVOPTS** ("**charconv** options") environment variable, which is described later.

- /?** Displays a list of valid **charconv** options on your screen.
- /7** Notifies you if **charconv** encounters any 8-bit characters and translates them to 7-bit characters. You cannot use this option with **/B**.
- /A** Causes **charconv** to quit if it encounters a character it cannot translate accurately. This option overrides the **/C**, **/M**, and **/S** options, which allow inexact translations.
- /B** Preserves 8-bit (binary) character representations. This option is on by default.
- /C x** Specifies *x* as the character to use for untranslatable characters. The default character is an asterisk (*).
- /D** Converts a file from DOS to UNIX format by removing carriage returns from the end of each line.
- /I tbl** Identifies the code page or code set used for the input file. This option is required unless you use the **/X** option. Refer to the *SCO Merge Release and Installation Notes* for a list of supported code pages and code sets.
- /O tbl**

Identifies the code page or code set used for the output file. This option is required unless you use the **/X** option. Refer to the *SCO Merge Release and Installation Notes* for a list of supported code pages and code sets.

For example, if you wanted to convert a DOS file from code page 437 to code page 850, use the **charconv** command as follows:

```
charconv -i pc437 -o pc850 sourcefile targetfile
```

- /L** Converts text to lowercase.
- /M** Converts single, untranslatable characters into multibyte characters, if possible. For example, ¾ is converted to 3/4. If a multibyte conversion is not possible, **charconv** uses the default asterisk (*) character.

- /P** Converts a text file from UNIX to DOS format by adding carriage returns to the end of each line.
- /Q** "Quiet." Prevents **charconv** from printing warning messages and character translation statistics to your screen. This is the default. Use **/Q** if necessary to override the **-V** option, which causes warning messages and conversion statistics to be displayed.
- /S** Converts single, untranslatable characters into the best single character translation. For example, ***I** (the DOS graphic character for intersection) is converted to **+**. If there is no best single translation, **charconv** uses the default asterisk (*****) character.
- /U** Converts text to uppercase.
- /V** Displays warning messages and character conversion statistics on your screen.
- /X** Specifies that **charconv** should not translate from code page to code set or vice versa. Use this command with the **/D** or **/P** options when you want to translate between the DOS and UNIX operating systems without changing code pages.
When you specify **/X**, **charconv** does not allow you to use the **/I** or **/O** options.
- /Z** Causes **charconv** to stop processing when it encounters the DOS end-of-file (EOF) character, **(Ctrl)Z**. This option removes the end-of-file marker when converting text from DOS to UNIX. By default, **charconv** converts the whole file.

Using the CONVOPTS environment variable

The **CONVOPTS** environment variable allows you to set **charconv** options once so you only need to type **charconv filename** to accomplish a task.

For example, suppose you want to convert a series of files from code page 437 to code page 850 and make all of the characters uppercase.

If you use **charconv** in the DOS environment, set the **CONVOPTS** environment variable on the DOS command line as follows:

```
set convopts=/u /i pc437 /o pc850
```

If you use **charconv** at the UNIX shell, define the **CONVOPTS** variable as follows (this example assumes you use the Bourne shell):

```
CONVOPTS="-u -ipc437 -opc850"  
export CONVOPTS
```

Now, you can just type the following line for each file:

```
charconv sourcefile targetfile
```

and **charconv** automatically converts from code page 437 to 850 and makes all characters uppercase.

Chapter 8

Other advanced topics

This chapter provides information of interest both to advanced SCO Merge users and to the SCO Merge system administrator. It covers a wide variety of techniques for customizing DOS and improving SCO Merge efficiency. It is organized as follows:

- “Changing Colors in DOS Windows” describes the mechanisms offered by SCO Merge for adjusting colors.
- “X System Resources for SCO Merge” describes the X resources users can set to control the appearance of their DOS windows.
- “Using Expanded Memory (EMS)” describes the *config.sys* entry for the SCO Merge EMS device driver.
- “Using Multiport Serial Cards” provides hints for using these cards with DOS processes.
- “SCO Merge Compatibility” compares the SCO system environment with a conventional, stand-alone DOS environment and discusses the requirements that DOS applications must observe in order to operate correctly with SCO Merge.
- “Passing UNIX Environment Variables to DOS” shows you how to customize several DOS features by using UNIX environment variables.
- “Running DOS or a DOS Application as the Default Shell” tells you how to start DOS or a specific DOS application as your default environment when you log into your SCO Merge computer.
- “Scheduling DOS Programs” illustrates how you can use the UNIX **at**, **batch**, **cron** and **nohup** commands to control the execution of DOS processes.

Changing colors in DOS windows

The X server you run may affect the color response your DOS programs display when you run them in a window. Systems running a 256-color X server will receive the same color response in an X window that the DOS program would give on a standard DOS system. Systems running a 16-color X server may require some extra tuning.

When you run SCO Merge programs in an X window on a 16-color server, you may occasionally need to redefine the colors appearing on your screen. To do this correctly, you need to understand the layers of software that are interacting and how they handle colors. The following discussion applies only to 16-color X server systems.

How DOS and X Windows handle colors

A 16-color X server places a ceiling on the number of colors you can use for each window on your screen.

SCO provides a special X Client, **Color**, which allows each user to choose his own Color Palette, a set of eight shades (plus black and white). Once you choose such a palette, it must serve the needs of all the X clients you run, including any DOS program you run in an X window.

When running in full-screen mode, SCO Merge supports all the normal DOS video standards. You can use MDA, Hercules, CGA, EGA or VGA. When running a DOS program in an X Window, however, EGA and VGA display modes are not supported. On a 16-color X server, text screens can display up to 16 colors on the screen at one time. These are predefined colors which cannot be changed. Graphics screens can be 4-color CGA or Hercules graphics.

When you zoom the window, the program running in that window gains full control of the screen. A DOS program running in a zoomed window will give you exactly the same color response it would under ordinary DOS. If you are running a VGA program, with a VGA monitor and with SCO Merge set for VGA mode, the program has access to the full range of VGA colors, up to 256 in some VGA modes. Other VGA modes offer 16 colors, but the user can select from an array of 64 or 256 hues.

When you are running under X Windows, however, the X server limits the number of colors available to each window. When your screen contains both UNIX and DOS windows, color scheme conflicts can occur because DOS programs and the SCO system have different color requirements.

Motif Windows achieve their sculptured appearance through shadowing effects, which require many shades of the same basic color. DOS screens are often designed around the assumption that they have access to a fixed array of contrasting primary colors. Consequently, when the number of colors available to your screen is small, no single set of tints will adequately serve both purposes. The range of subtle shades suitable to Motif window borders and dialog boxes can produce DOS screens in which the text is unreadable. The contrasting primary colors used by DOS programs generate Motif screens with a bizarre technicolor appearance.

Consequently, when you run both DOS and UNIX programs in windows and you move back and forth between them, you sometimes need to change your color selection.

Installing the DOS colors

To install the sixteen standard DOS colors as the color group being used for all your X windows:

1. In any window running SCO Merge, bring up the DOS Menu by holding down the right mouse button or typing `(Alt)D`.
2. From the DOS Menu, select DOS Colors.

These colors have not been optimized for UNIX work. Therefore your DOS windows will look correct, but any UNIX windows that may peek out from behind a DOS window will display wildly distorted shading. Since you usually work in only one window at a time, this has no effect on your productivity. When you switch to one of your UNIX windows, the last color palette selected through **Color** is automatically installed.

X System resources for SCO Merge

Users can control the appearance of their DOS window through a standard set of X resources. There are many X Resources available. This section lists only those that are really useful. Users can define them in their own personal resource file, `.Xdefaults`. The default values can be found in `/usr/lib/X11/app-defaults/DOS`. You can assign values to any of these resources using the standard X syntax:

dos*resource: string

- **autofreeze** - When set to yes, DOS applications will be suspended in windows when the keyboard focus has been moved to another window. The default is no.

- **autozoom** - When set to yes, applications will zoom automatically when they go into VGA graphics mode under X Windows. The default is no.
- **colors** - Sets the colors to be used by the driver.
- **dos_colors** - The normal method of selecting colors in DOS windows is through the **Color** utility or the "DOS Colors" option in the DOS Menu. We recommend that users apply these methods for all their ordinary needs. Those few users who have very special needs for precise color matching between their DOS and UNIX windows can adjust this mapping by manipulating the **dos_colors** variables.

When the **installcolormap** resource is set to false, SCO Merge uses the default UNIX colormap and attempts to find UNIX colors that best match the needed DOS colors. Users can try to fine-tune this color matching by specifying another UNIX color to be matched to a particular DOS color. For example, the following statement tells SCO Merge to look for the UNIX color "navyblue" to use as the DOS color "blue":

dos*blue: navyblue

If "navyblue" is not found, SCO Merge uses its own built-in match.

Following is the list of DOS colors that can be set:

black	gray
blue	lightblue
green	lightgreen
cyan	lightcyan
red	lightred
magenta	lightmagenta
brown	yellow
lightgray	white

See the SCO Merge system-wide resource file */usr/lib/X11/app-defaults/DOS* for the current default settings.

- **dosfont** - Sets the default font. SCO Merge supplies the following PC fonts:

6x13pc
8x14pc

NOTE Any fixed font can be used as the default font. However, if you plan to take advantage of the National Language Support (NLS) facilities provided under SCO Merge, then you should only use the fonts shown above.

- **installcolormap** - By default, the DOS window uses the color palette established by the color utility (**installcolormap: false**). This is sufficient for most purposes. When standard DOS colors are required in the DOS window, the user can set this variable to **installcolormap: true**. This installs the standard DOS colors for every session. The default is false.

Use of this variable is not encouraged. The DOS Colors button on the DOS Menu installs the standard DOS colors temporarily, for the period of that DOS session only. Since need for true DOS colors is usually infrequent, this method is preferred.

- **menukey** - Specifies the key sequence for invoking the DOS menu. The default is **<Alt>D**. The key sequence can be set to any combination of the **<Ctrl>**, **<Alt>**, and **<Shift>** keys and an ASCII character. Use the standard X name for the key, delimited by **<** and **>**. For example, to set the key sequence to **<Ctrl><Shift><F10>** key, you would set the following in your resource file:

dos*menukey: ctrl shift <F10>

The DOS menu can also be invoked by pressing and holding the right mouse button.

- **windowsHeight** - Sets the height of X windows while in Microsoft Windows mode. The default is 80% of your screen height.
- **windowsWidth** - Sets the width of X windows while in Microsoft Windows mode. The default is 80% of your screen width.

∞

Running SCO Merge on remote servers

Setting the XMERGE environment variable

In order for SCO Merge to display properly on a remote server, the **XMERGE** environment variable must be set on that server before the DOS process starts. It is recommended that it be set to **cga** mode. When SCO Merge is run locally, this environment variable is set automatically.

To set the **XMERGE** variable, use the following commands:

Bourne or Korn shell

XMERGE=cga; export XMERGE

C-shell

setenv XMERGE cga

SCO Merge fonts

When SCO Merge is run remotely, all font files it uses must be installed and compiled on the remote server's machine. To determine the names of the font files to install, examine the file `/usr/lib/merge/fonts/fonts.src`. This file is in two-column format, with the names of all of the fonts in the first column and the corresponding filenames in the second column. The actual font files are obtained from the subdirectory *misc* in the server's standard font directory.

Using expanded memory (EMS)

The DOS section of *Using SCO Merge* describes expanded memory from the SCO Merge user's perspective. Advanced users or the system administrator may need to understand the **EMS** device configuration command in the *root* directory *config.sys* file. Because this command is in the system default *config.sys* file, it is interpreted whenever any user starts a DOS environment or a DOS application, unless the user specifically requests that *config.sys* not be interpreted. The *config.sys* command is:

```
device=c:\usr\lib\merge\emm.sys
```

Due to built-in defaults in the driver, the above line is equivalent to the following:

```
device=c:\usr\lib\merge\emm.sys /start=D000 /address=208
```

In this line, "208" is the base port number used for expanded memory, and "D000" is the segment address in DOS space to which expanded memory is mapped. Do not use a value other than 208 for the base port number. However, you can use a segment address other than D000 if you have other devices that use that address. If you cannot use D000 because of a conflict like this, change the EMS device configuration command in *config.sys* to use one of the following segment addresses instead:

C000	C400	C800	CC00
D400	D800	DC00	E000

Using multiport serial cards

Many cards provide multiple serial ports on a single adapter. Each of these is a separate UNIX device, and UNIX processes can direct output to any of them. DOS, however, recognizes only a limited number of serial ports.

Standard practice assigns UNIX devices `/dev/tty1a` and `/dev/tty2a` to `com1` and `com2` respectively, but SCO Merge allows you to assign any of the available UNIX devices.¹ To use a serial port other than the standard devices for DOS `com1` or `com2`, issue the `dos` command with the `+a` option. Attach `com1` or `com2` to your DOS process and map `com1` or `com2` to the actual UNIX name of the serial port. For example, to run a DOS environment and attach `/dev/tty20` as `com1`, use the command:

```
$ dos +acom1=/dev/tty20
```

You must use indirect attachment, as illustrated in this example, to attach a serial port other than `com1` or `com2`. Attempts to use direct attachment, as shown in the following example, do not work:

```
$ dos +adcom2=/dev/tty15 (incorrect)
```

Refer to Chapter 6 and to the DOS section of *Using SCO Merge* for further information on using com ports.

SCO Merge/DOS compatibility

The following paragraphs summarize the significant technical differences between DOS as it runs on a conventional, stand-alone personal computer and DOS as it runs under SCO Merge. Despite these differences, nearly all DOS applications that run correctly under DOS also run correctly under SCO Merge.

DOS interrupts

DOS commands and applications that use standard DOS interrupt 21 function calls work properly with SCO Merge. Commands and applications that attempt to use interrupts 13, 25, or 26 to access the shared filesystem do not work.

Commands that use interrupts 13, 25, or 26 include **chkdsk**, **fdisk**, **format**, and **sys**. These commands are described as "exceptional" because they do not work on the filesystem shared by DOS and the UNIX operating system.

A few applications use interrupts 13, 25, or 26 to perform low-level disk I/O operations such as:

- "Unerasing" deleted files.
- Repacking segments on the fixed disk to reduce fragmentation.
- Changing segment interleaving on the fixed disk.

1. SCO Merge does not support "Smart" Multiport Serial cards unless the manufacturer of these cards has added special support.

You cannot use these applications to modify the filesystem shared by DOS and the UNIX system.

You can use the commands and applications described above on a real DOS filesystem, such as a diskette drive, a physical or virtual DOS partition, or a virtual floppy.

If you try to use any of these commands or applications to modify the shared filesystem, DOS displays an error message and the operation fails, but your data and programs are not harmed in any way.

DOS clock interrupt rate

By default, SCO Merge uses the standard DOS clock interrupt rate of approximately 18 clock ticks per second. All DOS applications that work at this clock interrupt rate should work in the SCO Merge environment. SCO Merge has an option that can change the clock interrupt rate to one per second, which improves the performance of many applications. See the description of the `merge` command `set fastclk` option in Chapter 4. The slower clock interrupt rate is compatible with most, but not all, DOS applications.

Applications that continuously poll the keyboard

Unlike standard DOS, SCO Merge automatically puts most DOS applications that continuously poll the keyboard to sleep until there is keyboard input. SCO Merge has an option that can prevent this default behavior. See the description of the `merge` command `set pollsleep` option in Chapter 4.

Configuring applications that buffer keystrokes

Some DOS applications that buffer keystrokes may not work as expected in the SCO Merge environment. These applications include applications, such as SuperKey, that create keyboard macros by mapping multiple keystrokes to a single key.

You can correct this problem by putting the following line in a `config.sys` file that is interpreted when you run DOS:

```
device=\usr\sbin\ansi.sys
```

This approach has a side effect, however. To improve system response, SCO Merge, by default, puts DOS applications that poll the keyboard to sleep while they wait for keyboard input. When you include the `device=\usr\sbin\ansi.sys` line in your `config.sys` file, SCO Merge no longer puts applications that poll the keyboard to sleep. Your computer is therefore likely to be more heavily loaded, especially if you run multiple DOS applications at once.

Local and remote drives and file handles

Unlike standard DOS, SCO Merge treats drives that access the filesystem shared by DOS and the UNIX system (C:, D:, and J:) and any files contained in them as remote drives and remote files in a network. SCO Merge has options that cause DOS to interpret these drives and files as local. See the description of the `merge` command `set drive` and `set handle` options in Chapter 4.

Limits on number of open files

DOS applications may not open more than 123 files in the shared filesystem using the new-style open calls (that is, require the presence in the `config.sys` file of a `FILES=n` command, where *n* is greater than 123).

Protected mode restrictions

SCO Merge allows you to access the High Memory Area. In this sense, you manage DOS memory as you would on an AT-compatible computer. In all other important respects, however, the SCO Merge environment emulates a PC-compatible computer.

SCO Merge creates a DOS environment by invoking the *Virtual 8086* mode of the Intel 80386 or 80486 processor. Any application you run in such an environment must be compatible with the Intel 8086 processor. Any DOS application absolutely requiring an 80286 or later processor does not run under SCO Merge, except for Windows 3.1 and Windows 3.1 applications.

Despite the fact that the application is running on an 80386 processor, as a general rule, it cannot make use of CPU capabilities beyond those provided by the 8086. In particular, it cannot access the processor's Protected Mode for multitasking or its ability to address memory beyond the one-Mbyte DOS limit. Any DOS application attempting to do so will misbehave in some way. It may exit gracefully, hang the application or kill your DOS session.

Most DOS applications are meant for the 8086. Of those designed for more powerful processors, many are smart enough to sense the nature of the processor on which they are running and adjust their operation accordingly. For these "intelligent applications" you need to do nothing special. They will respond to the V86 environment as an 8086 processor and automatically run in the 8086-compatible "real mode".

For applications that offer 286/386 features and fail to adjust to the V86 automatically, there are several ways to compensate. See Chapter 3 for details.

In making your choices as to which DOS software to run under SCO Merge, consider the following factors:

- **The amount of memory used by the program is not the problem; it is the way that memory is accessed.** Programs that use Expanded Memory will work; programs that use Extended Memory will not. SCO Merge fully supports DOS-based Expanded Memory, using the LIM 4.0 interface. DOS programs that use Expanded Memory will work properly under SCO Merge. Extended Memory, however, is available only in the processor's 286/386 mode. Programs that use Extended Memory will not run under SCO Merge. (SCO Merge allows a certain limited access to the High Memory Area, but all other Extended Memory is unavailable.)
- **Multitasking per se is not restricted; the issue is 286/386 features.** Any application which successfully performs multitasking on a standard 8086 DOS computer should run properly under SCO Merge. Any application which uses the processor's Protected Mode for its multitasking will not.
- **Except for Windows 3.1, if the program does not run on a PC or XT, it will not run under SCO Merge.** Any program requiring an 80286 or higher AT-style processor cannot be used under SCO Merge. Consult the program documentation under "System Requirements."
- **You may not require the application at all.** Most applications using 286/386 features are aimed at multitasking or managing large amounts of memory. SCO UNIX is a full-fledged, multiuser, multitasking operating system, allowing each user to run many programs at once. With SCO Merge, this can include any combination of UNIX or DOS applications. The UNIX operating system can easily run very large applications. Many DOS programs have been ported and your application may be available for the UNIX operating system.

Passing UNIX environment variables to DOS

DOS and the UNIX system use similar concepts of environment variables (also called "environment strings"). An environment variable is a value (assigned by you, the operating system, or an application) that is available to all commands and applications that run in the same environment. Refer to your DOS documentation for further information on environment variables.

You can set DOS environment variables with the DOS **set** command. By default SCO Merge sets five DOS environment variables when you start DOS. To view their current values, type **set** at the DOS prompt. The output is similar to the following:

```
COMSPEC=C:\usr\ldbin\command.com
PATH=C:\BIN;C:\USR\BIN;C:\USR\DBIN;C:\USR\LDBIN;
LANG=En
TEMP=C:\TMP\
TMP=C:\TMP\
```

Values are assigned to these variables as follows:

- **COMSPEC** is set to the *command.com* file that is used with SCO Merge.
- **PATH** is set to match the UNIX path. Note that the UNIX path includes the standard directories for DOS programs, */usr/dbin* and */usr/ldbin*.
- **LANG** is set to be the same as the UNIX **LANG** environment variable.
- **TEMP** and **TMP** are set to *C:\tmp*, which is the preferred temporary directory.

DOS applications may assume that you explicitly assign the values of required environment variables yourself. The Microsoft C Compiler, for example, assumes that you have assigned a value to the environment variable called **INCLUDE**, which tells the compiler where to look for *#include* files if they are not found in the current directory. To assign the value "*\usr\ldbin\include*" to the environment variable **INCLUDE** on DOS drive C:, type:

```
set include=c:\usr\ldbin\include
```

DOS environment variables are typically set in the *autoexec.bat* file, so they are available for use by any command or application whenever you use DOS.

You can use DOS environment variables in the SCO Merge environment exactly as you would use them on a conventional DOS computer. SCO Merge, in addition, has special features for manipulating DOS environment variables that:

- Can override the default DOS search path automatically.
- Set other DOS environment variables so they are available to DOS programs run either in the DOS environment or from the UNIX shell.

These features improve DOS performance and flexibility.

Setting your DOS search path with DSPATH

Because SCO Merge automatically sets the search path used for DOS programs, it is rarely necessary for you to change it. You can change your UNIX search path at any time using the procedures described in your UNIX manuals.

You can, if necessary, reset the DOS search path in the DOS environment so it is different from your UNIX search path. You can reset your search path manually by using the DOS **path** command at the DOS prompt, or else override the default path in one of two ways:

1. Set your DOS search path with the **path** command in your home directory *autoexec.bat* file.
2. Use the UNIX environment variable **DSPATH** to communicate your preferred search path to DOS when you enter the DOS environment.

The second method (using **DSPATH**) is preferred because:

- It is more efficient than running *autoexec.bat*.
- Since *autoexec.bat* is not always executed when you start a DOS environment or run DOS programs from the UNIX shell, any search path specified in *autoexec.bat* may not be effective from the UNIX shell. The **DSPATH** value, on the other hand, is always used.

To set your preferred DOS search path using the **DSPATH** environment variable, follow these steps:²

1. At the UNIX system prompt, type **DSPATH=** followed by the desired DOS search path. You can use either the UNIX slash (/) or the DOS backslash (\) as the path separator. If the path includes backslashes or multiple directories separated by semicolons, enclose the path in single quotes (') to prevent the UNIX shell from interpreting the backslashes or semicolons. For example:

```
DSPATH='c:/usr/dbin;c:/usr/ldbin;c:/usr/mary/dbin;a:'
```

2. Export **DSPATH** as you would any UNIX environment variable, by typing:

```
export DSPATH
```

You can use any valid DOS drive designator (such as a:, c:, d:, e:, or j:) in the **DSPATH** definition. All directories in the shared filesystem that you include

2. The examples in this section describe setting variables in the UNIX system Bourne or Korn shell. If you are working in the C shell, use the **setenv** command and do not export the variable.

in your path definition must be specified using valid UNIX names, including correct use of upper- and lowercase letters. Use UNIX names rather than DOS mapped names for any directory names that are not legal in DOS. Invalid UNIX names are not included in the DOS search path.

You can include the **DOSPATH** definition and **export** instruction in your home-directory *.profile* file if you want your specified DOS path to take effect every time you log into the SCO Merge system.

The system administrator can include a default **DOSPATH** definition in */etc/profile* that affects all users. Users can override a system **DOSPATH** with their own **DOSPATH** definition or append additional paths using a command in the form:

```
DOSPATH=$DOSPATH';newpath'
export DOSPATH
```

where *newpath* is the list of additional directories the user wants searched.

Setting other DOS environment variables with DOSENV

Use the UNIX **DOSENV** environment variable to set any DOS environment variables except for **PATH** or **COMSPEC** from the UNIX shell.

To use **DOSENV**, follow these steps:

1. Set the required DOS environment variables using standard UNIX syntax:

```
VARIABLE1=VALUE1
VARIABLE2=VALUE2
```

```
.
```

2. Assign to the UNIX environment variable **DOSENV** the names of all the DOS environment variables that you defined in step 1. Use commas (but no spaces) to separate the variable names:

```
DOSENV=VARIABLE1,VARIABLE2,...
```

3. Export each of the defined DOS environment variables and **DOSENV**:

```
export VARIABLE1 VARIABLE2... DOSENV
```

Assume, for example, that you want to:

- Change your DOS prompt to show your current working drive and directory by using the DOS **PROMPT** environment variable.
- Assign the value *C:\usr\ldbin\include* to the environment variable **INCLUDE**, used by the Microsoft C Compiler.

To accomplish these operations, you would type:

```
PROMPT='$p$g'  
INCLUDE='c:\usr\ldbin\include'  
DOENV=PROMPT,INCLUDE  
export PROMPT INCLUDE DOENV
```

Observe the following concerning this example:

- Quotes are used around `$p$g` in the first line to prevent the UNIX shell from interpreting the metacharacter `"$"`. (For further information on the syntax for the `PROMPT` environment variable, refer to your DOS documentation on the `prompt` command.)
- In the second line, DOS-style backslashes are used instead of UNIX-style slashes because `DOENV` passes the values of all variables to DOS literally, with no translation. The string `c:\usr\ldbin\include` is surrounded by single quotes to prevent the UNIX shell from interpreting the metacharacter `"\"`.
- Items in the list of environment variables following `DOENV=` are separated by commas, with no spaces.

You can define DOS environment variables using `DOENV` in your home-directory `.profile` file to make them effective every time you log into SCO Merge. The system administrator can include `DOENV` definitions in `/etc/profile` that affect all users. Users can override a system `DOENV` with their own `DOENV` definition or append additional `DOENV` variables to the system definition with a command in the form:

```
DOENV=$DOENV,VARIABLE1,VARIABLE2,...
```

where `VARIABLE1`, `VARIABLE2`, and so on, are the user's own preferred `DOENV` definitions.

Running DOS or a DOS application as the default shell

When configured to support SCO Merge, your computer runs the standard UNIX System V operating system. The Bourne shell is the default shell when you log in. When you type:

```
dos
```

SCO Merge presents you with a continuous DOS environment — an alternative shell that understands DOS commands.

You can use the DOS shell or a specific DOS application immediately when you log in, without typing `dos` or the application name, by invoking `DOS` or the application in your home directory `.profile` file.

Running DOS as the default shell

To run DOS as the default shell, add the **dos** command to the end of your *.profile*, optionally followed by the **exit** command.

The **dos** command starts a DOS environment automatically as soon as you log in. You leave the DOS environment in the normal way — by typing **quit**. If the **dos** command is not followed with an **exit** command in your *.profile*, you return to the UNIX shell prompt when you type **quit**. You can then continue using the UNIX system or log out in the usual way. If the **exit** command follows the **dos** command in *.profile*, typing **quit** terminates your DOS environment and logs you out of the UNIX system.

Running a DOS application as the default shell

If you want a specific DOS application to start automatically when you log into your SCO Merge computer, follow procedures similar to those described in the previous section. For example, if you want to run Lotus 1-2-3 immediately upon login every time you use your computer, add the **123** command (instead of **dos**) to the end of your home directory *.profile*, optionally followed by the **exit** command. Then when you log in, Lotus 1-2-3 starts automatically. When you exit Lotus, you either return to the UNIX shell or are logged out, depending on whether you include the **exit** command in *.profile*.

Scheduling DOS programs

You can use the UNIX commands **at**, **batch**, **cron** and **nohup** with DOS processes just as you use them for UNIX processes. The **at** command runs DOS or UNIX commands at a later time that you specify. The **batch** command works similarly but executes your specified processes when the system load allows rather than at a time you specify. The **cron** command is also similar but is used for commands that are to be run regularly at a specified time. The **nohup** command starts a process that is immune to hangups and quits.

You can run any of these commands from the UNIX shell using standard UNIX syntax and including any valid options in the DOS command. For example, you could tell the Microsoft C Compiler to compile and link the file *big.c* at 8:00 P.M. by typing:

```
at 2000
cl big.c
<Ctrl>d
```

You do not need to be logged in at the time the command executes, as long as the system is running. For further information on these UNIX scheduling commands, consult your UNIX documentation.

When you use these scheduling commands with DOS programs, observe the following precautions:

- You cannot use these scheduling commands with display-oriented DOS programs. (See the description of the **tb** option in Chapter 6 for more on display- and stream-oriented programs.)
- Output is treated in the standard UNIX way for the scheduling command you use. If you do not redirect the output, **at**, **batch** and **cron** mail the output to you. The **nohup** command causes output to be directed to the file **nohup.out** unless you specify otherwise.
- If the DOS program requires a key disk, the key disk must be in the diskette drive at the time the program runs.
- You cannot send input from your keyboard to programs started with these scheduling commands. If the program accepts redirected input, you can redirect keyboard input from a file at the time you issue the command. For example:

```
at 2000
doit < kbdinput > output
(Ctrl)d
```

Appendix A

Using the system console and terminals

SCO Merge works from a wide variety of displays, including:

- The UNIX system console, which can use any of several different types of display adapters.
- PC scancode terminals, which emulate some of the features of the console.
- Serial terminals, including those that display only ASCII characters and those that are enhanced to display some or all of the full IBM PC character set used by DOS.

SCO Merge characteristics differ depending on the capabilities of your display. This appendix describes the features and limitations of SCO Merge when you use each type of display.

Characteristics of DOS on the console

SCO Merge works on any UNIX system console that consists of a monochrome or color monitor and a PC-compatible keyboard connected to one of the following display adapter boards:

- a Video Graphics Array (VGA) adapter
- a Color Graphics Adapter (CGA)
- a Hercules adapter
- a Monochrome Display Adapter (MDA)

In CGA, Hercules, EGA, and VGA modes, the console can display bit-mapped graphics. In CGA, EGA, and VGA modes, the console can display color.

Just as on a conventional DOS computer, you cannot run DOS applications that require a more powerful display adapter than you have in your console. For example, you cannot run a program that requires CGA graphics on an MDA card or run a program that requires full VGA capabilities on a CGA card. Note, however, that many DOS applications can sense the type of display adapter you have and adapt themselves to your hardware. For example, many DOS applications can display color and bit-mapped graphics when run on CGA or VGA cards and still run correctly (without displaying color) on an MDA card.

In general, SCO Merge automatically senses the type of display adapter your console uses and can run any DOS applications that are compatible with that adapter.

The console also supports the UNIX MultiScreen mechanism (also sometimes called *virtual terminals*), which means you can run multiple display-oriented DOS or UNIX processes at once, each in its own screen. Refer to Chapters 2 and 6 and to your UNIX documentation for more information on the MultiScreen feature.

Using EGA cards

EGA cards have several characteristics and optional configurations that do not apply to other display adapters. The following sections describe these characteristics and options.

Prerequisites to using EGA cards

You can run applications that use EGA video modes on your console only if you have an EGA card that is 100% compatible with the standard IBM EGA. Your SCO system must also have a DOS EGA image. There are many EGA cards available that are not 100% compatible with the IBM EGA. These cards do not work properly when you use them with applications that require an EGA card on the SCO system.

If your display uses an EGA card, be sure that the card is set up in pure EGA mode. Some EGA cards can be configured to emulate other display adapters or to switch into different modes automatically. EGA cards that are set up to emulate other cards or autoswitch between different modes may not work properly when you use SCO Merge.

Even if your computer has an incompatible EGA card, you can still run DOS applications. However, only monochrome and CGA modes are usable. When you use an incompatible EGA card, SCO Merge treats your display as a CGA display by default.

Consult your system administrator if you are uncertain whether your the SCO system includes an IBM-compatible EGA card and has a DOS EGA image.

Using EGA “fast” and “reliable” modes

When you run a DOS application on an IBM-compatible EGA card, SCO Merge can run the application in *fast* EGA mode or in *reliable* EGA mode. SCO Merge uses fast EGA mode by default.

Performance in fast EGA mode is faster than in reliable EGA mode. In fast EGA mode, however, your display can become corrupted when you use the UNIX MultiScreen feature to switch to another screen and then switch back to your DOS application. More specifically, your DOS screen may be corrupted under the following circumstances:

- The DOS application is in the middle of switching video modes at the moment you switch screens.
- The DOS application is actively drawing graphics on your display when you switch screens.
- The DOS application is writing directly to the EGA video registers when you switch screens.

If you do not use the UNIX MultiScreen feature while running DOS, you do not need to be concerned about these limitations. If you do use the UNIX MultiScreen feature, you can minimize the risk of corrupting your display by switching screens only when your application is not changing the DOS screen. You can also completely avoid these limitations by using the SCO Merge reliable EGA option.

Reliable EGA mode avoids the limitations of fast EGA mode, but it causes DOS to run slower. To use reliable EGA mode, start DOS with the **+arega** option. For example, to start the DOS environment in reliable EGA mode, type the command:

```
dos +arega
```

The DOS environment starts and the DOS C: prompt appears as usual, and you can run any DOS commands or applications.

You can also start a specific application in reliable EGA mode from the UNIX shell. For example, to start an application called **colorama**, you could issue the command:

```
dos +arega colorama
```

Chapter 5 describes how you can use the **dosopt** command to configure permanently the DOS environment or specific DOS applications to use reliable EGA mode.

DOS processes that suspend when their screens are not visible

When you run a DOS application that requires high-resolution graphics mode on an EGA card, SCO Merge suspends the application if you use the UNIX MultiScreen feature to switch from the DOS screen to another screen. When you switch back to the DOS screen, the application resumes execution from the point where it was suspended.

This behavior is the same whether you use fast EGA mode or reliable EGA mode.

Using VGA cards

SCO Merge is compatible with IBM VGA cards and can also run DOS applications correctly on many VGA cards made by other manufacturers. By default, SCO Merge can use only the standard IBM video modes even if you have a VGA card that has nonstandard (extended) video modes. Your system administrator can create a new DOS image that enables you to use your VGA card's extended modes. See Chapters 2 and 6 or consult your system administrator for further information.

If your display uses an VGA card, be sure that the card is set up in pure VGA mode. Some VGA cards can be configured to emulate other display adapters or to switch into different modes automatically. VGA cards that are set up to emulate other cards or autoswitch between different modes may not work properly when you use SCO Merge.

When you run a DOS application that requires high-resolution graphics mode on an VGA card, SCO Merge suspends the application if you use the UNIX MultiScreen feature to switch from the DOS screen to another screen. When you switch back to the DOS screen, the application resumes execution from the point where it was suspended.

Using PC scancode terminals

PC scancode terminals emulate the keyboard layout and 25-line monochrome screen display of a standard PC and generate PC scancodes rather than ASCII codes. These terminals are connected to the SCO Merge system via the UNIX serial driver. PC scancode terminals include Wyse Technology's WY-60 and Kimtron Corporation's Model KT-70/PC. You can run any monochrome DOS applications on a PC scancode terminal with the following limitations:

- You cannot run programs that display bit-mapped graphics.

- The UNIX MultiScreen (virtual terminal) feature is not available. When you start a DOS environment, you can return to the UNIX environment only by exiting from DOS (by typing **quit**, for example) or by forking a new UNIX shell.

Using serial terminals

Serial terminals have the same limitations as PC scancode terminals. In addition, they typically have the following limitations:

- They can display only 24 of the 25 lines that DOS normally displays on a personal computer. The SCO Merge **shift screen** command allows you to see either the top or bottom 24 lines.
- Many serial terminals cannot display the full IBM PC character set used by DOS. SCO Merge maps characters that cannot be displayed to similar ASCII characters or to a period (.).

Some serial terminals can display the entire IBM PC character set. If you use the correct **TERM** definition for these terminals, SCO Merge displays all PC characters as they would be displayed on a conventional DOS computer. (Refer to the *Multiuser Guide* for further information on setting up serial terminals.)

- The keyboards on most serial terminals are missing important keys such as <Alt>, <PgUp>, and <PgDn> that exist on PC keyboards. SCO Merge lets you simulate these keys with combinations of standard ASCII keyboard keys. This release of SCO Merge defines key sequences for the complete standard 101- and 102-key PC keyboards.

Serial terminal keyboard mapping

PC keyboards contain keys not commonly available on standard serial terminals. Because DOS applications assume the presence of these keys, SCO Merge supplies a method of simulating the missing PC keys using an ASCII keyboard. The following table shows the escape sequences that simulate these keystrokes on serial terminals.

A. Using the system console and terminals

PC Key to Simulate	Escape Sequence
The 12 Function Keys	
⟨F1⟩	⟨Esc⟩ 1
⟨F2⟩	⟨Esc⟩ 2
⟨F3⟩	⟨Esc⟩ 3
⟨F4⟩	⟨Esc⟩ 4
⟨F5⟩	⟨Esc⟩ 5
⟨F6⟩	⟨Esc⟩ 6
⟨F7⟩	⟨Esc⟩ 7
⟨F8⟩	⟨Esc⟩ 8
⟨F9⟩	⟨Esc⟩ 9
⟨F10⟩	⟨Esc⟩ 0
⟨F11⟩	⟨Esc⟩ ;
⟨F12⟩	⟨Esc⟩ :
The Numeric Keypad	
4/←	⟨Esc⟩ f
6/→	⟨Esc⟩ g
8/↑	⟨Esc⟩ t
2/↓	⟨Esc⟩ v
7/⟨Home⟩	⟨Esc⟩ h
9/⟨PgUp⟩	⟨Esc⟩ p
3/⟨PgDn⟩	⟨Esc⟩ q
0/⟨Ins⟩	⟨Esc⟩ i
. /⟨Del⟩	⟨Esc⟩ d
1/⟨Enter⟩	⟨Esc⟩ e
5	⟨Esc⟩ .
⟨Num Lock⟩	⟨Esc⟩ n
/	⟨Esc⟩ &
*	⟨Esc⟩ *
-	⟨Esc⟩ -
+	⟨Esc⟩ +
⟨Enter⟩	⟨Esc⟩ ^
Shift, Control, and Alt Keys	
⟨Left Shift⟩	⟨Esc⟩ z
⟨Right Shift⟩	⟨Esc⟩ /
⟨Left Ctrl⟩	⟨Esc⟩ c
⟨Right Ctrl⟩	⟨Esc⟩ C
⟨Left Alt⟩	⟨Esc⟩ a
⟨Right Alt⟩	⟨Esc⟩ A

PC Key to Simulate	Escape Sequence
Other Special Keys	
⟨PrtSc⟩/⟨SysReq⟩	⟨Esc⟩ r
⟨Scroll Lock⟩	⟨Esc⟩ s
⟨Pause⟩/⟨Break⟩	⟨Esc⟩ l
⟨CapsLock⟩	⟨Esc⟩ k
⟨SysReq⟩	⟨Esc⟩ =
⟨Esc⟩	⟨Esc⟩ ⟨Esc⟩
102nd key on international keyboards	⟨Esc⟩ <
10 Cursor Keypad Keys	
⟨Ins⟩	⟨Esc⟩ I
⟨Home⟩	⟨Esc⟩ H
⟨PgUp⟩	⟨Esc⟩ P
⟨Del⟩	⟨Esc⟩ D
⟨Enter⟩	⟨Esc⟩ E
⟨PgDn⟩	⟨Esc⟩ Q
↑	⟨Esc⟩ T
←	⟨Esc⟩ F
↓	⟨Esc⟩ V
→	⟨Esc⟩ G
Special Function Escape Sequence	
All shifts released	⟨Esc⟩ ⟨Space⟩

To simulate PC keys listed in the left column of the chart, press keys listed in the "Escape Sequence" column of the chart. For example, to simulate the ⟨Home⟩ key on an ASCII keyboard, press:

⟨Esc⟩ h

To simulate the ⟨F1⟩ function key on an ASCII keyboard, press:

⟨Esc⟩ 1

In a DOS environment, the PC keyboard's ⟨Ctrl⟩, ⟨Alt⟩, and ⟨Ctrl⟩ keys are usually used *in conjunction with* another keyboard character. To simulate these multiple-keystroke key combinations on a serial terminal, enter the escape sequence for the first key (⟨Esc⟩ c, ⟨Esc⟩ a, or ⟨Esc⟩ z), and then enter the escape sequence for the second key.

For example, to simulate the <Ctrl> <F1> key sequence, you would first press:

<Esc> z

to simulate the <Left Shift> key; then press:

<Esc> 1

to simulate the <F1> function key.

The <Ctrl>, <Alt>, and <Ctrl> key simulations all work in this manner. The escape sequences for these three keys send the “key down” (make) code to DOS. From the point of view of DOS, the key you are simulating remains down until you enter a keystroke that *does not* simulate <Ctrl>, <Alt>, or <Ctrl>.

For example, to simulate the common <Ctrl><Alt> reboot key combination from an ASCII keyboard, enter:

<Esc> c <Esc> a <Esc> d

In this example, <Esc> c simulates a depressed *and held* <Ctrl> key, <Esc> a simulates a depressed *and held* <Alt> key, and <Esc> d simulates the key. Since <Esc> d is the first key combination in this example that does *not* simulate <Ctrl>, <Alt>, or <Ctrl>, SCO Merge sends to DOS both the simulation of the key *and* the “key up” (break) code for all three keys (<Ctrl>, <Alt>, and).

If the key combination you want to simulate ends with <Ctrl>, <Alt>, or <Ctrl>, you must end it with the <Esc> <Space> sequence, which sends the key up (break) code for any <Ctrl>, <Alt>, or <Ctrl> keystrokes currently in effect. For example, to simulate the <Ctrl><Alt> key combination used to access the Sidekick main menu, enter:

<Esc> c <Esc> a <Esc> <Space>

The escape sequence <Esc> <Esc> sends a real DOS escape character from an ASCII keyboard.

NOTE If your serial terminal has programmable function keys, you may set them up to generate single- or multiple-keystroke escape sequences that you use frequently. See the manual that accompanies the terminal for procedures.

In some cases, the function keys and arrow keys on a serial terminal are mapped to generate the proper PC-specific scancodes. This depends on the TERM setting for the type of terminal used.

Console and terminal control codes

Key combinations called *control codes* allow you to perform several useful functions on your console, PC scancode terminal, or ASCII terminal. These functions are necessary and work only when you are in the DOS environment or running a DOS program. The following chart lists the key combinations you use to perform these functions:

Function	ASCII Terminal Control Keys	PC Scancode Terminal Control Keys	Console Control Keys
KILL DOS	⟨Esc⟩ ⟨Ctrl⟩K	⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩K	⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩K
DOS DEVICE INFO	⟨Esc⟩ ⟨Ctrl⟩i	⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩-i	⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩i
FORK SHELL	⟨Esc⟩ ⟨Ctrl⟩F	⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩-F	N/A
SHIFT SCREEN	⟨Esc⟩ ⟨Ctrl⟩U	⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩U	N/A
REDRAW SCREEN	⟨Esc⟩ ⟨Ctrl⟩R	⟨Ctrl⟩⟨Esc⟩ ⟨Ctrl⟩R	N/A

Use these control codes as follows:

- | | |
|-----------------|---|
| KILL DOS | If a DOS process becomes irrecoverably hung (for example, when ⟨Ctrl⟩⟨Break⟩ and ⟨Ctrl⟩⟨Alt⟩ ⟨Del⟩ fail), use this key combination to kill DOS. |
| DOS DEVICE INFO | When you run many different DOS environments, it is easy to forget which devices are attached. The <i>DOS Device Information Popup Window</i> displays all the devices available to your current DOS session. |
| FORK SHELL | <p>FORK SHELL starts a new UNIX shell from the DOS environment or a DOS application, but it does not use the standard UNIX MultiScreen (virtual terminal) mechanism. You therefore cannot use the standard UNIX MultiScreen key sequences to switch between your DOS session and UNIX shell. You can, however, start any UNIX or DOS process—including a new DOS environment—in the UNIX shell. Terminate the UNIX shell and return to your original DOS session just as you would a UNIX shell started from a UNIX prompt: press ⟨Ctrl⟩D or type exit.</p> <p>The FORK SHELL function is not available on the console.</p> |
| SHIFT SCREEN | On a 24-line display, this control code causes the screen to shift from displaying the lower 24 lines (the default) to displaying the upper 24 lines. Pressing this control |

code again shifts the screen back to the default display. This control code has no effect on a 25-line display or PC scancode display.

REDRAW SCREEN If the DOS display is overwritten or corrupted, use this control code to redraw the screen. This control code has no effect on the console.

Note that on some serial terminals, the key sequence `<Ctrl><Esc>` accomplishes the same thing as just pressing `<Esc>`. Therefore, you can use the same `<Ctrl><Esc>` sequence on these ASCII terminals to introduce terminal control codes as you do on PC scancode terminals and the console.

Appendix B

Terminfo guide

This appendix is divided into the following sections:

- “Basic concepts” describes the **terminfo** file, the terminals it serves, and the special key sequences used to activate its services.
- “Output on standard terminals” and “Input on standard terminals” describe the **terminfo** items used for standard terminals.
- “About PC scancode terminals” discusses the process of switching between standard mode and PC scancode mode on a PC terminal.
- “Output on PC scancode terminals” and “Input on PC scancode terminals” describe some **terminfo** items that allow SCO Merge to take advantage of the extra capabilities of PC scancode mode.
- “PC-character-to-ASCII conversions” contains the PC-character-to-7-bit-ASCII conversions that are built into SCO Merge.
- “**terminfo** escape sequences” displays the **terminfo** key definitions and the standard escape sequences that SCO Merge uses to simulate a 101/102-key PC keyboard.

The primary requirement for DOS to run on a terminal is a good **terminfo** description. Therefore, most of this appendix describes the standard **terminfo** items used by SCO Merge. In addition, there are special **terminfo** items used by SCO Merge to enhance the use of DOS on a terminal.

This appendix is not a tutorial on how to construct **terminfo** descriptions for standard terminals; for that, refer to the appropriate UNIX documentation. We recommend that you study the **terminfo(M)** man page for information about the structure and use of **terminfo** source files.

Basic concepts

terminfo

terminfo is the UNIX terminal description utility. It includes a set of descriptions of terminal capabilities and a set of routines that use those descriptions. **terminfo** files are compiled databases that describe the capabilities of the terminals attached to the system and how those terminals perform their various operations.

This appendix describes how SCO Merge uses **terminfo** entries to run DOS sessions on serial terminals. It also presents various ways of enhancing your existing **terminfo** files for better DOS support.

DOS programs expect their keyboard input in the form of PC scancodes and write characters to the screen in a PC character set. Serial terminals use entirely different codes for their keyboard input and their screen output. Therefore, running DOS on serial terminals requires several conversions. The terminal's keyboard input must be converted to the PC scancodes that DOS programs expect. The PC character set output by the DOS program must be converted to whatever code sequences are used to position and display characters on that particular terminal's screen.

SCO Merge uses two processes that do this conversion. The **tkbd** process converts the input and the **crt** process converts the output. These processes use the **terminfo** descriptions for the terminal to do their processing. Besides hardware and the standard components already provided by the SCO Merge system, the only additional item you need to run DOS on a terminal is a good, complete **terminfo** description.

terminfo files

Each **terminfo** file is a database of terminal capability descriptions. Conceptually, the descriptions can be split into two categories: *terminal input* and *terminal output*.

terminfo descriptions use two methods of describing each terminal capability. *Variable* is the name the programmer uses to access the **terminfo** capability. *Capname* (capability name) is a shortened name used in the text of the database. It is used primarily by people updating the database.

For terminal input, the database describes how keyboard input is to be interpreted. These descriptions are called *key definitions* in this appendix and have **terminfo** variable names beginning with **key_**. For example, the **terminfo** **key_f1** string (capname **kf1**) holds the characters that are generated when function key <F1> has been pressed.

For terminal output, the database describes how to display information on the screen. For example, it describes how to display a character in reverse video.

In addition, **terminfo** also describes some special commands to control the operation of the terminal. One example is a command to turn on a special scrolling mode.

Standard and PC scancode terminals

There are two basic varieties of serial terminals, *standard* terminals and *PC scancode* terminals.

Standard terminals are the type normally used with UNIX systems. These standard terminals display several different character sets. Some display only 7-bit ASCII characters. Others display 8-bit character sets, such as the ISO Latin-1 character set. Some terminals can display multiple character sets (though usually only one at a time).

Note that the 7-bit ASCII character set is a subset of most character sets used by standard terminals, including PC character sets. The principal characteristic of a “standard” terminal is that the keyboard sends the same character codes that the terminal displays (or a subset). For example, on an ASCII terminal, when you press the “A” key, the ASCII code for “A” is transmitted. Shift keys, control keys, or *meta* keys do not send codes. They merely modify the code that is sent when a regular key is pressed. The code is sent only when the key is pressed. No code is sent when the key is released.

Every PC scancode terminal is based on a standard terminal. Thus they have many of the same qualities, with the PC feature as a special mode of the terminal. When in PC scancode mode, the terminal sends PC scancodes instead of the normal keyboard codes. A scancode is sent when a key is pressed, and another is sent when the key is released. All PC scancode terminals can display the entire PC character set, at least when in PC scancode mode.

The PC scancode mode is turned on and off with a command sent from the server system to the terminal. It is usually the responsibility of the program controlling the terminal (such as SCO Merge) to put the terminal into scancode mode when starting and put it back into standard mode when finished.

Note that a terminal can have the ability to display the PC character set but still not be considered a “PC terminal.” A “PC terminal” must be able to send PC scancodes as well as display a PC character set. This is why they are called *PC scancode* terminals.

Escape sequences

An *escape sequence* is a pattern of bytes used to send some sort of special command. Originally the term referred to sending the ASCII escape character followed by a byte sequence used as a command addressed to hardware. This usage has broadened over time. Current usage includes special command sequences sent either by programs, or from the keyboard, by users.

The SCO Merge implementation of the escape sequence refers to a pattern of key presses trapped by SCO Merge before it reaches the user's application program. The application may or may not be terminated as a result. SCO Merge may act upon the key sequence itself or pass it on to the DOS program as keyboard input.

The escape sequence is so called because the first key is often <Esc>. This initial keystroke is configurable, but even if it has been changed, these special patterns of keystrokes are still referred to as "escape sequences." SCO Merge uses escape sequences for two purposes: key emulation and special actions.

DOS programs expect input from certain special DOS keys that may not be physically present on a particular terminal's keyboard. Function keys and arrow keys are examples. If the key does not physically exist, you can emulate that key's action for DOS by pressing a sequence of keys that do exist. SCO Merge intercepts this pattern and translates it into the codes expected by the DOS program. This is referred to as a *key emulation escape sequence*. It is usually <Esc> plus another key.

While running a DOS program, you sometimes need to communicate directly with SCO Merge itself, to request some special action. You may want the screen redrawn or to call up the DOS Device Information Popup Window. SCO Merge provides special escape sequences for this, and this appendix calls them *special action escape sequences*.

The SCO Merge lead-in sequence

The *lead-in sequence* is one or more keystrokes that initiate an escape sequence. SCO Merge uses both one- and two-character lead-ins. In some computing environments, the lead-in is the <Esc> key alone. In others it is <Ctrl><Esc>. The lead-in sequence is configurable, and you can specify any keys you find convenient.

Escape sequences used by SCO Merge

ASCII and ISO terminal keyboards lack some PC keys, so key emulation sequences are needed for those terminals. The default setup emulates missing DOS keys with <Esc> plus a character. The default sequence for special actions is <Esc> plus a control character.

PC scancode terminals have all the PC keys, so key emulation sequences are not needed. They do, however, need to send requests for special actions to SCO Merge and they do this with a two-key lead-in plus another key. The default is (Ctrl)(Esc) plus a control character.

SCO Merge intercepts all keyboard input from the terminal. Most of it is passed on to the current DOS program as input. The SCO Merge lead-in character triggers two different responses, based upon what comes next.

The lead-in character plus a standard ASCII character generally signals a key emulation. The codes for the emulated key are passed on to the DOS program as input. The lead-in sequence plus a control character ((Ctrl) held down while another key is pressed) generally signals a special action. This sequence is acted upon by SCO Merge itself.

Output on standard terminals

This section discusses **terminfo** items that SCO Merge uses to display DOS characters on serial terminals.

The terminal must have the following minimum capabilities to support DOS:

Variable name	Capname
cursor_address	cup
clear_screen	clear
clr_eol	el

This value must also be set:

Variable name	Capname
lines	lines

Screen scrolling must be available. The first two capabilities listed below represent primitive, line-oriented capabilities that can be used if they are all that exist. The next three capabilities provide much better facilities. One group or the other must exist for DOS to use the terminal.

Variable name	Capname
insert_line	il1
delete_line	dl1

Variable name	Capname
change_scroll_region	csr
scroll_forward	ind
scroll_reverse	ri

The following capabilities are used if they exist but are not required:

Variable name	Capname
cursor_invisible	civis
cursor_normal	cnorm
cursor_up	cuul
cursor_down	cud1
cursor_left	cub1
cursor_right	cuf1
enter_ca_mode	smcup
exit_ca_mode	rncup
enter_am_mode	smam
exit_am_mode	rmam
enter_pc_charset_mode	smpch
exit_pc_charset_mode	rmpch

All but two of the **terminfo** items listed above are explained in your UNIX **terminfo** documentation. The **smpch** (`enter_pc_charset_mode`) and **rmpch** (`exit_pc_charset_mode`) capabilities are special command strings. They set the modes of the terminal when starting and finishing a DOS task. For example, if the terminal has special commands to enable and disable a PC character set, those commands would be put in **smpch** and **rmpch**. Other commands can also be put in these strings to be executed when starting and finishing a DOS task. Examples of this are commands that change the cursor shape to an underscore when starting DOS and change it back to normal when exiting DOS. Refer to “About PC scancode terminals” for more information about PC scancode mode terminals.

NOTE Commands that actually switch a terminal into and out of PC scancode mode *must not* be put into **smpch** and **rmpch**. There are separate commands for that. See **smcsc** and **rmcsc** in “About PC scancode terminals.”

The following **terminfo** commands can be executed (if they exist) when starting and ending a DOS session. They are listed in the order that they are used.

Variable Name	Capname
Starting:	
enter_pc_charset_mode	smpch ¹
exit_am_mode	rmam
enter_ca_mode	smcup
clear_screen	clear
set_attributes	sgr
Ending:	
change_scroll_region	csr
cursor_address	cup
clr_eol	el
exit_ca_mode	rmcup
enter_am_mode	smam
exit_attribute_mode	sgr0
cursor_normal	vnorm
exit_pc_charset_mode	rmppch ¹

Other important settings are those that describe the sequences for controlling the various display modes, such as Bold, Reverse, Underline, and Blink. The following is a list of the **terminfo** display mode commands used when displaying a DOS session. These are not required, but, if they are available, they make the display look much more like a real DOS console screen. Of these, **sgr** (set_attributes) is the most important. In fact, a well-written **sgr** entry can make all the difference between a good DOS display and a substandard one.

Variable Name	Capname
set_attributes	sgr
enter_reverse_mode	rev
enter_secure_mode	invis
enter_underline_mode	smul
enter_bold_mode	bold
enter_blink_mode	blink
exit_attribute_mode	sgr0

Note that real PC monochrome text has more display modes than some terminals. Some terminals have only one brightness level. Others have brightness levels that do not match those of a PC monochrome display. (PC monochrome has "normal" and "high intensity" levels; some terminals have only "dim" and "full" brightness levels.) For this reason it is sometimes difficult to choose which terminal mode best conveys the information the DOS program is trying to communicate.

1. Remember that, despite their names, these commands are *not* used to switch the terminal into and out of the PC scancode mode. Refer to the descriptions of **smsc** and **rmsc** under "Switching Commands" in "About PC Scancode Terminals."

PC monochrome characters have four mutually exclusive display modes:

- Normal (light character on dark background)
- Reverse video (dark character on light background)
- Invisible
- Underline

In addition, the characters can be high-intensity or blinking. Thus there are 16 possible ways to display a character (including four invisible ones). The only way all these combinations can be correctly displayed is with the **sgr** capability. The **sgr** string is a parameterized string. **sgr** and parameterized strings are discussed in detail in the **terminfo(4)** manual page. Please refer to that document for more information.

Special terminfo descriptors

We have discussed standard **terminfo** descriptors, plus two special ones, **smprch** and **mpch**. There are some additional special **terminfo** descriptors used by SCO Merge to enhance the operation of DOS on terminals. The next two sections discuss two of the most important, **dispc** and **pctrm**. Additional special descriptors are discussed later, in the sections covering terminal input and the PC terminal.

Variable Name	Capname
pc_term_options	pctrm
display_pc_character	dispc

pctrm

The **pctrm** string is used to set special input and output processing options for SCO Merge. The output related options are:

- **LRC_OK**
- **ISO_TERM**
- **CS_SW_INDEPENDENT**
- **PCMODECP**

LRC_OK is set when it is OK to write to the lower right corner of the display. Many terminals incorrectly scroll when that location is written to, even when the automatic scroll mode is turned off. Consequently, by default, SCO Merge does not write to this location. If the terminal does not have this problem, set **LRC_OK** in the **pctrm** string.

The NLS (National Language Support) features of SCO Merge provide support for special character sets. (The prime example is ISO 8859.) SCO Merge offers special facilities for terminals that can display these character sets. For this type of terminal, put **ISO_TERM** into the **pctrm** string.

When you find it inconvenient to modify the **terminfo** to set this option, you can set a UNIX environment variable **ISO_TERM** instead. The section "How PC characters are displayed on ISO terminals," later in this appendix, provides details on how this is done.

When an ISO terminal has a special mode that enables it to display a PC character set, the options **CS_SW_INDEPENDENT** and **PCMODECP** can be used. The section "Combined ISO and PC character set terminals," later in this appendix, discusses these two special options.

Any combination of options is valid within the **pctrm** string; the options can be placed in any order. Every option in the **pctrm** string must be delimited by leading and trailing colons (:). The following examples show the valid combinations of **LRC_OK** and **ISO_TERM**.

```
pctrm=:LRC_OK:ISO_TERM;
pctrm=:ISO_TERM:LRC_OK;
pctrm=:ISO_TERM;
pctrm=:LRC_OK;
```

dispc

By default, SCO Merge uses an internal table to map each PC character to a "matching" ASCII character that the terminal can display. Many extended PC characters that have no good ASCII match are displayed simply as "***". See "PC-character-to-ASCII-conversions," at the end of this appendix, for these built-in conversions.

The **terminfo** string **dispc** can be used to override SCO Merge's built-in table. **dispc** can be used in one of two ways. It can either contain the name of an alternate conversion table (for example, **dispc=/usr/lib/dispc/myterminal**), or it can be a parameterized string that describes the *rules* for displaying PC characters.

If you have a serial terminal capable of displaying *some* of the PC characters (but not all), or if you want to redefine some of the built-in character conversions, the first form of **dispc** suits your needs best. On the other hand, for a PC scancode terminal or any terminal that can display most of the PC characters, the second form of **dispc** is more efficient.

The next section describes the first form of **dispc**. The section “Output on PC scancode terminals” discusses the parameterized form of **dispc**.

How PC characters are displayed on terminals

The first form of **dispc** string is used to specify a file containing a PC-character-to-terminal-code conversion table. This is known as a PCT file.

The PCT file is a normal UNIX text file, with each line defining a single PC-character-to-terminal-character conversion. In a PCT file each line is composed of fields, separated by space or tab characters. Any field that starts with “/” starts a comment. In addition, lines that have a “#” as the first character in the first field are comments. All comments end at the end of the line. Each line can have a maximum of eight usable fields.

The first field is the PC character code to be displayed. The following fields (up to seven) are the codes to be sent to the terminal to display this character. The order of the lines is not important. If a particular PC character conversion is not specified in the file, the built-in conversion is used for that PC character. Each field can be specified in four ways: it can be a 7-bit ASCII character, a decimal number, a hex number, or an octal number.

A decimal number is represented as a number 0 to 255 with a period at the end (for example “129.”). A hex number is represented as a number 0 to FF with an “h” at the end (for example “A8h”). An octal number is represented as a number 0 to 377 with an “o” at the end (for example “267o”). Here are a few sample PCT lines:

```
z z          /* This converts 'z' to 'z'
              /* (no conversion needed).
elh 33o 1. B /* el is the beta symbol, and the beta is
              /* displayed with an escape (33 octal)
              /* followed by binary '1' which in turn is
              /* followed by the ASCII character 'B'.
b3h 27. b3h  /* To display the b3 (vertical line)
              /* character, it must be preceded with an
              /* escape (27 decimal).
```

How PC characters are displayed on ISO terminals

SCO Merge supports NLS automatic character set conversion. Each character in the PC character set being used by the DOS session is converted to the closest character in the ISO character set. The ISO character is then displayed on the terminal.

SCO Merge automatically keeps track of the current PC code page of the DOS session. This allows it to convert from the current PC code page to the terminal’s character set, even when the PC code page is changed in the middle of a session.

Not all NLS character sets are supported. Factory-supplied NLS character set support includes (at a minimum) ISO 8859 (the Latin-1 alphabet), plus PC code pages 437 and 850. Other character sets may also be supported. Refer to your *SCO Merge Release and Installation Notes* for a full list of supported character sets.

Code set is the UNIX term for the character set your terminal uses. SCO Merge determines which code set your terminal uses via the **LANG** environment variable or by reading **/etc/default/lang** if it exists. For details about the **LANG** environment variable and instructions on how it is used to specify the code set, refer to Chapter 7 of this book and to your UNIX documentation.

SCO Merge automatically keeps track of the current PC code page. Provided that both your terminal's code set and the PC code page in use are supported, translation between the two is automatic.

For example, suppose the PC character to be displayed is code number 0x82 in code page 850 (é). SCO Merge translates this into the code number for "é" in the ISO 8859 character set your terminal can display.

This automatic character set conversion is enabled whenever **ISO_TERM** has been included in the **terminfo pctrm** string or the **ISO_TERM** environment variable is set. **ISO_TERM** does not need to be set to anything in particular. It merely needs to exist. From the Bourne or Korn shell, use the following command to set it:

```
ISO_TERM=
export ISO_TERM
```

When an exact conversion for a character is not possible, SCO Merge uses the closest match it can find in its own built-in conversion table. If a specific conversion has been specified with **dispc**, that transformation is used instead.

Combined ISO and PC character set terminals

As a very special case, SCO Merge also supports those terminals that display both an ISO code set and a PC code page. Whenever the DOS process is using a code page that matches the code page supported by the terminal, the terminal is switched to PC character set mode. SCO Merge sends the same output to the terminal it would send to any terminal that can display that PC character set.

When the DOS process switches to a PC code page other than the one the terminal supports, the terminal is switched to ISO mode. SCO Merge then sends the same screen output it would send to a normal ISO terminal.

There are several restrictions. The **smpch** and **rmpch** terminfo items must be specified. They must contain only the commands that switch the terminal from ISO code set display to the PC code page display. The terminal cannot change its mode in any respect other than the character set being displayed. It cannot use a PC scancode mode that changes how the keyboard operates, or changes how the character attributes are set. **dispc** must also be specified, so that SCO Merge knows how to display PC characters when in PC character mode.

By default, it is assumed that code page 437 is used by the terminal when in PC code page mode. If another code page is to be used, the string, **PCMODECP=pcnnn** is set in **pctrm**. This specifies that code page *nnn* is used by this terminal.

To enable ISO-to-PC-character-set switching, the **pctrm** string must contain both **ISO_TERM** and **CS_SW_INDEPENDENT**. The **PCMODECP=pcnnn** string is optional. For example, if code page 850 is to be used, **pctrm** would be set like this:

```
pctrm=:ISO_TERM:CS_SW_INDEPENDENT:PCMODECP=pc850;
```

When the DOS session is using a code page that matches the PC code page the terminal can display, the terminal is switched into that mode with **smpch**. **dispc** is used to display the characters. When switching from that code page to another, **rmpch** is used to switch the terminal back to ISO display mode.

Input on standard terminals

On standard terminals, the keyboard codes coming from the terminal are either 7-bit ASCII or 8-bit ISO 8859 (the Latin-1 character set). DOS programs assume that their input is coming from a PC keyboard and expect it in the form of PC scancodes. To emulate a DOS keyboard, SCO Merge translates serial terminal keyboard input codes into the scancode *make* and *break* signals sent by the corresponding keys on a PC keyboard.

SCO Merge uses internal tables to translate the ASCII or ISO input characters into the equivalent PC character. Then the PC characters are sent to DOS as PC keyboard make and break scancodes.

To translate the keyboard input, SCO Merge needs to know what character set is being used on the terminal. It gets this information from the UNIX environment variable **LANG**. (Refer to Chapter 7 for more information about the **LANG** environment variable and code sets.) SCO Merge keeps track of the current DOS code page character set and adjusts its translation from the UNIX code set as the DOS code page changes.

When the character code coming from the terminal corresponds to an unshifted PC keyboard key, the make and break for that key are simply delivered to the DOS process. For the "A" key, this would be scancodes 1E for the make and 9E for the break.

When the keyboard code coming from the terminal corresponds to a *shifted* PC key (either regular (Shift) or (Ctrl)(Shift)), more is required. The make and break signals for that key are preceded by a make of the appropriate (Shift)-type key and followed by a break of that (Shift) key. For (Ctrl)A, this would be four separate codes: 1D for the make of the (Ctrl) key, 1E for the make of the "A" key, 9E for the break of the "A" key, and 9D for the break of the (Ctrl) key. This behavior is called *auto-break*. Nothing special needs to be done to cause the key break to happen.

Where the DOS program requires input from PC keys that are not physically present on the terminal keyboard, SCO Merge provides key emulation escape sequences. For example, the sequence to simulate the (SysReq) key, is (Esc)=. You send keyboard codes equivalent to ASCII Esc and the ASCII equal sign. SCO Merge translates this into the make and break scancodes for the (SysReq) key and sends those codes to the DOS process.

The tables in Appendix A list the default key emulation escape sequences. All these keys are auto-break; when the sequence is received by SCO Merge, the make and break codes for the appropriate key are automatically sent to DOS.

The (Shift)-type keys ((Shift), (Ctrl), and (Alt) keys) are *semi-auto-break*. When received, the make code for the (Shift)-type key is sent. The break code is delayed until one of two things occurs:

- An auto-break occurs for another key.
- The release-all-shifts sequence is received.

The release-all-shifts sequence allows you to force the release of the (Shift)-type keys (Shift), (Ctrl), or (Alt). The default sequence is (Esc)(Space). This causes breaks for all made (Shift)-type keys, so that isolated (Shift)-type key actions may be simulated.

For example if you need to simulate the pressing and releasing of the left and right (Shift) keys, you would use this key-stroke sequence: (Esc), Z, (Esc), /, (Esc), (Space).

Default "key emulation" escape sequences

Each key emulation escape sequence is made up of two keyboard codes. By default the lead-in is the ASCII ESC code (hex 1b). This is followed by an ASCII character code. See "Serial terminal keyboard mapping" (page 147) for a list of the default key emulation escape sequences that SCO Merge uses to simulate DOS keys not physically present on the terminal keyboard.

Default “special action” escape sequences

In addition to the key emulation escape sequences, there are *special action escape sequences* that are intercepted by and acted on by SCO Merge itself. The following table lists these.

Press (Esc) plus	To Activate
[^] U	See top or bottom 24 lines
[^] F	Fork a new shell
[^] R	Redraw screen
[^] K	Kill DOS
[^] Z	Suspend DOS ²
[^] I	Display DOS Devices

Again, each special action escape sequence is a grouping of two codes: the lead-in sequence followed by a control-key combination. Remember that (Ctrl) is a (Shift)-type key. (Ctrl) plus a second key press is a single event. The (Ctrl)-*second-key* combination is sent as a single code.

The default lead-in on standard terminals is the ASCII ESC code (hex 1b). On PC scancode terminals, the default lead-in is (Ctrl)(Esc). The lead-in is followed by pressing (Ctrl) plus one of the letter keys listed in this table.

In the above table, [^]U stands for (Ctrl)U. Similarly, [^]F, [^]R, [^]K, [^]Z and [^]I stand for (Ctrl)F, (Ctrl)R, (Ctrl)K, (Ctrl)Z and (Ctrl)I, respectively.

Using terminfo descriptors for key emulation

We have already described the escape sequences built into SCO Merge for key emulation. The built-in key emulation escape sequences cover every PC key a DOS program expects and assume that the terminal has no such keys whatever.

Most terminals, however, have some of the same keys found on PC keyboards (such as the function and arrow keys). The code sequences they send when those keys are pressed are not standardized, but the keys are often physically present. Consequently, a well-constructed **terminfo** file for such a terminal already contains descriptors for whatever function and arrow keys the terminal has.

2. The “Suspend DOS” function is only available under a job control shell.

Whenever they exist, SCO Merge uses these pre-existing descriptors, in addition to its own built-in key emulation escape sequences. If you are writing or editing a **terminfo**, you can add or modify its entries in any way you find convenient, and SCO Merge uses those values in addition to its built-ins.

Thus, when working at a terminal equipped with a physical (F11) key, you can send (F11) to SCO Merge in two ways. You can press the physical (F11) key on the keyboard or activate the built-in sequence with (Esc);

SCO Merge monitors the code sequences sent by the terminal constantly. Whenever a code sequence matches either its own built-in key emulation escape sequence or one of the key descriptions in **terminfo**, SCO Merge sends the make and break codes for that key to the DOS process.

Refer to “terminfo escape sequences,” at the end of this appendix, for a comprehensive list of terminfo entries and how they can be mapped to PC keyboard keys. Wherever a PC key has no corresponding key on the terminal keyboard, and thus no corresponding entry in the **terminfo** file, an unused entry has been chosen to represent it. For example, the PC (SysReq) key has been mapped to the **key_options (kopt)** entry. Where no pre-existing entry bears any relationship to the PC key, an entry for an unused function key has been used. If you create or customize your own **terminfo** files, use “terminfo escape sequences” as a guide.

pctrm

If you want to configure the terminal so that it uses *only* the built-in key emulation escape sequences and ignores any existing **terminfo** key descriptions, use the **pctrm** string. We have discussed its output-related options. The one input-related option is **NO_TI**.

Whenever this keyword is in the **pctrm** string, SCO Merge ignores the key descriptors in **terminfo**. This is useful when there are key descriptors that conflict with the built-in SCO Merge escape sequences.

Sometimes it is easier to force SCO Merge to use only the built-in sequences than to make the user resort to some new escape sequence. For example, if the (F1) key on the terminal transmits (Esc)A, this conflicts with the built-in escape sequence for the right (Alt) key. A new escape sequence for the right (Alt) key could be put into **terminfo**, but users might be more comfortable using all the default escape sequences.

In this example the user would not press the (F1) key (or any of the special keys). The user would press (Esc)1 to simulate the (F1) key for DOS. All the other default escape sequences would also be used to simulate the special keys.

The following example shows how to set the `NO_TI` option.

```
pctrm=:NO_TI;
```

Changing the lead-in character

By default, the `(Esc)` key itself is used as the lead-in (first) character for the default escape sequences.³

Sometimes there are conflicts between the built-in escape sequences and other key combinations used by the terminal. When the terminal needs the `(Esc)` key for some other purpose, you can reconfigure the lead-in character. You specify another lead-in character with the terminfo `scesc` descriptor.

Variable Name	Capname
<code>scancode_escape</code>	<code>scesc</code>

For example, `scesc=Z` sets the lead-in sequence to `Z`. `scesc=^X` sets it to `(Ctrl)X`.

About PC scancode terminals

This section discusses the characteristics of PC terminals and how their special capabilities can be activated by the information in **terminfo**.

PC terminals are based on standard serial terminals, so most of the foregoing issues apply to PC terminals just as they do to standard terminals. PC terminals have a special mode, however. In PC mode they emulate a PC's Monochrome Display Adapter (MDA) and the PC scancode keyboard.

The keyboard has the same layout as a PC keyboard and can send PC keyboard scancodes. The screen can display all the same characters as an IBM-compatible MDA.

These terminals are usually in standard mode, acting as normal serial terminals for the UNIX system. When a DOS session is started, SCO Merge switches the terminal into the PC scancode mode. When the DOS session is finished, SCO Merge switches the terminal back to the standard mode. The following **terminfo** items are used when switching between standard and PC scancode modes.

3. Remember, an escape sequence must also be used for the `(Esc)` key itself. This sequence is `(Esc)(Esc)`. To send the `(Esc)` key to DOS, you must type it twice.

Variable Name	Capname
enter_scancode_mode	smisc
exit_scancode_mode	rmisc
user0	u0
user1	u1
user2	u2
user3	u3
pc_term_options	pctrm

Switching commands

The first two are the *switching commands*. Note the name similarity between these two commands and two commands described earlier. **smisc** and **rmisc** put the terminal into PC scancode mode and return it to standard mode. **smprch** and **rmprch**, described earlier, are command strings that specify the modes of the terminal when the DOS process starts and when it exits.

Alternate flow control

Items **u0**, **u1**, and **u2** control the so-called **XPC** feature. Many PC scancode terminals offer these alternate flow-control characters for PC scancode mode.

Standard terminals use two characters for flow control, normally referred to as XON and XOFF. On standard terminals, they are usually keyed with (Ctrl)Q and (Ctrl)S. These characters cannot be used in scancode mode, so alternate flow-control characters are needed.

The **u0** string defines these scancode mode XPC flow-control characters. The first character in the string is the XON character and the second is XOFF. The **u1** string is the command sent to the terminal to enable the XPC flow control. **u2** disables it. If the terminal automatically uses XPC flow control when put into scancode mode, **u1** and **u2** do not need to be specified.

For example, the typical XPC characters are 0x65 for XON and 0x67 for XOFF. Since the number 0x65 is ASCII "e" and 0x67 is ASCII "g", the **u0** specification that defines these XPC characters is:

u0=eg,

Serial line parameters

The **u3** string contains the command sent to the terminal to set the communication parameters: baud rate, number of bits, stop bits, and parity.

Whenever the terminal is in scancode mode, the serial line must be in 8-bit mode. Normally, the user sets up the serial line in such a way that it is always in 8-bit mode when using a PC terminal. In this case, the command is not needed.

The sole reason for this command is for those situations where the line has not been so configured. If the user forgets to put the line into 8-bit mode, the software can perform the service automatically. The line is placed in 8-bit mode when putting the terminal into scancode mode. The line is returned to 7-bit mode when the terminal exits scancode mode.

This is a **terminfo** parameterized string. For more information about parameterized strings, see the **terminfo(4)** manual page. The following table defines the parameters.

Parameter	Use
1st	Not used
2nd	Number of data bits (7 or 8)
3rd	Number of stop bits (1 or 2)
4rd	Parity (0=none, 1=odd, 2=even)
5th	Baud rate (1 thru 15 decimal)

Baud rate IDs 1 through 15 correspond to baud rates of 50 thru 38400. The baud IDs are drawn from the **CBAUD** field of **termio**. (They are defined in the file `/usr/include/sys/termio.h`). The following table defines the ID for each baud rate.

Baud	ID	Baud	ID	Baud	ID
50	1	300	7	9600	13
75	2	600	8	19200	14
110	3	1200	9	38400	15
134	4	1800	10		
150	5	2400	11		
200	6	4800	12		

It may seem surprising that baud rate is included in a single command with the other communication parameters. This is because some terminals have only a single command that sets all of these parameters at once. Any time you change any one parameter, all parameters, including the baud rate, must be specified.

pctrm

The **pctrm** string defines special options for either a standard terminal or a PC scancode terminal. They are activated by inserting specific terms in the **pctrm** string.

One special option relates to switching PC terminals between standard and PC scancode modes. This option is:

USE_TERM

This option is required when a terminal's PC scancode mode uses completely different commands than those used in standard mode.

One example is the WY-60 terminal. The Wyse **terminfo** descriptions for standard mode do not work when the terminal is in PC scancode mode. **USE_TERM** is employed to specify an alternate **terminfo** name for use when the terminal is in PC scancode mode. The syntax of this option is **USE_TERM=xxx** where **xxx** is the **terminfo** name to be used when in PC scancode mode. As with the other **pctrm** options, any combination and order of options is valid within the **pctrm** string. Every option in the **pctrm** string must be delimited by leading and trailing colons (:).

For example, suppose the **terminfo** name to be used when the Wyse is in PC scancode mode is **wy60-pc**. The **pctrm** string would be defined in the standard **terminfo** as:

```
pctrm=:USE_TERM=wy60-pc,
```

Sequence of use

The following table presents a list of the **terminfo** commands used (if they exist) when starting and ending a DOS session on a PC scancode terminal.

Variable Name	Capname
Starting:	
user3	u3
enter_scancode_mode	smisc
user1	u1
enter_pc_charset_mode	smprch
exit_am_mode	rmam
enter_ca_mode	smcup
clear_screen	clear
set_attributes	sgr
Ending:	
change_scroll_region	csr
cursor_address	cup
clr_eol	el
exit_ca_mode	rmcup

(Continued on next page)

(Continued)

Variable Name	Capname
enter_am_mode	smam
exit_attribute_mode	sgr0
cursor_normal	vnorm
exit_pc_charset_mode	rmpch
user2	u2
exit_scancode_mode	rmsc
user3	u3

SCO scancode terminal support

SCO UNIX has an API (Application Program Interface) that provides a higher level of control over the switching of PC scancode terminals to and from PC scancode mode.

When the SCO scancode API is available, SCO Merge uses it instead of switching the mode of the terminal itself. Rather than using **smsc**, **rmsc**, **u0**, **u1**, **u2**, or **u3**, SCO Merge uses the API to put the terminal into scancode mode. In all other respects, SCO Merge uses the terminal in scancode mode just as it would if the API did not exist.

The SCO scancode API uses **smsc** and **rmsc** to switch modes, just as SCO Merge does. The API does *not* use **u0**, **u1**, or **u2** to set alternate XON/XOFF, nor does it use **u3** or the **USE_TERM** option in **pctrm**.

The API also needs some additional descriptors. Refer to the documentation on this API for more information on how it works and any special support it might need in **terminfo**.

A single **terminfo** can serve both purposes. If you are writing a **terminfo** and you want it to work in all cases, whether the API is available or not, specify all of the above in the file. Include the descriptors needed by SCO Merge and those needed by the API. The appropriate items are used or bypassed, according to whether the API is present or absent.

Output on PC scancode terminals

This section discusses **terminfo** items used to display DOS characters on PC terminals.

In displaying characters on the screen, the only difference between standard serial terminals and PC terminals is the character set used. PC terminals use a PC character set; standard terminals use either the ASCII or the ISO 8859 character set. Typically, the PC character set used is based on DOS code page 437.

SCO Merge uses **terminfo** exactly the same way. The only difference lies in the **dispc terminfo** descriptor. This section discusses the form of **dispc** used for those terminals that can display the entire PC character set.

As previously discussed, the **dispc terminfo** descriptor is used to describe how PC characters are to be displayed on the terminal. There are two forms of usage of the **dispc** descriptor. The first is the filename of a PCT file, which contains information about how each character is to be displayed. This form *can* be used for PC terminals, but a second form is preferred.

The second form for **dispc** is a **terminfo** *parameterized string*; it defines a conversion from PC characters to terminal commands that display each character. This second form is preferred for several reasons. It is more efficient, self contained, and requires no external PCT file.

dispc has two forms because different methods work better on different terminal types. A parameterized string can be used for PC terminals because the rules for displaying PC characters are normally very easy and regular. Creating a parameterized string **dispc** for an ASCII terminal would normally be too complex for the limited size of a parameterized string.

Parameterized strings

A parameterized string is a small logic program that takes input, processes it, and produces output. It uses a *stack* to hold temporary values, handling it in the manner of a Reverse Polish Notation (postfix) calculator.

Logically, the program is called with a character as a parameter. It converts the input character to a string of zero or more output characters. The program can do arithmetic (addition, subtraction, multiplication, division, negation, and mod). It can also do logical operations (and, or, equal to, less than, greater than), and bit operations (and, or, xor, not). The logical operations are used in an *if-then-else* form that can be fully nested.

For more information about parameterized strings, see the **terminfo(M)** manual page.

dispc as a parameterized string

The one thing to note about using the **dispc** parameterized string is that the input character is the *second* parameter (%p2), not the first. (None of the other parameters are used internally by **dispc**.) This unusual usage is necessary because the **terminfo** library makes assumptions about the first parameters to parameterized strings that are not correct for what **dispc** needs to do.

The following example shows how to create a `dispc` parameterized string for the Kimtron KT70 PC terminal. The PC characters on the KT70 are displayed using these rules:

- For characters 00-1f and 7f (decimal 0-31,127), precede each with the ESC character.
- For characters 20-7e,80-ff (decimal 32-255, except 127), just output the character.

The parameterized string that encodes this rule is:

```
%?%p2%{32}%<%p2%{127}%=% | %t\E%p2%c%e%p2%c%;
```

This may seem confusing at first glance. Actually it is quite straightforward. The "%" character identifies operators. (For a full description of the operators see the `terminfo(M)` man page.) Here is a breakdown of the string:

Operator		Meaning
%?	means	"if"
%p2%{32}%<	means	"input character less than 32 decimal?"
%p2%{127}%=	means	"input character equal to 127 decimal?"
%	means	"logical or of the previous two tests"
%t	means	"then"
\E	means	"output the escape character"
%p2%c	means	"output the input character"
%e	means	"else"
%p2%c	means	"output the input character"
%;	means	"endif"

Creating a parameterized string for `dispc` is a two-step process. First, determine the *if-then-else* logic for displaying PC characters. Then translate it into the postfix parameterized string.

Code pages

The particular DOS code page (character set) displayed by the terminal when in PC mode is fixed. By default, it is assumed to be code page 437.

If the terminal uses another code page, put the `PCMODECP` option in the `pctrm` string. The syntax is `PCMODECP=pcnnn`, where *nnn* is the number of the code page used by the terminal. As with the other `pctrm` options, any combination and order of options is valid within the `pctrm` string. Each option is delimited by colons.

For example, if the code page is 850, then `pctrm` would be set like this:

```
pctrm=:PCMODECP=pc850;
```

Input on PC scancode terminals

Input on PC terminals is not at all complex. They send exactly the scancodes needed by the DOS session. Thus, for normal use of the keyboard, nothing special is needed in **terminfo** except the codes that switch the terminal into and out of PC scancode mode.

Nevertheless, there are some special action escape sequences. These are used for the same functions as previously described in “Input on standard terminals”—redrawing the screen, forking a new shell, terminating or suspending DOS, shifting the screen, and displaying DOS devices.

The default lead-in is the two-key sequence, (Left-Ctrl)(Esc), that is, pressing the (Esc) key while the left (Ctrl) key is pressed. Then you release those keys. This lead-in is followed by pressing (Ctrl) and another character. The second sequence determines which function is to be done.

An alternative is to press and hold the left (Ctrl) key. While holding it down, press (Esc), let it up, and press another key.

Press (Left-Ctrl) (Esc) plus	To cause this action
<code>^R</code>	Redraw the screen.
<code>^F</code>	Fork a new shell.
<code>^K</code>	Kill DOS.
<code>^U</code>	See top or bottom 24 lines (when the display is not a 25-line display).
<code>^Z</code>	Suspend DOS (on systems that support suspending processes).
<code>^I</code>	Display DOS devices.

For example, to kill DOS, you press the left (Ctrl) key and hold it. While keeping it down, press and release the (Esc). Still holding the left (Ctrl), press and release the “K” key. You can also release and re-press the left (Ctrl) key between the (Esc) and the “K” key. Either method works.

Changing the lead-in sequence

Using **terminfo**, the (Left-Ctrl)(Esc) lead-in for the sequence can be changed to any other key combination. The **terminfo** `scesc` descriptor is used to specify a different initial key.

Variable Name	Capname
<code>scancode_escape</code>	<code>scesc</code>

The **scesc** string must have at least one character. The first is the value of the scancode for the key. The following characters are the scancode values of the **<Shift>**-type keys that must be pressed at the same time.

Shift-type Key	Octal Sequence
<Left Ctrl>	35
<Right Ctrl>	340 35
<Left Alt>	70
<Right Alt>	340 70
<Left Shift>	52
<Right Shift>	66
E0-<Left Shift>	340 52
E0-<Right Shift>	340 66

For example, suppose the lead-in is to be the **<F12>** function key (scancode value octal 130), with the left **<Shift>** and the left **<Ctrl>** keys held down. For this **<Ctrl><Shift><F12>** lead-in, **scesc** would be set like this:

scesc=\130\035\052

PC-character-to-ASCII conversions

The following table shows the built-in PC-character-to-7-bit-ASCII conversions used to display PC characters on a serial terminal. The first field is the PC character number (in hex). The second is the 7-bit ASCII character to be displayed.

00h	*	20h	SP	40h	at	60h	'	80h	c	A0h	a	C0h	+	E0h	*
01h	*	21h	!	41h	A	61h	a	81h	u	A1h	i	C1h	+	E1h	*
02h	*	22h	"	42h	B	62h	b	82h	e	A2h	o	C2h	+	E2h	*
03h	*	23h	#	43h	C	63h	c	83h	a	A3h	u	C3h	+	E3h	*
04h	*	24h	\$	44h	D	64h	d	84h	a	A4h	n	C4h	-	E4h	*
05h	*	25h	%	45h	E	65h	e	85h	a	A5h	N	C5h	+	E5h	*
06h	*	26h	&	46h	F	66h	f	86h	a	A6h	a	C6h	+	E6h	*
07h	*	27h	'	47h	G	67h	g	87h	c	A7h	o	C7h	+	E7h	*
08h	*	28h	(	48h	H	68h	h	88h	e	A8h	?	C8h	+	E8h	*
09h	*	29h	)	49h	I	69h	i	89h	e	A9h	-	C9h	+	E9h	*
0Ah	*	2Ah	*	4Ah	J	6Ah	j	8Ah	e	AAh	-	CAh	+	EAh	*
0Bh	*	2Bh	+	4Bh	K	6Bh	k	8Bh	i	ABh	*	CBh	+	EBh	*
0Ch	*	2Ch	,	4Ch	L	6Ch	l	8Ch	i	ACH	*	CCh	+	ECh	*
0Dh	*	2Dh	-	4Dh	M	6Dh	m	8Dh	i	ADh	!	CDh	=	EDh	*
0Eh	*	2Eh	.	4Eh	N	6Eh	n	8Eh	A	Aeh	<	Ceh	+	Eeh	*
0Fh	*	2Fh	/	4Fh	O	6Fh	o	8Fh	A	Afh	>	Cfh	+	Efh	*
10h	>	30h	0	50h	P	70h	p	90h	E	B0h	*	D0h	+	F0h	=
11h	<	31h	1	51h	Q	71h	q	91h	a	B1h	*	D1h	+	F1h	+
12h	*	32h	2	52h	R	72h	r	92h	A	B2h	*	D2h	+	F2h	>
13h	*	33h	3	53h	S	73h	s	93h	o	B3h		D3h	+	F3h	<
14h	*	34h	4	54h	T	74h	t	94h	o	B4h	+	D4h	+	F4h	*
15h	*	35h	5	55h	U	75h	u	95h	o	B5h	+	D5h	+	F5h	*
16h	-	36h	6	56h	V	76h	v	96h	u	B6h	+	D6h	+	F6h	*
17h	*	37h	7	57h	W	77h	w	97h	u	B7h	+	D7h	+	F7h	~
18h	^	38h	8	58h	X	78h	x	98h	y	B8h	+	D8h	+	F8h	O
19h	v	39h	9	59h	Y	79h	y	99h	o	B9h	+	D9h	+	F9h	*
1Ah	>	3Ah	:	5Ah	Z	7Ah	z	9Ah	u	BAh		DAh	+	FAh	*
1Bh	<	3Bh	;	5Bh	[	7Bh	{	9Bh	c	BBh	+	DBh	*	FBh	V
1Ch	*	3Ch	<	5Ch	\	7Ch		9Ch	L	BCh	+	DCh	*	FCh	n
1Dh	*	3Dh	=	5Dh	]	7Dh	}	9Dh	Y	BDh	+	DDh	*	FDh	2
1Eh	^	3Eh	>	5Eh	^	7Eh	~	9Eh	P	BEh	+	DEh	*	FEh	*
1Fh	v	3Fh	?	5Fh	_	7Fh	*	9Fh	S	BFh	+	DFh	*	FFh	SP

00h *	20h SP	40h @	60h `	80h c	a0h a	c0h +	e0h *
01h *	21h !	41h A	61h a	81h u	a1h i	c1h +	e1h *
02h *	22h "	42h B	62h b	82h e	a2h o	c2h +	e2h *
03h *	23h #	43h C	63h c	83h a	a3h u	c3h +	e3h *
04h *	24h \$	44h D	64h d	84h a	a4h n	c4h -	e4h *
05h *	25h %	45h E	65h e	85h a	a5h N	c5h +	e5h *
06h *	26h &	46h F	66h f	86h a	a6h a	c6h +	e6h *
07h *	27h '	47h G	67h g	87h c	a7h o	c7h +	e7h *
08h *	28h (	48h H	68h h	88h e	a8h ?	c8h +	e8h *
09h *	29h)	49h I	69h i	89h e	a9h -	c9h +	e9h *
0ah *	2ah *	4ah J	6ah j	8ah e	aah -	cah +	eah *
0bh *	2bh +	4bh K	6bh k	8bh i	abh *	cbh +	ebh *
0ch *	2ch ,	4ch L	6ch l	8ch i	ach *	cch +	ech *
0dh *	2dh -	4dh M	6dh m	8dh i	adh !	cdh =	edh *
0eh *	2eh .	4eh N	6eh n	8eh A	aeh <	ceh +	eeh *
0fh *	2fh /	4fh O	6fh o	8fh A	afh >	cfh +	efh *
10h >	30h 0	50h P	70h p	90h E	b0h *	d0h +	f0h =
11h <	31h 1	51h Q	71h q	91h a	b1h *	d1h +	f1h +
12h *	32h 2	52h R	72h r	92h A	b2h *	d2h +	f2h >
13h *	33h 3	53h S	73h s	93h o	b3h	d3h +	f3h <
14h *	34h 4	54h T	74h t	94h o	b4h +	d4h +	f4h *
15h *	35h 5	55h U	75h u	95h o	b5h +	d5h +	f5h *
16h -	36h 6	56h V	76h v	96h u	b6h +	d6h +	f6h *
17h *	37h 7	57h W	77h w	97h u	b7h +	d7h +	f7h ~
18h ^	38h 8	58h X	78h x	98h y	b8h +	d8h +	f8h 0
19h v	39h 9	59h Y	79h y	99h o	b9h +	d9h +	f9h *
1ah >	3ah :	5ah Z	7ah z	9ah u	bah	dah +	fah *
1bh <	3bh ;	5bh [	7bh {	9bh c	bbh +	dbh *	fbh V
1ch *	3ch <	5ch \	7ch	9ch L	bch +	dch *	fch n
1dh *	3dh =	5dh]	7dh }	9dh Y	bdh +	ddh *	fdh 2
1eh ^	3eh >	5eh ^	7eh ~	9eh P	beh +	deh *	feh *
1fh v	3fh ?	5fh _	7fh *	9fh S	bfh +	dfh *	ffh SP

terminfo escape sequences

These **terminfo** escape sequences provide alternatives to the standard, built-in escape sequences for each of the functions. Both the default sequences and those specified in **terminfo** work concurrently. The ones in **terminfo** take precedence over the defaults.

terminfo Variable Name	terminfo Capname	AT Keyboard Key Name	Default Escape Sequence	
Numeric Keypad				
key_sleft	kLFT	Left Arrow	4	Esc f
key_sright	kRIT	Right Arrow	6	Esc g
key_sr	kri	Up Arrow	8	Esc t
key_sf	kind	Down Arrow	2	Esc v
key_shome	kHOM	Home	7	Esc h
key_previous	kprv	Pg Up	9	Esc p
key_next	knxt	Pg Down	3	Esc q
key_il	kil1	Ins	0	Esc i
key_sdc	kDC	Del	.	Esc d
key_send	kEND	End	1	Esc e
key_b2	kb2	Center Key	5	Esc .
key_f53	kf13	Num Lock		Esc n
key_f54	kf54	/ Key		Esc [
key_f55	kf55	* Key		Esc *
key_f56	kf56	- Key		Esc -
key_f57	kf57	+ Key		Esc +
key_enter	kent	Enter		Esc ^

Shift Keys

key_f58	kf58	Left Shift		Esc z
key_f59	kf59	Right Shift		Esc /
key_f60	kf60	Ctrl		Esc c
key_f61	kf61	Alt		Esc a
key_f62	kf62	Right Ctrl		Esc C
key_f63	kf63	Right Alt		Esc A

terminfo Variable Name	terminfo Capname	AT Keyboard Key Name	Default Escape Sequence
Cursor Keypad			
key_il	kill	Insert	Esc I
key_home	kHome	Home	Esc H
key_ppage	kpp	Page Up	Esc P
key_dc	kDC	Delete	Esc D
key_end	kEND	End	Esc E
key_npage	knxt	Page Down	Esc Q
key_up	kri	Up Arrow	Esc T
key_left	kLFT	Left Arrow	Esc F
key_down	kind	Down Arrow	Esc V
key_right	kRIT	Right Arrow	Esc G
Special Keys			
key_print	kpri	Print Screen	Esc r
key_f50	kf50	Scroll Lock	Esc s
key_restart	krst	Pause/Break	Esc
key_f51	kf51	Caps Lock	Esc k
key_f52	kf52	102nd key	Esc <
key_options	kopt	Sys Req key	Esc =
key_sic	kIC	Esc key	Esc esc
Function Keys			
key_f1	kf1	F1	Esc 1
key_f2	kf2	F2	Esc 2
key_f3	kf3	F3	Esc 3
key_f4	kf4	F4	Esc 4
key_f5	kf5	F5	Esc 5
key_f6	kf6	F6	Esc 6
key_f7	kf7	F7	Esc 7
key_f8	kf8	F8	Esc 8
key_f9	kf9	F9	Esc 9
key_f10	kf10	F10	Esc 0
key_f11	kf11	F11	Esc ;
key_f12	kf12	F12	Esc :

terminfo Variable Name	terminfo Capname	AT Keyboard Key Name	Default Escape Sequence
Shifted Function Keys			
key_f13	kf13	Shift F1	
key_f14	kf14	Shift F2	
key_f15	kf15	Shift F3	
key_f16	kf16	Shift F4	
key_f16	kf17	Shift F5	
key_f18	kf18	Shift F6	
key_f19	kf19	Shift F7	
key_f20	kf20	Shift F8	
key_f21	kf21	Shift F9	
key_f22	kf22	Shift F10	
key_f23	kf23	Shift F11	
key_f24	kf24	Shift F12	
Controlled Function Keys			
key_f25	kf25	Ctrl F1	
key_f26	kf26	Ctrl F2	
key_f27	kf27	Ctrl F3	
key_f28	kf28	Ctrl F4	
key_f29	kf29	Ctrl F5	
key_f30	kf30	Ctrl F6	
key_f31	kf31	Ctrl F7	
key_f32	kf32	Ctrl F8	
key_f33	kf33	Ctrl F9	
key_f34	kf34	Ctrl F10	
key_f35	kf35	Ctrl F11	
key_f36	kf36	Ctrl F12	

terminfo Variable Name	terminfo Capname	AT Keyboard Key Name	Default Escape Sequence
Alted Function Keys			
key_f37	kf37	Alt F1	
key_f38	kf38	Alt F2	
key_f39	kf39	Alt F3	
key_f40	kf40	Alt F4	
key_f41	kf41	Alt F5	
key_f42	kf42	Alt F6	
key_f43	kf43	Alt F7	
key_f44	kf44	Alt F8	
key_f45	kf45	Alt F9	
key_f46	kf46	Alt F10	
key_f47	kf47	Alt F11	
key_f48	kf48	Alt F12	
Special Actions			
key_f0	kf0	All Shifts Released	Esc Space
key_beg	kbeg	TopBottom Shift Screen	Esc ^U
key_resume	kres	Fork A New Shell	Esc ^F
key_refresh	krfr	Redraw Screen	Esc ^R
key_exit	kext	Kill DOS	Esc ^K
key_suspend	kspd	Suspend DOS	Esc ^Z
key_reference	kref	Display DOS Devices	Esc ^I

Index

A

- A: drive, 18
- Aborting, DOS processes, 151
- Accessing, DOS partition, 90
- Account, SCO Merge, 6
- Adding, printer, 14
- Applications
 - DOS, configuring, 47
 - DOS, installing, 41, 42, 43, 44, 48, 49, 50, 51, 52, 53
 - DOS, removing, 54
 - public, 44
 - running in background, 141, 142
 - SCO Merge, installing, 41, 42
 - shared by DOS and UNIX systems, 44, 47-50
 - tailoring, 70, 71
- ASCII characters, 155
- at(C), using with DOS processes, 141, 142
- Attaching
 - com port, to DOS, 88
 - devices to SCO Merge, 29-34
 - printer port, to DOS, 89
 - printers to DOS, 16
- attrib command, DOS, 23
- autoexec.bat file, 15
 - chcp command, 118
 - device command, 112
 - interpretation under SCO Merge, 67, 72, 100
 - keyb command, 118
 - MODE command, 120
 - mode command, 112, 118
 - NLSFUNC command, 120
 - nlsfunc command, 112, 118
 - running, 100
 - use with NLS, 112, 114, 115, 118, 119, 120

B

- B: drive, 18
- backup command, DOS, 18
- Backups
 - SCO Merge, 18
 - shared filesystem, 18

- Batch files, use with merge command, 61
- batch(C), using with DOS processes, 141, 142
- Booting
 - from a virtual diskette, dosboot command, 28
 - games, from diskette, 98
 - virtual machine, 99
- Bourne shell, 140
 - in an SCO Merge NLS environment, 116
- Brackets, in SCO Merge commands, 83
- BUFFERS command, DOS, 36

C

- C shell, in an SCO Merge NLS environment, 116
- Canadian French code page, 111
- CGA
 - option for specifying, 88
 - SCO Merge, 9
- Changing
 - DOS memory allocation, 71
 - DOS options, with dosadmin command, 67
 - DOS partition protection, 23
- Character set, 108, 109, 112
 - ISO 8859-1, 112
- Characters
 - ASCII, 155, 177
 - ASCII vs PC, 155, 162
 - ISO vs PC, 162, 164
 - PC, 154, 155, 177
- charconv command, SCO Merge, 108, 114, 123, 124, 125, 126
- chcp command, DOS, 108, 112, 116, 117, 118, 123
- chkdsk command, DOS, 133
- chmod(C)
 - changing the protection of DOS partitions, 23
 - configuring DOS applications, 50
 - setting permissions, 12, 45
- Clock interrupt rate, setting with merge
 - fastclk option, 56

Code page, 108, 110, 111, 112, 114, 117, 118, 119, 120, 121
 Canadian French, 111
 hardware, 111
 multilingual, 111
 Nordic, 111
 Portuguese, 111
 purpose of, 110
See also DOS code page, 108
 setting up in DOS, 114
 software, 111
 supported, 111
 switching, 112
 Code set. *See* Code page
 CODEPAGE environment variable, UNIX, 116, 119, 121, 124
 com ports
 attaching to DOS, 88
 SCO Merge, 12, 33
 command.com, passing DOS commands to, from UNIX shell, 95
 Commands
 DOS *See* DOS commands, 53
 shared by DOS and UNIX system, 42
 config.sys file
 buffer keystrokes, 134
 changing special commands, 36
 command for EMS, 132
 country command, 112, 118
 DEVICE command, 120
 device command, 118
 how SCO Merge interprets, 68, 71, 96
 special commands, 36
 specifying alternate, 97
 use with NLS, 112, 114, 115, 118, 119, 120
 Configuration file, *See also* config.sys file
 Configuration menu, dosadmin
 explanation of fields, 69, 70
 selecting, 69
 Configuring
 DOS applications, 7, 49, 50
 DOS applications, with dosadmin command, 41, 42, 47, 48, 49, 51
 DOS characteristics, 65, 66, 67
 print service, 13
 printer, with sysadmsh(ADM), 13
 Console, SCO Merge, 7
 Control, codes, 151-152
 Conventions, notational, 2
 CONVOPTS environment variable, UNIX, 126

copy command, DOS, 101
 Copy protection, 18, 20, 50, 51
 Country code, 118
 country command, DOS, 108, 112, 116, 117, 118
 COUNTRY environment variable, UNIX, 116, 119, 121
 .cpi file, 111, 120
 Creating, dosenv.def file, personal, 75
 cron command, using with DOS processes, 141, 142
 Ctrl-Alt key combination, 150
 Ctrl-Alt-Del keystroke, 150
 Ctrl-C, effects on printing, 101

D

del command, DOS, 54
 Device
 DOS configuration, 30, 31, 32, 33
 SCO Merge, 18
 specifications, 30
 Device access
 canceling specifications, 93
 options available, 91
 preventing, 92
 DEVICE command, DOS, 120
 device command, DOS, 112, 117, 118
 Device specification file. *See* dosdev file
 dir command, DOS, 93
 Direct device attachment, 29, 30, 31, 32, 33, 34
 disable(C)
 disabling com port, 12
 printer, 15
 Disabled, terminal, 151
 Disabling
 com port, 12
 printer, 15
 Disk
 DOS partition, 20, 21, 22, 23, 24
 key, 51
 SCO Merge, 18
 Disk drives, DOS, 18
 setting initial current drive, 96
 Diskette
 See also Floppy disk
 booting from, under SCO Merge, 98

disp, 173
 terminfo descriptor, 160, 161
 Display adapters, 143, 144
 default virtual display for different terminal types, 88
 NLS restrictions, 117
 NLS sample setup, 118
 SCO Merge, 7, 8, 9
 specifying with dos +a option, 88
 supported under SCO Merge, 87
 display modes, terminfo items, 159
 Displaying
 DOS environment variable settings, 137
 DOS options, with dosopt command, 76, 103
 dosenv.def file, 76
 Display-oriented DOS programs, 93-94
 DOS
 See also SCO Merge
 batch files, specifying with dosadmin command, 68
 characteristics, configuring, 65, 66, 67
 command interpreter, 95
 configuration file. *See* config.sys file
 configuring applications, 7, 47, 49, 50
 copy protection, 50, 51
 device file. *See* config.sys file
 display, 7, 8, 9, 10
 Executable File, field, 48
 exit code, setting, 60
 images, 34, 35, 37, 38
 interrupts, 58, 133
 links to UNIX system, 54
 partition, 22, 25, 26
 primary terminal requirement, 153
 printer output, specifying printer, 16
 program execution, from A: or E: drive, 95
 removing applications, 54
 syntax for specifying options, 64, 65
 without SCO Merge, 53
 X client, setting, 104
 DOS application programs
 display adapters, compatibility, 144
 installing, 41, 42, 43, 44, 48, 49, 50, 51, 52
 PC scancode terminal, 146
 Protected Mode, 46, 135
 running as the default shell, 141
 running in background, 141, 142
 scheduling, limitations of, 142
 specifying display-oriented, 94

DOS application programs (*continued*)
 specifying stream-oriented, 94
 tailoring, 70, 71
 DOS character sets, 109, 112
 DOS code page, 108, 110, 111, 112, 114, 117, 118, 119, 120, 121
 hardware, 111
 software, 111
 switching, 112
 dos command, SCO Merge
 +a option, 8
 listing available options, 97
 specifying DOS images, 98
 syntax, 64, 83, 84, 85
 DOS commands
 attrib, 23
 backup, 18
 BUFFERS, 36
 chcp, 108, 112, 116, 117, 118, 123
 chkdsk, 133
 compatibility with SCO Merge, 133
 copy, 101
 country, 108, 112, 116, 117, 118
 del, 54
 DEVICE, 120
 device, 112, 117, 118
 dir, 93
 DOS, 36
 FCBS, 36
 fdisk, 133
 format, 53, 133
 keyb, 108, 112, 116, 117, 118
 MODE, 120
 mode, 17, 108, 112, 116, 117, 118, 123
 NLSFUNC, 120
 nlsfunc, 108, 112, 118
 print, 101
 restore, 18
 sort, 93
 STACKS, 36
 subst, 43, 44
 SWITCHES, 36
 sys, 133
 tree, 93
 type, 94
 DOS images
 defaults, 98
 SCO Merge configuration for, 98
 specifying with dos command, 98
 specifying with dosopt command, 98

- DOS images (*continued*)
 - starting environment without, 98
- DOS memory
 - changing allocations for startup, 71
 - specifying amount, 99
- DOS options
 - See also* dosopt command
 - assigning with dosopt command, 73, 74
 - changing with dosadmin command, 67
 - creating a null record, 104
 - displaying values, 103
 - help screen, 97
 - removing assigned values, 104, 105
 - running autoexec.bat, 100
 - specifying on DOS command line under SCO Merge, 64, 65
- DOS partition
 - defined, 20
 - extended, 20
 - installing applications on, 51, 52
 - physical, 20, 21, 22, 23, 87, 90, 92
 - virtual, 20, 23, 24, 25
- DOS startup file
 - See also* autoexec.bat and config.sys files
 - specifying interpretation of, 71
 - specifying with dos command, 96, 100
 - specifying with dosadmin menus, 72
- dos2unix command, SCO Merge, 108, 124
- dosadmin command, SCO Merge, 65
 - changing DOS options, 67
 - configuring applications, 41, 42, 47, 48, 49, 51
 - defined, 6
 - differences from dosopt command, 73
 - DOS images, 37, 38
 - menu, 48, 49, 71
 - specifying DOS startup file, 68
 - specifying startup memory, 67
 - tailoring DOS application programs with, 71
 - tailoring DOS environment with, 71, 73
 - tailoring DOS operation with, 67, 69, 70
 - virtual diskette, 27, 28
 - virtual partition, 24, 25
- dosboot command, SCO Merge, 99
 - booting from a virtual diskette, 28
- dosdev file, 91
 - device specification file, 32, 33
 - finding the device name for the extended partition, 22
- dosdev file, 91 (*continued*)
 - setting up, 32
 - syntax, 32, 33
- DOSENV environment variable
 - how to set, 139, 140
 - using in UNIX .profile file, 140
- dosenv.def file
 - creating personal, 75
 - differences between personal and system, 66
 - displaying personal, 76
 - displaying system, 76
 - purpose, 68
 - removing installed options, 77
 - restoring installed option files, 77
 - uses for personal, 66, 74
 - uses for system, 74
- doslp, printer, 13, 14
- dosopt command, SCO Merge, 33, 65, 73, 74, 75, 76, 83, 84
 - automatic attachment of DOS partition, 26
 - configuring DOS applications, 50
 - differences from dosadmin command, 73
 - displaying assignments of DOS options, 103
 - displaying current DOS options, 76
 - DOS processes controlled by, 74
 - example uses of, 75, 76
 - explained, 74
 - installing and removing DOS options, 73, 74, 75, 76
 - listing available options, 97
 - null option, 104
 - removing assigned values, 104, 105
 - restoring DOS application default values, 77
 - specifying DOS images, 98
 - syntax, 74, 76, 83, 84, 85
- DOSPATH environment variable
 - how to set, 138
 - when to use, 138

E

- E: drive, 52
- EGA card
 - characteristics and options, 144, 145
 - requirements, 144
 - UNIX MultiScreen functionality, 146

EMS
 configuration for, 132
 SCO Merge command option for, 89
 enable(C), com port, 12
 Enabling
 com port, 12
 printer, 16
 Environment variable, DOS
 CONVOPTS, 125, 126
 displaying settings, 137
 how to set, 137
 PROMPT, 140
 setting from UNIX shell, 139
 setting in UNIX .profile file, 140
 Environment variable, SCO Merge, XMERGE, 131
 Environment variable, UNIX
 CODEPAGE, 116, 119, 121, 124
 CONVOPTS, 126
 COUNTRY, 116, 119, 121
 DOSENV, 139, 140
 DOSPATH, 138
 ISO_TERM, 121
 KEYB, 116, 119, 121
 LANG, 112, 113, 116, 121, 124
 passing to DOS, 136
 Escape sequences
 defined, 156
 special action, 166
 /etc/cshrc, in an SCO Merge NLS environment, 116
 /etc/default/lang, 121
 /etc/dosdev file. *See* dosdev file
 /etc/profile
 in an SCO Merge environment, 119
 in an SCO Merge NLS environment, 116
 Exclusive device access, specifying, 91
 Exit code, DOS, setting, 60
 Expanded memory. *See* EMS

F

F1 function key, simulating on ASCII keyboard, 149
 FCBS command, DOS, 36
 fdisk command, DOS, 133
 File
 permissions, 23
 SCO Merge, 39

Filename, mapping in a multilanguage environment, 110, 122
 Fixed disk, 54
 Floppy disk
 drive, SCO Merge, 18
 mounting, 18
 virtual, 24, 27, 28
 Fonts, 132
 format command, DOS, 53, 133

G

Games
 booting from diskette, 98
 game port, 89
 running under SCO Merge, 99
 getty(M), process, 12

H

Hard disk
 DOS partition, 20-24
 partitions, 21
 SCO Merge, 18
 Hardware devices
 accessible to DOS, 86
 accessing DOS partition, 90
 attaching com ports to DOS, 88
 attaching printer ports to DOS, 89
 default display types, 88
 game port, 89
 keyboard, 86
 mouse, 87
 preventing access, 92
 Help screen, for dos and dosopt commands, 97
 Hercules adapter, option for specifying, 88
 HOME key, simulation on ASCII keyboard, 149
 Hyphen character (-), 102

I

Ill-behaved DOS programs. *See* display-oriented DOS programs
 Images, DOS, 34, 35, 37, 38
 Immediate device access, specifying, 91

Installing

- copy-protected applications, 50, 51
- DOS applications, 41, 42, 44, 46, 48, 49, 50, 51, 52, 53, 135

Interrupt rate, clock, setting with merge
fastclk option, 56

ISO, characters, 162, 164

ISO terminal

- in an SCO Merge NLS environment, 108, 115, 121

SCO Merge NLS setup procedures, 121

using with DOS, 162, 164

ISO_TERM environment variable, UNIX, 121

K

Key disk, 51

Key sequences

- emulating missing keys, 157, 165, 167
- escape, 156
- special action, 166

Key simulation on ASCII keyboard, 150

keyb command, DOS, 108, 112, 116, 117, 118

KEYB environment variable, UNIX, 116, 119, 121

Keyboard

- code, 118
- DOS, 134
- mapping on a serial terminal, 147, 149
- polling, 56
- under SCO Merge, 86

KEYBOARD.SYS file, 120

Killing, DOS processes, 151

L

LANG environment variable, UNIX, 112, 113, 116, 121, 124

Lead-in key sequence

- changing, 168, 175
- default, 175
- defined, 156

Link, DOS/UNIX, 54

Listing

- dos command options, 97
- dosopt command options, 97

ln(C) command, configuring DOS applications, 50

Load image. *See* DOS images

Local

drives, interpreting as remote, 57, 58

files, interpreting as remote, 58

M

Mapping, keyboard on a serial terminal, 147, 149, 150

MDA, option for specifying, 88

Memory, startup

- acceptable values for, 72
- specifying with dosadmin command, 67

merge command

- display option, 60
- drive option, 57, 58
- fastclk option, 56
- handle option, 58
- pollsleep option, 56
- return option, 60
- syntax, 55
- uses for, 55
- using batch files with, 61

Message, files, 113

Minus symbol (-), in SCO Merge commands, 83

mkimg command, SCO Merge, 120

DOS images, 37

MODE command, DOS, 120

mode command, DOS, 17, 108, 112, 116, 117, 118, 123

Monochrome display adapter. *See* MDA

Mounting, floppy disk, SCO Merge, 18

Mouse, 87

Multilingual code page, 111

Multiport serial cards, 132

MultiScreen

- interaction with EGA card, 146
- interaction with VGA card, 146
- SCO Merge, 9

N

National Language Support, *See* NLS, 107

NLS

ASCII terminal support, 114

Australia, 117

autoexec.bat file, 112, 114, 115, 118, 120

background, 109, 110, 111, 112, 113, 114, 115, 116, 117

NLS (continued)

- Belgium, 117
- Bourne shell, 116
- C shell, 116
- Canadian French code page, 111
- capabilities, 108
- CGA support, 114
- character conversion, 110, 114
- character sets, 108, 109, 112
- character translation, 108
- code page, 108, 110, 111, 112, 114, 117, 118, 119, 120, 121
- code set, 108, 110, 111, 112
- CODEPAGE environment variable, 116, 119, 121, 124
- config.sys file, 112, 114, 115, 118, 120
- CONVOPTS environment variable, 126
- countries supported, 117
- country code, 117, 118
- COUNTRY environment variable, 116, 119, 121
- .cpi file, 111, 120
- defined, 107
- display considerations, 113
- display restrictions, 111
- DOS code page, 108
- DOS image, 114, 115, 118, 119, 120
- environment variables, 118
- /etc/cshrc, 116
- /etc/default/lang, 113, 121
- /etc/profile, 116, 119
- filename mapping, 110
- Finland, 117
- France, 117
- French, 112, 118, 119, 121
- Germany, 117
- Hercules support, 114
- ISO terminal setup procedures, 121
- ISO terminal support, 108, 115, 117, 121
- ISO_TERM environment variable, 121
- Italy, 117
- KEYB environment variable, 116, 119, 121
- keyboard code, 117, 118
- KEYBOARD.SYS file, 120
- LANG environment variable, 112, 113, 116, 121, 124
- Latin America, 117
- MDA support, 114
- message files, 113
- MODE command, 120

NLS (continued)

- monitors, 111
- multilingual code page, 111, 118, 119
- Netherlands, 117
- NLSFUNC command, 120
- nlsfunc command, 118
- Nordic code page, 111
- PC scancode terminal support, 114
- Portuguese code page, 111
- prerequisites, 108
- printers, 111
- .profile file, 116, 119, 121
- SCO Merge setup procedures, 118, 119, 120, 122
- Spain, 117
- startup files, 114, 115, 118, 119
- supported displays, 108, 117, 118, 121
- Sweden, 117
- Swiss, 112
- Switzerland, 117
- territory, 112
- under DOS, 111, 112
- under SCO Merge, 113, 114, 115, 116, 117
- under UNIX system, 112, 113
- United Kingdom, 117
- United States, 117
- UNIX code set, 108
- /usr/merge/image/std/vga, 119
- VGA setup procedures, 118, 119, 120
- VGA support, 108, 113, 117
- X Window System support, 108, 113, 117
- NLSFUNC command, DOS, 120
- nlsfunc command, DOS, 108, 112, 118
- nohup command, using with DOS processes, 141, 142
- Nordic code page, 111
- Notation, general conventions, 2

O

- On-card ROM, DOS images, 35, 38
- Options file, dosenv.def, 65, 66, 68, 74, 75, 76, 77

P

- Partition, *See also* DOS partition

Path

- DOS, 38
- search, 45, 54
- Path separators, translating, under SCO Merge, 102
- PC characters, 154, 155, 162, 169, 173
- PC keyboard simulation, 149
- PC scancode terminal
 - input, 175
 - output, 173
 - setting up, 11
 - supported by SCO Merge, 87
 - switchable, 11
 - using, 146
 - using with DOS, 155, 169, 173, 175
- PC scancodes, 154, 155, 162, 164, 169, 173
- PCT file, 162, 173
- pcterm, 167
 - terminfo descriptor, 160
- Permissions
 - DOS partition, 22
 - SCO Merge, 22, 23
- Plus symbol (+), in SCO Merge commands, 83
- Port, com. *See* com port
- Portuguese code page, 111
- print command, DOS, 101
- Print streams, 14
- Printer
 - adding, 14
 - defaults, 13, 14
 - disabling, 15
 - enabling, 16
 - NLS support, 111
 - output, sending to printer, 16
 - timeout, 17, 101
- printer command, SCO Merge, 14, 15, 16, 17, 101, 123
 - using to change printer timeout, 17
- Printer port
 - attaching to DOS, 89
 - identifying, 15
- Printing
 - attaching printers to DOS, 16
 - defaults, 13, 14
 - DOS, specifying printer, 16
 - effects of Ctrl-C keystroke, 101
 - in a multilanguage environment, 123
 - process control, 15
 - SCO Merge, 13, 14, 15, 16

.profile file

- in a SCO Merge NLS environment, 116, 119, 121
- UNIX, 45
- .profile file, UNIX
 - setting DOS environment variables in, 140
 - specifying DOS as default shell, 141
- PROMPT environment variable, how to set, 140
- Protected Mode, 46, 135
- Protection, DOS, 22, 23

R

- Read-only device access, specifying, 91
- Redrawing, screen, 152
- Remote
 - drives, interpreting as local, 57, 58
 - files, interpreting as local, 58
- Removing
 - DOS applications, 54
 - DOS option assigned values, 104, 105
 - installed options, from dosenv.def file, 77
- Resources, X, 129, 130
- restore command, DOS, 18
- Restoring, installed option files, 77
- rm(C), command, 54
- ROM, DOS images, 35, 38
- Running
 - autoexec.bat file, 100
 - DOS application programs, in background, 141, 142
 - games, under SCO Merge, 99

S

- Scancode terminal. *See* PC scancode terminal
- scancodes, 154, 155, 162, 164, 169
- Scheduling commands, UNIX
 - precautions when using with DOS, 142
 - using with DOS, 141
- SCO Merge
 - See also* DOS
 - application programs, 47, 49, 50
 - backup, 18
 - disk drives, 18
 - display, 8, 9, 10
 - environment variable, XMERGE, 131

SCO Merge (*continued*)

- files affected by common administrative procedures, 39
- installing applications, 41
- problems, 134

Screen

- shifting, 151
- switch, 10

Search path, 45, 54

Security

- DOS partition protection, 21, 22, 23
- file permissions, 23
- preventing access to devices, 92
- specifying read-only device access, 91

Selecting, Configuration menu, dosadmin, 69

Sequence, escape, 156

Serial card, multiport, 132

Serial terminal, 87

- keyboard mapping, 147, 149, 150
- limitations, 147
- setting up, 11
- TERM setting, 150
- using with DOS, 154, 155, 162

Shared

- DOS/UNIX applications, 44-50
- DOS/UNIX filesystem, configuring the DOS environment, 57, 58
- DOS/UNIX filesystem, DOS interrupts, 133, 134
- filesystem, backup procedures, 18

Sharing, disk drives, 18

Shell, UNIX, setting DOS environment variables from, 139, 140

Slash character (/), 102

Snapshots, 34

sort command, DOS, 93

Special, action escape sequences, 166

Specifications, device, 30

STACKS command, DOS, 36

Standard ROM, DOS images, 35, 38

standard terminals, output, 157

Starting

DOS, 52

DOS environment, without DOS image, 98

Startup memory, acceptable values for, 72

Stopping, DOS processes, 151

Stream-oriented DOS programs

- definition, 93
- specifying, 93
- test for verifying, 94

subst command, DOS, 43, 44

Superkey, 134

Switch characters, translating, under SCO Merge, 102

Switch screens, 10

Switchable scancode terminals, examples of, 11

SWITCHES command, DOS, 36

switchkey command, SCO Merge, 10

Syntax

- dos command, 64, 83, 84, 85
- dosdev file, 32, 33
- dosopt command, 74, 76, 83, 84, 85
- merge command, 55
- specifying DOS options, 64, 65

sys command, DOS, 133

sysadmsh(ADM), configuring printer, 13

System

- backup, 18
- file, 39

System administrator, DOS, 6

T

Tailoring DOS Environment menu, 72

TERM environment variable, setting, 12

Terminal

- disabled, 151
- display adapters used, 88
- input, 164
- NLS sample setup for ISO, 121
- scancode. *See* PC scancode terminal, 147
- serial. *See* Serial terminal, 147
- standard vs PC scancode, 169
- supported by SCO Merge, 87, 143
- using with DOS, 153, 155, 162, 164

Terminals, standard vs PC scancode, 155

terminfo(F)

- defined, 154
- files, 154

standard terminal output, 157

terminfo(M), 153

Text files, converting, 124

tree command, DOS, 93

Troubleshooting, SCO Merge, 134

type command, DOS, 94

U

- UNIX code set, 108, 110, 111, 112
- UNIX commands, using scheduling commands
 - with DOS, 141, 142
- UNIX message files, 113
- UNIX print spooler
 - bypassing, for DOS printing, 16
 - changing printing timeout, 17
- UNIX shell
 - control code for forking, 151
 - default, 140, 141
 - setting DOS environment variables from, 139, 140
- unix2dos command, SCO Merge, 108, 124
- User, SCO Merge, 6
- /usr/merge/image/std/vga, use with NLS, 119

V

- VGA
 - in an SCO Merge NLS environment, 108, 117, 118
 - SCO Merge, 9
- VGA card
 - requirements, 146
 - UNIX MultiScreen functionality, 146
- Virtual
 - diskette, 24
 - DOS floppies, 27
 - partition, 23, 24, 25
- Virtual machine, booting, 99

W

Well-behaved DOS programs. *See* stream-oriented DOS programs

X

- X client setting, DOS, 104
- X resources, 129, 130
- X Window System, in an SCO Merge NLS environment, 108
- XMERGE environment variable, 131

Please help us to write computer manuals that meet your needs by completing this form. Please post the completed form to the Publications Manager nearest you: The Santa Cruz Operation, Ltd., Croxley Centre, Hatters Lane, Watford WD1 8YN, United Kingdom; The Santa Cruz Operation, Inc., 400 Encinal Street, P.O. Box 1900, Santa Cruz, California 95061, USA or SCO Canada, Inc., 130 Bloor Street West, 10th Floor, Toronto, Ontario, Canada M5S 1N5.

Volume title: _____
(Copy this from the title page of the manual)

Product: _____
(for example, SCO UNIX System V Release 3.2 Operating System Version 4.0)

How long have you used this product?

- ☐ Less than one month ☐ Less than six months ☐ Less than one year
☐ 1 to 2 years ☐ More than 2 years

How much have you read of this manual?

- ☐ Entire manual ☐ Specific chapters ☐ Used only for reference

	Agree		Disagree	
The software was fully and accurately described	☐	☐	☐	☐
The manual was well organized	☐	☐	☐	☐
The writing was at an appropriate technical level (neither too complicated nor too simple)	☐	☐	☐	☐
It was easy to find the information I was looking for	☐	☐	☐	☐
Examples were clear and easy to follow	☐	☐	☐	☐
Illustrations added to my understanding of the software	☐	☐	☐	☐
I liked the page design of the manual	☐	☐	☐	☐

If you have specific comments or if you have found specific inaccuracies, please report these on the back of this form or on a separate sheet of paper. In the case of inaccuracies, please list the relevant page number.

May we contact you further about how to improve SCO documentation?
If so, please supply the following details:

Name _____ Position _____

Company _____

Address _____

City & Post/Zip Code _____

Country _____

Telephone _____ Facsimile _____



1 July 1993



BH03702P001

65543



AU03701P001